BCIS 1305 Business Computer Applications

San Jacinto College

This course was developed from generally available open educational resources (OER) in use at multiple institutions, drawing mostly from a primary work curated by the Extended Learning Institute (ELI) at Northern Virginia Community College (NOVA), but also including additional open works from various sources as noted in attributions on each page of materials.

Cover Image: “Keyboard” by John Ward from https://flic.kr/p/tFuRZ licensed under a Creative Commons Attribution License.

BCIS 1305 Business Computer Applications by Extended Learning Institute (ELI) at NOVA is licensed under a Creative Commons Attribution 4.0 International License, except where otherwise noted.

CONTENTS

MODULE 1: INTRODUCTION TO COMPUTERS

READING: FILE SYSTEMS

https://saylor.org/course/viewphp?id=94&sectionid=973

Licensing & Attributions

Public domain content

- License: *Public Domain: No Known Copyright*

READING: BASIC COMPUTER SKILLS

https://www.oercommons.org/courses/basic-computer-skills

Licensing & Attributions

Public domain content

- License: *Public Domain: No Known Copyright*

READING: COMPUTER CONCEPTS

https://www.oercommons.org/courses/computer-concepts-video-lectures

Licensing & Attributions

Public domain content

- License: *Public Domain: No Known Copyright*

TUTORIALS: COMPUTER BASICS

Tutorial is provided by Goodwill Community Foundations tutorials HERE. & covers the following topics.

1. Getting to Know Computers

2. Understanding Operating Systems
3. Understanding Applications
4. Web Apps and the Cloud
5. Basic Parts of a Desktop Computer
6. Buttons, Sockets and Slots on a Desktop Computer
7. Inside a Desktop Computer
8. Laptop Computers and Netbooks
9. Getting to Know Mobile Devices
10. Setting Up a Computer
11. Beginning to Use Your Computer
12. Getting to Know the OS
13. Connecting to the Internet
14. Computer Safety and Maintenance
15. Basic Troubleshooting Techniques

Work through the Computer Basic Lessons, Interactives, and take the quiz under Extras. The quiz is not graded, but will provide you a good feedback on your understanding of the content.

Licensing & Attributions

Public domain content

- License: *Public Domain: No Known Copyright*

MODULE 2: COMPUTER HARDWARE & MEMORY

READING: HARDWARE

Introduction

Hardware is the most visible part of any information system: the equipment such as computers, scanners and printers that is used to capture data, transform it and present it to the user as output. Although we will focus mainly on the personal computer (PC) and the peripheral devices that are commonly used with it, the same principles apply to the complete range of computers:

- Supercomputers, a term used to denote the fastest computing engines available at any given time, which are used for running exceptionally demanding scientific applications.
- Mainframe computers, which provide high-capacity processing and data storage facilities to hundreds or even thousands of users operating from (dumb) terminals.
- Servers, which have large data storage capacities enabling users to share files and application software, although processing will typically occur on the user's own machine.
- Workstations, which provide high-level performance for individual users in computationally intensive fields such as engineering.
- Personal computers (including laptop/notebook computers) have a connected monitor, keyboard and CPU, and have developed into a convenient and flexible business tool capable of operating independently or as part of an organizational network.
- Mobile devices such as personal digital assistants or the latest generation of cellular telephones, offer maximum portability plus wireless connection to the internet, although they do not offer the full functionality of a PC.

And we are already moving into the age of wearable computers for medical or security applications, embedded computers in appliances ranging from motor cars to washing machines, and the smart card which will provide identification, banking facilities, medical records and more!

Input devices

Data may enter an information system in a variety of different ways, and the input device that is most appropriate will usually depend on the type of data being entered into the system, how frequently this is done, and who is responsible for the activity. For example, it would be more efficient to scan a page of typed text into an information system rather than retyping it, but if this happens very seldom, and if typing staff are readily available, then the cost of the scanner might not be justified. However, all of the input devices described in this chapter have at least one thing in common: the ability to translate non-digital data types such as text, sound or graphics into digital (i.e. binary) format for processing by a computer.

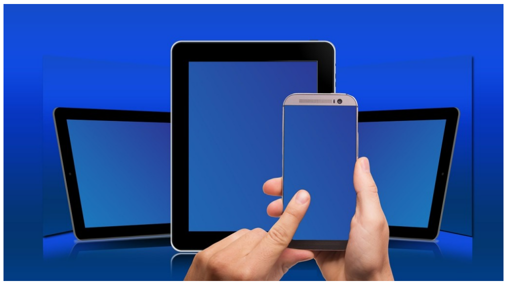
The keyboard

A lot of input still happens by means of a keyboard. Usually, the information that is entered by means of a keyboard is displayed on the monitor. The layout of most keyboards is similar to that of the original typewriter on which it was modeled. Ironically, this “QWERTY” keyboard layout was originally designed to slow the operator down, so that the keys of the typewriter would not get stuck against each other. This layout now works counter-productively since a computer can process keyboard input many times faster than even the fastest typist can manage. A number of attempts have been made to design alternative layouts by rearranging the keys (the Dvorak keyboard) or by reducing the number of keys. None of these alternative designs has really caught on. Special keyboards have also been designed for countries that use a non-Roman alphabet, and also for disabled people.

Pointing devices

The now ubiquitous electronic *mouse* is an essential input device for use with any graphical user interface. It consists of a plastic moulded housing, designed to fit snugly in the palm of the hand, with a small ball at its bottom. Moving the mouse across a flat surface will translate the movements into a rolling action of the ball. This is translated into electronic signals that direct the corresponding movement of a cursor on the computer monitor. Buttons on the mouse can then be used to select icons or menu items, or the cursor can be used to trace drawings on the screen.

The less popular *trackball* operates exactly like an “upside-down” mouse except that the ball is much larger and, instead of the mouse being moved over a surface, the user manipulates the ball directly. Since the trackball can be built into the side of the keyboard, it obviates the need for a free surface area and is therefore handy in situations where desktop surface area is at a premium or not available. Originally popular in educational laboratory settings and for laptop computers, trackballs are now mainly confined to exhibition displays and other public terminals.



Touch-screens are computer monitors that incorporate sensors on the screen panel itself or its sides. The user can indicate or select an area or location on the screen by pressing a finger onto the monitor. *Light and touch pens* work on a similar principle, except that a stylus is used, allowing for much finer control. Touch pens are more commonly used with handheld computers such as personal organizers or digital assistants. They have a pen-based interface whereby a stylus (a pen without ink) is used on the small touch-sensitive screen of the handheld computer, mainly by means of ticking off pre-defined options, although the fancier models support data entry either by means of a stylized alphabet, which resembles a type of shorthand, or some other more sophisticated handwriting recognition interface.

Digitizer tablets also use a pressure sensitive area with a stylus. This can be used to trace drawings. A similar conceptual approach is used for the touch pad that can be found on the majority of new notebook computers, replacing the more awkward joystick or trackball. The user controls the cursor by moving a finger across a fairly small rectangular touch-sensitive area below the keyboard, usually about 5 cm by 7 cm.

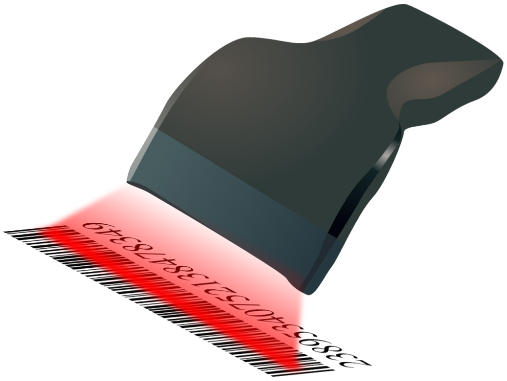
A large number of game interfaces have been developed to provide a more realistic and natural interface in various gaming situations and simulations: the joy stick, steering wheel, foot pedal and other gaming devices. They all perform functions similar to the mouse in that they allow the user to control a cursor or simulate generally real-time motion control. Contact your nearest game arcade for details.

Although the data glove also fits under the previous category, it is technically a lot more complex. It looks like a hand glove but contains a large number of sensors and has a data cable attached; though the latter is being replaced by means of infrared cordless data transmission. Not only does the data glove allow for full three-dimensional movement but it also senses the position of individual fingers, translating this into a grip. The glove is currently used in virtual reality simulators where the user moves around in an artificially rendered environment projected onto tiny LCD screens fitted into vision goggles. The computer generates various imaginary objects, which the user can “pick up” and manipulate by means of the glove. Advanced models even allow for tactile feedback by means of small pressure pockets built into the glove.

Optical scanners and readers

There are a number of different optical scanner technologies on the market.

- Optical Scanners use light-emitting devices to illuminate the printing on paper. Depending on how much light is reflected, a light-sensor determines the position and darkness (or color) of the markings on the paper. Special-purpose optical scanners are in use by postal services to read and / interpret hand-written postal codes. General-purpose scanners are used with personal computers to scan in images or text. These vary from handheld devices (see picture) to flatbed scanners which feed input documents one sheet at a time. A common use of optical scanners is the scanning of black-and-white or color images and pictures. When scanning text, it is necessary to load additional optical character recognition (OCR) software that converts the scanned raster-image of the text into the equivalent character symbols, so that they can be edited using word processing software.
- Barcode scanners detect sequences of vertical lines of different widths, the ubiquitous barcode as found also on the back of this book. These scanners have become very popular with retailers due to the fact that all pre-packaged products are now required to have a product bar code on their packaging, following the standard laid down by the South African Article Numbering Association (SAANA). Libraries and video shops now also commonly use bar code scanners. They are more generally used for tracking and routing large numbers of physical items such as for asset inventory purposes in many larger organizations, postal items by the postal services and courier services, or for luggage handling by airlines.
- Optical mark readers are capable of reading dark marks on specially designed forms. The red multiple choice answer sheets in use at many educational and testing institutions are a good example.



Other input devices

A magnetic card reader reads the magnetized stripe on the back of plastic credit-card size cards. These cards need to be pre-recorded following certain standards. Although the cards can hold only a tiny amount of information, they are very popular for access (door) control and financial transactions (ATMs and point-of-sale terminals).

Magnetic ink character recognition (MICR) uses a special ink (containing magnetizable elements) and a distinct font type. It is used mainly in the banking sector for the processing of cheques.

Touch-tone devices can use a voice telephone to contact computer-based switchboards or enter information directly into remote computers. Many corporate telephone help-lines rely on the customer pressing the touch-tone telephone buttons to route his/her call to the correct operator by selecting through a menu of possible options. South African banks also enable their clients to perform a number of banking transactions via telephone.

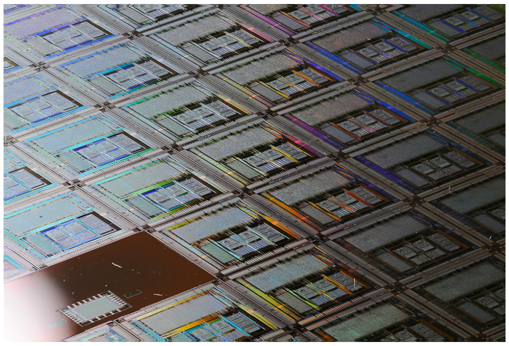
Digital cameras allow you to make pictures of physical objects directly in a digital, i.e. computer-readable, format. Relatively low-cost digital still picture cameras are now available that capture images directly on electronic disk or RAM media instead of the traditional film. Apart from being very compact, most of these digital cameras can also interface directly with personal computers and are thus becoming a popular tool to capture pictures for e-mailing or loading on the world-wide Web.

Biometric devices are used to verify personal identity based on fingerprints, iris or retinal scanning, hand geometry, facial characteristics etc. A scanning device is used to capture key measurements and compare them against a database of previously stored information. This type of authentication is becoming increasingly important in the control of physical access.

Finally, voice input devices are coming of age. Voice-recognition has recently made a strong entry into the market with the availability of low-cost systems that work surprisingly well with today's personal computers. These systems allow for voice control of most standard applications (including the operating system). With voice control, the computer recognizes a very limited number (50 or less) of frequently used, programmable system commands ("save", "exit", "print"…) from a variety of users. In fact, these systems are not only used for the interface of computer programs; they are also slowly making an appearance in consumer appliances, novelty items and even motor cars!

Much more difficult to achieve than voice control, is true voice dictation used to dictate e.g. a letter to your word processor. The difficulty is that the computer must not only distinguish between many tens of thousands of possible words, but it must also recognize the almost unnoticeable breaks in between words, different accents and intonations. Therefore, voice dictation typically requires a user to train the voice recognition software by reading standard texts aloud. Nevertheless, for personal purposes and slow typists, voice recognition is rapidly becoming a viable alternative to the keyboard.

Central Processing Unit (CPU)



Once data has been entered into a computer, it is acted on by the CPU, which is the real brain of the computer. The CPU takes specific program instructions (usually one at a time), applies them to the input data and transforms the input into output.

Components of the CPU

The CPU has two major components.

- The Arithmetic and Logic Unit (ALU) executes the actual instructions. It knows how to add or multiply numbers, compare data, or convert data into different internal formats.
- The Control Unit does the "housekeeping" i.e. ensures that the instructions are processed on time, in the proper sequence, and operate on the correct data.

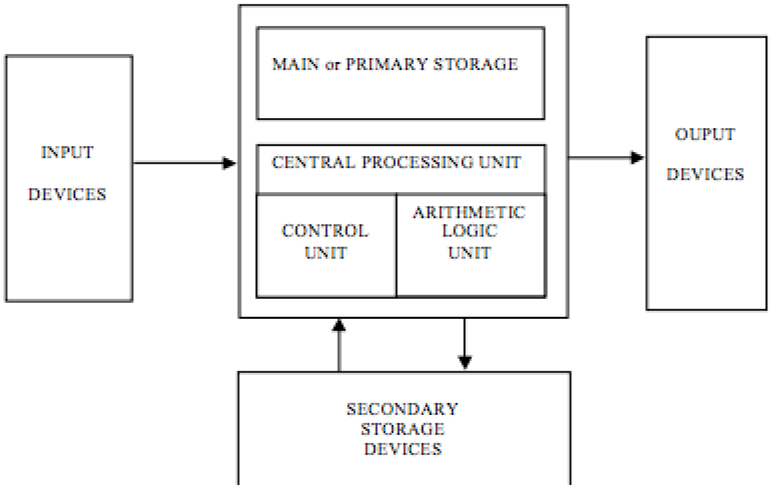


Figure 1: Detailed view of a computer system

Types of CPUs

The CPU is an electronic device based on microchip technology, hence it is also often called the microprocessor. It is truly the showcase and culmination of the state-of-the-art in the electronics industry: a tiny silicon-based chip occupying less than 1 square cm contains several millions of transistor elements, measuring less than a thousandth of a millimeter across. They operate at speeds way beyond our comprehension: a typical CPU can multiply more 7-digit numbers in one second than a human could do in ten lifetimes, but uses less energy than a light bulb!

Think of the motor car industry: there are different manufacturers or makes of cars (Volkswagen, Toyota, etc.), each with different models (Golf, Jetta, …), which come out in different versions (City Golf, Sports model, coupe, etc.). In addition, there exist custom-made special-purpose cars. It is the same in the computer chip business. There are many different types of CPUs on the market. The best-known manufacturer is Intel, which produces the microprocessors for the IBM-compatible personal computer (PC). Some of its competitors produce clones or imitations (e.g. AMD), others manufacturers produce different types of microprocessors or concentrate on small volumes of highly specialized or very fast microprocessors. Intel has produced a large number of CPU types: the earliest model used in the Personal Computer was the 8088, followed by the 8086, the 80286, the 386, 486 and the line of Pentium processors.

Speed of processing

How does one measure the speed of, say a Porsche 911? One could measure the time that it takes to drive a given distance e.g. the 900 km from Cape Town to Bloemfontein takes 4'/2 hours (ignoring speed limits and traffic jams). Alternatively, one can indicate how far it can be driven in one standard time unit e.g. the car moves at a cruising speed of 200 km/hour.

In the same way, one can measure the speed of the CPU by checking the time it takes to process one single instruction. As indicated above, the typical CPU is very fast and an instruction can be done in about two billionths of a second. To deal with these small fractions of time, scientists have devised smaller units: a millisecond (a thousandth of a second), a microsecond (a millionth), a nanosecond (a billionth) and a picosecond (a trillionth).

However, instead of indicating the time it takes to execute a single instruction, the processing speed is usually indicated by how many instructions (or computations) a CPU can execute in a second. This is exactly the inverse of the previous measure; e.g. if the average instruction takes two billionths of a second (2 nanoseconds) then the CPU can execute 500 million instructions per second (or one divided by 2 billionths). The CPU is then said to operate at 500 MIPS or 500 million of instructions per second. In the world of personal computers, one commonly refers to the rate at which the CPU can process the simplest instruction (i.e. the clock rate). The CPU is then rated at 500 MHz (megahertz) where mega indicates million and Hertz means “times or cycles per second”. For powerful computers, such as workstations, mainframes and supercomputers, a more complex instruction is used as the basis for speed measurements, namely the so-called floating-point operation. Their speed is therefore measured in megaflops (million of floating-point operations per second) or, in the case of very fast computers, teraflops (billions of flops).

In practice, the speed of a processor is dictated by four different elements: the “clock speed”, which indicates how many simple instructions can be executed per second; the word length, which is the number of bits that can be processed by the CPU at any one time (64 for a Pentium IV chip); the bus width, which determines the number of bits that can be moved simultaneously in or out of the CPU; and then the physical design of the chip, in terms of the layout of its individual transistors. The latest Pentium processor has a clock speed of about 4 GHz and contains well over 100 million transistors. Compare this with the clock speed of 5 MHz achieved by the 8088 processor with 29 000 transistors!

Moore’s Law (see Figure 2) states that processing power doubles for the same cost approximately every 18 months.

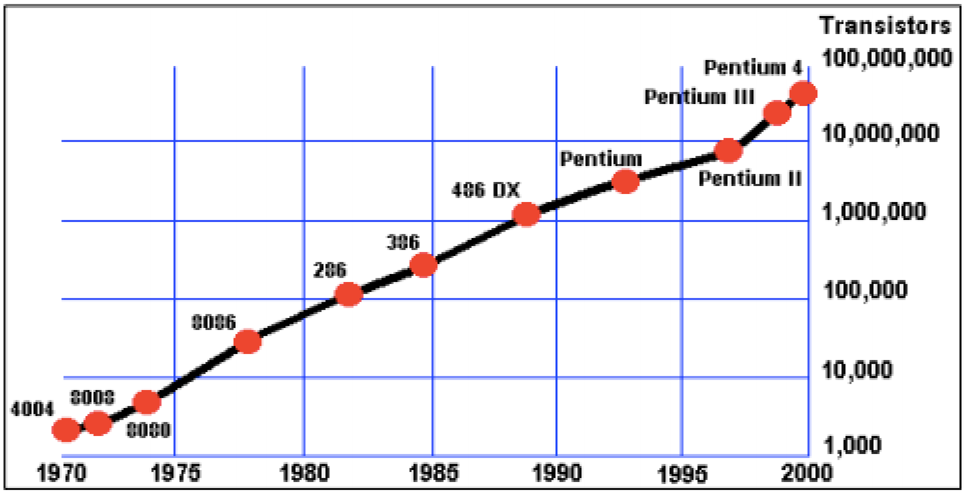


Figure 2. Illustration of Moore's Law

Von Neumann versus Parallel CPU Architecture

The traditional model of the computer has one single CPU to process all the data. This is called the Von Neumann architecture because he engineered this approach to computers in the days when computers were still a dream.

Except for entry-level personal computers, most computers now have two, four, or up to sixteen CPUs sharing the main processing load, plus various support processors to handle maths processing, communications, disk I/O, graphics or signal processing. In fact many CPU chips now contain multiple "cores" each representing an individual CPU.

Some super-computers that have been designed for massive parallel processing, have up to 64,000 CPUs. These computers are typically used only for specialized applications such as weather forecasting or fluid modeling. Today's supercomputers are mostly clusters (tight networks) of many thousands of individual computers.

Possible Future CPU Technologies

Perhaps the major future competitor of the microchip-based microprocessor is optical computing. Although the technology for developing electronic microchips suggests that CPUs will continue to increase in power and speed for at least the next decade or so, the physical limits of the technology are already in sight. Switching from electronic to light pulses offers a number of potential advantages: light (which consists of photons) can travel faster, on narrower paths and does not disperse heat. In theory, one can even process different signals (each with a different light frequency) simultaneously using the same channel. Although the benefits of optical processing technology have already been proven in the areas of data storage (CD-Rom, CD-R) and communication (fibre optics), the more complex all-optical switches required for computing are still under development in the research laboratories.

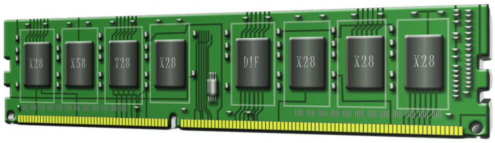
A very experimental alternative to optical and electronic technologies is the organic computer. Research indicates that, for certain applications, it is possible to let a complex organic molecule act as a primitive information processor. Since even a tiny container filled with the appropriate solutions contains many trillions of these molecules, one obtains in effect a hugely parallel computer. Although this type of computer can attack combinatorial problems way beyond the scope of traditional architectures, the main problem is that the programming of the bio-computer relies entirely on the bio-chemical properties of the molecules.

Another exciting but currently still very theoretical development is the possible use of quantum properties as the basis for a new type of computer architecture. Since quantum states can exist in juxtaposition, a register of qubits (a bit value in quantum state) takes on all the possible values simultaneously until it is measured. This could be exploited to speed up extremely parallel algorithms and would affect such areas as encryption, searching and error-correction. To date, experimental computers with a few qubits have been built but the empirical validation of the actual usefulness of quantum computing still remains an open question.

Main Memory

The function of main memory (also referred to as primary memory, main storage or internal storage) is to provide temporary storage for instructions and data during the execution of a program. Main memory is usually known as RAM, which stands for Random Access Memory. Although microchip-based memory is virtually the only technology used by today's computers, there exist many different types of memory chips.

Random Access Memory (RAM)



RAM consists of standard circuit-inscribed silicon microchips that contain many millions of tiny transistors. Very much like the CPU chips, their technology follows to the so-called law of Moore, which states that they double in capacity or power (for the same price) every 18 months. A RAM chip easily holds hundreds of Megabytes (million characters). They are frequently pre-soldered in sets on tiny memory circuit boards called SIMMS (Single In-line Memory Modules) or DIMMS (Dual …) which slot directly onto the motherboard: the main circuit board that holds the CPU and other essential electronic elements. The biggest disadvantage of RAM is that its contents are lost whenever the power is switched off.

There are many special types of RAM and new acronyms such as EDO RAM, VRAM etc. are being created almost on a monthly basis. Two important types of RAM are:

- Cache memory is ultra-fast memory that operates at the speed of the CPU. Access to normal RAM is usually slower than the actual operating speed of the CPU. To avoid slowing the CPU down, computers usually incorporate some more expensive, faster cache RAM that sits in between the CPU and RAM. This cache holds the data and programs that are needed immediately by the CPU. Although today's CPUs already incorporate an amount of cache on the circuit itself, this on-chip cache is usually supplemented by an additional, larger, cache on the motherboard.
- Flash RAM or flash memory consists of special RAM chips on a separate circuit board within a tiny casing. It fits into custom ports on many notebooks, hand-held computers and digital cameras. Unlike normal RAM, flash memory is non-volatile i.e. it holds it contents even without external power, so it is also useful as a secondary storage device.

Read-Only Memory (ROM)

A small but essential element of any computer, ROM also consists of electronic memory microchips but, unlike RAM, it does not lose its contents when the power is switched off. Its function is also very different from that of RAM. Since it is difficult or impossible to change the contents of ROM, it is typically used to hold program instructions that are unlikely to change during the lifetime of the computer. The main application of ROM is to store the so-called boot program: the instructions that the computer must follow just after it has been switched on to perform a self-diagnosis and then tell it how load the operating system from secondary storage. ROM chips are also found in many devices which contain programs that are unlikely to change over a significant period of time, such as telephone switch boards, video recorders or pocket calculators. Just like RAM, ROM comes in a number of different forms:

- PROM (Programmable Read-Only Memory) is initially empty and can be custom-programmed once only using special equipment. Loading or programming the contents of ROM is called burning the chip since it is the electronic equivalent of blowing tiny transistor fuses within the chip. Once programmed, ordinary PROMs cannot be modified afterwards.
- EPROM (Erasable Programmable Read-Only Memory) is like PROM but, by using special equipment such as an ultra-violet light gun, the memory contents can be erased so that the EPROM can be re-programmed.
- EEPROM (Electrically Erasable Programmable Read-Only Memory) is similar to EPROM but it can be re-programmed using special electronic pulses rather than ultraviolet light so no special equipment is required.

Secondary Storage Devices

Since the main memory of a computer has a limited capacity, it is necessary to retain data in secondary storage between different processing cycles. This is the medium used to store the program instructions as well as the data required for future processing. Most secondary storage devices in use today are based on magnetic or optical technologies.

Disk drives

The disk drive is the most popular secondary storage device, and is found in both mainframe and microcomputer environments. The central mechanism of the disk drive is a flat disk, coated with a magnetizable substance. As this disk rotates, information can be read from or written to it by means of a head. The head is fixed on an arm and can move across the radius of the disk. Each position of the arm corresponds to a “track” on the disk, which can be visualized as one concentric circle of magnetic data. The data on a track is read sequentially as the disk spins underneath the head. There are quite a few different types of disk drives.

In Winchester hard drives, the disk, access arm and read/write heads are combined in one single sealed module. This unit is not normally removable, though there are some models available where the unit as a whole can be swapped in and out of a specially designed drive bay. Since the drives are not handled physically, they are less likely to be contaminated by dust and therefore much more reliable. Mass production and technology advances have brought dramatic improvements in the storage capacity with Terabyte hard drives being state of the art at the end of 2006. Current disk storage costs as little Rl per gigabyte.

Large organizations such as banks, telcos and life insurance companies, require huge amounts of storage space, often in the order of many terabytes (one terabyte is one million megabytes or a trillion characters). This was typically provided by a roomful of large, high-capacity hard drive units. Currently, they are being replaced increasingly by redundant arrays of independent disks (RAIDs). A RAID consists of an independently powered cabinet that contains a number (10 to 100) of microcomputer Winchester-type drives but functions as one single secondary storage unit. The advantage of the RAID is its high-speed access and relatively low cost. In addition, a RAID provides extra data security by means of its fault-tolerant design whereby critical data is mirrored (stored twice on different drives) thus providing physical data redundancy. Should a mirrored drive fail, the other drive steps in automatically as a backup.

A low-cost, low-capacity version of the hard disk was popularized by the microcomputer. The diskette consists of a flexible, magnetic ti^J surface coated mylar disk inside a thin, non-removable, plastic sleeve. The early versions of the diskette were fairly large (8″ or 5W) and had a flexible sleeve, hence the name floppy diskette. These have rapidly been replaced by a diskette version in a sturdier sleeve, the stiffy disk, that despite its smaller size (3 W’) can hold more data. Although the popular IBM format only holds 1,44 megabytes, a number of manufacturers have developed diskette drives that can store from 100 to 250 megabytes per stiffy. An alternative development is the removable disk cartridge, which is similar in structure to an internal hard drive but provides portability, making it useful for backup purposes.

Magnetic tape

While disk and optical storage have overtaken magnetic tape as the most popular method of storing data in a computer, tape is still used occasionally – in particular for keeping archive copies of important files.

The main drawback of magnetic tape is that it is not very efficient for accessing data in any way other than strictly sequential order. As an illustration, compare a CD player (which can skip to any track almost instantly) with a music tape recorder (which has to wind the tape all the way through if one wants to listen to a song near the end). In computer terms, the ability to access any record, track, or even part within a song directly is called the direct access method. In the case of the tape recorder one may have to wind laboriously through the tape until one reaches the song required – this is referred to as the sequential access method.

The high-density diskette and recordable optical disk have all but eroded the marginal cost advantage that tape storage enjoyed. This technology is therefore disappearing fast.

Optical disk storage

Optical disks, on the other hand, are rapidly becoming the storage medium of choice for the mass distribution of data/programs and the backup of data. Similar to disk storage, information is stored and read from a circular disk. However, instead of a magnetic read head, a tiny laser beam is used to detect microscopic pits burnt onto a plastic disk coated with reflective material. The pits determine whether most of the laser light is reflected back or scattered, thus making for a binary “on” or “off”. In contrast to hard disks, data is not stored in concentric cylinders but in one long continuous spiral track.

Trivial fact: The spiral track used to store data on a CD is over six kilometers long.

A popular optical disk format is the 12-cm CD-ROM. The widespread use of music compact discs has made the technology very pervasive and cheap. Production costs for a CD-ROM are less than RI, even for relatively small production volumes. The drive reader units themselves have also dropped in price and are now hardly more than the cost of a diskette drive. A standard CD-ROM can store 650 megabytes of data and the data can be transferred at many megabytes per second, though accessing non-sequential data takes much longer.

The CD-ROM is a read-only medium. Data cannot be recorded onto the disk. The low cost and relatively large capacity makes the CD-ROM ideally suited to the distribution of software. They are also ideal for the low-cost distribution of large quantities of information such as product catalogues, reference materials, conference proceedings, databases, etc. It is indispensable for the storage of multimedia where traditional textual information is supplemented with sound, music, voice, pictures, animation, and even video clips.

The limitation of the read-only format lead to the development of low-cost recordable optical disks. The compact disk recordable (CD-R) is a write-once, read-many (WORM) technology. The CD-R drive unit takes a blank optical disk and burns data onto it using a higher-powered laser. This disk can then be read and distributed as an ordinary CD-ROM, with the advantage that the data is non-volatile i.e. permanent. The rapid drop in the cost of drive units and blank recording media (less than R2 per CD-R) is making this a very competitive technology for data backup and small-scale data distribution.

Although the 650 megabytes initially seemed almost limitless, many multimedia and video applications now require more storage. A new format, the Digital Video Data (DVD) standard increased the capacity of the CD-ROM by providing high-density, double-sided and double-layered CDs. By combining the increased storage capacity with sophisticated data compression algorithms, a DVD disc can easily store 10 times as much as a CD, sufficient for a full-length high-quality digital motion picture with many simultaneous sound tracks.

Even the DVD is not sufficient storage capacity and currently two optical technologies have been developed to increase storage capacity even further. The basic specification of both HD-DVD and Blu-Ray provide for more than 25 GB of storage on a disc although multi-layer Blu-Ray discs with capacities of more than 200 GB have already been developed.

A promising research area involves the use of holographic disk storage whereby data is stored in a three-dimensional manner. Though in its infancy, early prototypes promise a many-fold increase in storage capacity and it could become the answer to the ever increase storage requirements of the next decade

Figure 3: Comparison of secondary storage devices

Device	Access Speed	Capacity	Cost
RAM	< 2 nanosec	256 MB (chip)	<R1/MB
Tape	serial only	500 MB-4 GB	<10c/MB
Diskette (3 1/2″)	300 ms	1,44 MB	R1/MB
PC hard disk	10 ms	40-750 GB	<2c/MB
M/F hard disk	25 ms	100+ GB	R2/MB
CD-ROM	<100 ms	660 MB	<0.1c/MB
CD-R	<100 ms	660 MB	<0.2c/MB
DVD	<100 ms	8 GB	<0.1c/MB
HD-DVD	<100 ms	30 GB	?
Blu-Ray	<100 ms	25 GB-200GB	?

Output Devices

The final stage of information processing involves the use of output devices to transform computer-readable data back into an information format that can be processed by humans. As with input devices, when deciding on an output device you need to consider what sort of information is to be displayed, and who is intended to receive it.

One distinction that can be drawn between output devices is that of hardcopy versus softcopy devices. Hardcopy devices (printers) produce a tangible and permanent output whereas softcopy devices (display screens) present a temporary, fleeting image.

Display screens

The desk-based computer screen is the most popular output device. The standard monitor works on the same principle as the normal TV tube: a “ray” gun fires electrically charged particles onto a specially coated tube (hence the name Cathode-Ray Tube or CRT). Where the particles hit the coating, the “coating” is being “excited” and emits light. A strong magnetic field guides the particle stream to form the text or graphics on your familiar monitor.

CRTs vary substantially in size and resolution. Screen size is usually measured in inches diagonally across from corner to corner and varies from as little as 12 or 14 inches for standard PCs, to as much as 40+ inches for large demonstration and video-conferencing screens. The screen resolution depends on a number of technical factors.

A technology that has received much impetus from the fast-growing laptop and notebook market is the *liquid crystal display* (LCD). LCDs have matured quickly, increasing in resolution, contrast, and colour quality. Their main advantages are lower energy requirements and their thin, flat size. Although alternative technologies are already being explored in research laboratories, they currently dominate the “flat display” market.

Organic light-emitting diodes (OLED) can generate brighter and faster images than LED technology, and require thinner screens, but they have less stable colour characteristics, making them more suitable for cellular telephone displays than for computers.

Another screen-related technology is the *video projection unit*. Originally developed for the projection of video films, the current trend towards more portable LCD-based lightweight projectors is fuelled by the needs of computer-driven public presentations. Today's units fit easily into a small suitcase and project a computer presentation in very much the same way a slide projector shows a slide presentation. They are rapidly replacing the flat transparent LCD panels that needed to be placed on top of an overhead projection unit. Though the LCD panels are more compact, weigh less and are much cheaper, their image is generally of much poorer quality and less bright.

Printers and plotters

Printers are the most popular output device for producing permanent, paper-based computer output. Although they are all hardcopy devices, a distinction can be made between impact and non-impact printers. With impact printers, a hammer or needle physically hits an inked ribbon to leave an ink impression of the desired shape on the paper. The advantage of the impact printer is that it can produce more than one simultaneous copy by using carbon or chemically-coated paper. Non-impact printers, on the other hand, have far fewer mechanically moving parts and are therefore much quieter and tend to be more reliable.

The following are the main types of printers currently in use.

- *Dot-matrix printers* used to be the familiar low-cost printers connected to many personal computers. The print head consists of a vertical row of needles each of which is individually controlled by a magnet. As the print head moves horizontally across the paper, the individual needles strike the paper (and ribbon in between) as directed by the control mechanism to produce text characters or graphics. A close inspection of a dot-matrix printout will reveal the constituent dots that make up the text. Although it is one of the cheapest printer options, its print quality is generally much lower that that of laser and ink-jet printers. However, today's models are quick and give a much better quality by increasing the number of needles.
- *Laser printers* are quickly growing in market share. They work on the same principle as the photocopier. A laser beam, toggled on and off very quickly, illuminates selected areas on a photo-sensitive drum, where the light is converted into electrical charge. As the drum rotates into a "bed" of carbon particles ("toner") with the opposite charge, these particles will adhere to the drum. The blank paper is then pressed against the drum so that the particles "rub off onto the paper sheet. The sheet then passes through a high-temperature area so that the carbon particles are permanently fused onto the paper. Current high-end laser printers can cope with extremely large printing volumes, as is required e.g. by banks to print their millions of monthly account statements. The laser technology continues to develop in tandem with photocopier technology. Laser printers can now handle colour printing, double-sided printing or combine with mail equipment to perforate, fold, address and seal automatically into envelopes. At the lower end of the scale are the low-cost "personal" laser printers, which give a very good printing quality at a relatively modest cost.
- *Thermal printers* use heat to print. The older thermal printers used heat-sensitive paper, similar to the special fax paper. A slight heat or pressure will leave a darker area. This produced very cheap but low-quality output. Currently, thermal-printing technology is used mainly for high-quality color printing. These new thermal printers use colored wax sticks and melt the wax onto the paper. Although they are slower than competing color laser and inkjet technologies, they give a much more vibrant, color-saturated image.
- *Inkjet printers* are probably the most popular low-cost printing technology. Liquid ink is squirted onto the paper in the form of tiny droplets. These printers are about the same price as dot-matrix printers, albeit more expensive in terms of consumables. Their quality is close to that of the laser printers. Their great advantage is that the printers can easily be adapted to use coloured ink, thus making popular color printers.

- *Plotters* are mainly used for engineering and architectural drawings. A plotter consists of one (or several—in the case of color plotters) pen(s) affixed to an arm. As the arm moves across the sheet of paper, the pen draws lines onto the paper. It is ideal for line drawings such as plans, especially in cases where the paper size exceeds that which can be accommodated by the other types of printers.
- *Chain and line printers* are still popular in mainframe environments for the quick production of large volumes of internal printing. The line printer consists of a horizontal, rotating "drum" with 132 cylinders,

each containing a full character set. As the 132-column wide paper moves up past the drum, a line at a time, each one of the 132 hammers on the other side of the paper strikes at the exact moment that the corresponding cylinder “shows” the correct character. The hammer hits the drum (and ink ribbon) and leaves an imprint of the character on the paper. The chain printer works on the same principle, but uses a horizontally rotating chain with engraved characters, instead of a drum. As anyone with some working experience in a large organization knows, the print quality of these “computer printouts” is not very high.

Figure 4 compares the various output devices in terms of a number of characteristics.

Figure 4: Comparison of output devices

Device	Technology	Quality	Speed	Duplicates?	Graphics?	Fonts?	Colour?
CRT	softcopy	high	very fast	n/a	yes	yes	yes
LCD	softcopy	fair	very fast	n/a	yes	yes	yes
Plotter	hardcopy	fair	slow	no	yes	yes	yes
Chain/line printer	hardcopy	low	very fast	yes	no	no	no
Laser printer	hardcopy	high	fast/fair	no	yes	yes	yes
Dot-Matrix printer	hardcopy	fair	fast/fair	yes	yes	yes	some
Inkjet printer	hardcopy	good	fair	no	yes	yes	yes

Audio-output devices

A type of output that is becoming increasingly popular different types of audio output. is audio output. There are many different types of audio output.

- *Sound output* is required by most multimedia applications and sophisticated games. The sound card in many of today’s personal computers synthesizes sound by drawing from a library of stored sounds, essentially using the same process as found in music keyboards. More advanced multimedia workstations are equipped for full stereo multi-channel surround sound and easily surpass many a modern hi-fi system in cabling and speaker complexity.
- *MIDI in/output*. Modern day music production would be impossible without a vast array of electronic instruments and keyboards. These are typically controlled by a personal computer by means of Musical Instrument Digital Interface (MIDI), a common standard for linking, controlling and processing electronic music.
- *Speech synthesis* is the production of speech-like output using an artificial voice. Although the lack of intonation still makes the voice sound artificial, the technology is reasonably mature and can be found anywhere from talking clocks and luxury cars to automated responses for telephonic directory enquiries.

Other Output Devices

Many other, extremely specialized input and output devices have been developed. Process control, for example, is a very specialized field but extremely important for automated factories (car manufacturing, canneries), continuous process environments (nuclear plants, refineries) or hazardous places (microbiological research laboratories, space exploration). For these applications, the computer relies on a multitude of sensors for its inputs: temperatures, speed, pressure, flow rates, weight, position, … These sensor inputs are then processed by the computers, which in turn control directly robot arms and other mechanical devices such as cutters, welding equipment, valves, switches, mixers etc.

South African Perspective

A number of car manufacturers have introduced new model vehicles that optionally includes a vehicle safety system that could reduce road deaths and injuries by foreseeing an unavoidable collision and activating passenger restraint and protection systems before it happens. “Pre-crash safety” has three elements:

- A sensor uses millimeter-wave radar to detect vehicles and obstacles on the road ahead.
- An electronic control unit (ECU) determines whether a collision is imminent based on the position, speed and course of the object. If it is…
- The seat belts retract to pull the passengers back into their seats and emergency brake assistance pressure is built, ready for the driver to hit the pedal.

Until now, vehicle safety devices have only been able to activate after a collision.

The car’s radar, Toyota says, works even in rain and snow and is constantly scanning ahead. Newly developed computer software can quickly determine whether a collision is imminent based on the expected course of the host vehicle as well as the position, speed and expected course of preceding or oncoming vehicles. This could be the solution we need for South Africa’s unacceptably high road death rate – all we need is for every South African driver to be able to afford the new Toyota!

Beyond the Basics

Commercial development is set to begin on the next generation of memory: the samarium cube. This technology will allow the storage of up to one terabyte (1000 gigabytes) of data in a cubic centimeter of glass. When an extremely short pulse of laser light is applied to a piece of glass containing the rare earth element samarium, a dot around 400 nanometers in diameter becomes luminous, allowing the glass to be used as an optical memory. These luminous dots can be spaced 100 nanometers apart, and up to 2000 layers of dots can be stored and read within a cubic centimeter of glass, producing a three-dimensional storage medium. The pulse of light used to irradiate the cube lasts for only 1000-trillionth of a second (a femtosecond), because a longer pulse of light will create heat that can cause the glass to crack.

Exercises

PC specifications

A friend of yours wants to buy a personal computer for her small, home-based service business. She wants to use industry-standard software to create brochures, do accounts and financial calculations and maintain a database of customers, suppliers, products and orders. She copied down the specifications for a computer that she saw advertised on TV at a competitive price, but she is not sure whether she would really need all the components, and she doesn’t understand all the technical “buzzwords”. As a knowledgeable friend, she has asked you

- to explain in non-technical terms her questions about the various components;
- to identify any obviously incorrect specifications that she might have copied down wrongly from the advertisement, and briefly explain why they are wrong.

The following is her specifications sheet:

Specification	Question	Correct?
1.7 GHz Pentium-IV	What does “1.7 GHz” mean?	
4 MB RAM	What is RAM used for?	

500 GB Hard Disk	What sort of things would be stored on the hard disk?	
X50 CD-ROM	Would I use this to make backups? If not, what would I use it for?	
32 MB SVGA Graphics card	What does this do?	
Stiffy drive	Why do I need one if I have a CD-ROM?	
102 keyboard	Should I get any other input devices as well?	
14" monitor	Is this likely to be a modern flat screen like you get on laptops, or the old fashioned sort of monitor?	
Color inkjet printer	Why not get a dot-matrix printer?	

Input/Output devices

A standard Automatic Teller Machine ("ATM") has a large number of input and output devices. List as many of its I/O devices as you can (you may include sensors as well).

Licensing & Attributions

CC licensed content, Shared previously

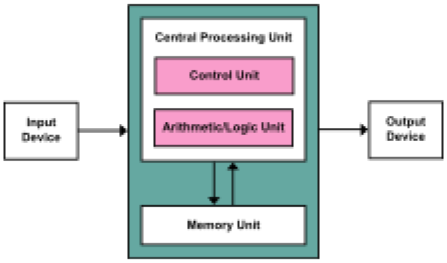
- Discovering Information Systems Chapter 4: Hardware. **Authored by:** Jean-Paul Van Belle, Mike Eccles, and Jane Nash. **Located at:** https://archive.org/details/ost-computer-science-discoveringinformationsystems. **License:** *CC BY-NC-ND: Attribution-NonCommercial-NoDerivatives*
- Mainframe computer. **Authored by:** Naotake Murayama. **Located at:** https://www.flickr.com/photos/naotakem/120643390/. **License:** *CC BY: Attribution*
- CPU wafer stack. **Authored by:** Mark Sze. **Located at:** https://www.flickr.com/photos/marksze/4231114748/. **License:** *CC BY-NC-ND: Attribution-NonCommercial-NoDerivatives*
- Computer mouse. **Authored by:** Geralt. **Located at:** https://pixabay.com/en/mouse-computer-hardware-input-160032/. **License:** *CC0: No Rights Reserved*
- Touch screen devices. **Authored by:** Geralt. **Located at:** https://pixabay.com/en/mobile-phone-smartphone-tablet-630412/. **License:** *CC0: No Rights Reserved*
- Bar code scanner. **Authored by:** OpenClipartVectors. **Located at:** https://pixabay.com/en/bar-code-scanner-bar-code-laser-155766/. **License:** *CC0: No Rights Reserved*
- Floppy Disks. **Authored by:** PublicDomainPictures. **Located at:** https://pixabay.com/en/floppy-disk-data-computer-214975/. **License:** *CC0: No Rights Reserved*
- Inkjet printer. **Authored by:** OpenClipartVectors. **Located at:** https://pixabay.com/en/printer-computer-hardware-inkjet-159612/. **License:** *CC0: No Rights Reserved*
- VHS tapes. **Authored by:** wongpear. **Located at:** https://pixabay.com/en/vhs-video-tapes-recording-398740/. **License:** *CC0: No Rights Reserved*
- RAM. **Authored by:** lrcfree. **Located at:** https://pixabay.com/en/ram-technology-pc-computer-683250/. **License:** *CC0: No Rights Reserved*

READING: COMPUTER HARDWARE

Computer hardware (usually simply called **hardware** when a computing context is concerned) is the collection of physical elements that constitutes a computer system. Computer hardware is the physical parts or components of a computer, such as the monitor, mouse, keyboard,computer data storage, hard disk drive (HDD), graphic cards, sound cards, memory, motherboard, and so on, all of which are physical objects that are tangible. In contrast, software is instructions that can be stored and run by hardware.

Software is any set of machine-readable instructions that directs a computer's processor to perform specific operations. A combination of hardware and software forms a usable computing system.

Von Neumann architecture



Von Neumann architecture scheme.

The template for all modern computers is the Von Neumann architecture, detailed in a 1945 paper by Hungarian mathematician John von Neumann. This describes a design architecture for an electronic digital computer with subdivisions of a processing unit consisting of an arithmetic logic unit and processor registers, a control unit containing an instruction register and program counter, a memory to store both data and instructions, external mass storage, and input and output mechanisms.[3] The meaning of the term has evolved to mean a stored-program computer in which an instruction fetch and a data operation cannot occur at the same time because they share a common bus. This is referred to as the Von Neumann bottleneck and often limits the performance of the system.

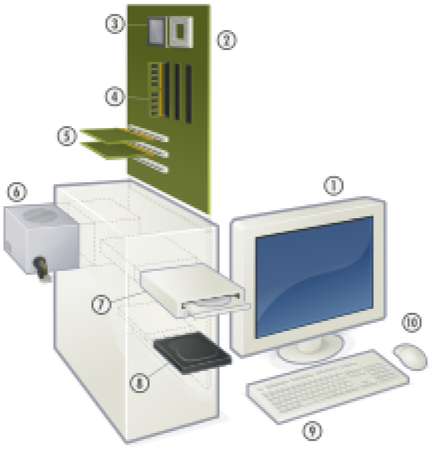
Sales

For the third consecutive year, U.S. business-to-business channel sales (sales through distributors and commercial resellers) increased, ending 2013 up nearly 6 percent at $61.7 billion. The impressive growth was the fastest sales increase since the end of the recession. Sales growth accelerated in the second half of the year peaking in fourth quarter with a 6.9 percent increase over the fourth quarter of 2012.

Different systems

There are a number of different types of computer system in use today.

Personal computer



Hardware of a modern personal computer:
1. Monitor 2.Motherboard 3.CPU 4. RAM 5.Expansion cards6. Power supply 7.Optical disc drive8. Hard disk drive9. Keyboard 10.Mouse.

The personal computer, also known as the PC, is one of the most common types of computer due to its versatility and relatively low price. Laptops are generally very similar, although may use lower-power or reduced size components.

Case

The computer case is a plastic or metal enclosure that houses most of the components. Those found on desktop computers are usually small enough to fit under a desk, however in recent years more compact designs have become more common place, such as the all-in-one style designs from Apple, namely the iMac. Though a case can basically be big or small, what matters more is which form factor of motherboard it’s designed for. Laptops are computers that usually come in a clamshell form factor, again however in more recent years deviations from this form factor have started to emerge such as laptops that have a detachable screen that become tablet computers in their own right.

Inside a custom-built computer: power supply at the bottom has its own cooling fan.

Power supply

A power supply unit (PSU) converts alternating current (AC) electric power to low-voltage DC power for the internal components of the computer. Laptops are capable of running from a built-in battery, normally for a period of hours.

Mainboard

The motherboard is the main component of a computer. It is a large rectangular board with integrated circuitry that connects the other parts of the computer including the CPU, the RAM, the disk drives(CD, DVD, hard disk, or any others) as well as any peripherals connected via the ports or the expansion slots.

Components directly attached to or part of the motherboard include:

- The CPU (Central Processing Unit) performs most of the calculations which enable a computer to function, and is sometimes referred to as the “brain” of the computer. It is usually cooled by a heat sink and fan. Most newer CPUs include an on-die Graphics Processing Unit (GPU).
- The Chipset, which includes the north bridge, mediates communication between the CPU and the other components of the system, including main memory.
- The Random-Access Memory (RAM) stores the code and data that are being actively accessed by the CPU.
- The Read-Only Memory (ROM) stores the BIOS that runs when the computer is powered on or otherwise begins execution, a process known as Bootstrapping, or “booting” or “booting up”. The BIOS (Basic Input Output System) includes boot firmware and power management firmware. Newer motherboards use Unified Extensible Firmware Interface (UEFI) instead of BIOS.
- Buses connect the CPU to various internal components and to expand cards for graphics and sound.
- The CMOS battery is also attached to the motherboard. This battery is the same as a watch battery or a battery for a remote to a car’s central locking system. Most batteries are CR2032, which powers the memory for date and time in the BIOS chip.

Expansion cards

An expansion card in computing is a printed circuit board that can be inserted into an expansion slot of a computer motherboard or backplane to add functionality to a computer system via the expansion bus. Expansions cards can be used to obtain or expand on features not offered by the motherboard.

Storage devices

Computer data storage, often called storage or memory, refers to computer components and recording media that retain digital data. Data storage is a core function and fundamental component of computers. The price of solid-state drives (SSD), which store data on flash memory, has dropped a lot in recent years, making them a better choice than ever to add to a computer to make booting up and accessing files faster.

- Fixed media
  - Data is stored by a computer using a variety of media. Hard disk drives are found in virtually all older computers, due to their high capacity and low cost, but solid-state drives are faster and more power efficient, although currently more expensive than hard drives, so are often found in more expensive computers. Some systems may use a disk array controller for greater performance or reliability.
- Removable media
  - To transfer data between computers, a USB flash drive or Optical disc may be used. Their usefulness depends on being readable by other systems; the majority of machines have an optical disk drive, and virtually all have a USB port.

Input and output peripherals

Input and output devices are typically housed externally to the main computer chassis. The following are either standard or very common to many computer systems.

- Input
  - Input devices allow the user to enter information into the system, or control its operation. Most personal computers have a mouse and keyboard, but laptop systems typically use a touchpad instead of a mouse. Other input devices include webcams, microphones, joysticks, and image scanners.
- Output device
  - Output devices display information in a human readable form. Such devices could include printers, speakers, monitors or a Braille embosser.

Mainframe computer

A mainframe computer is a much larger computer that typically fills a room and may cost many hundreds or thousands of times as much as a personal computer. They are designed to perform large numbers of calculations for governments and large enterprises.

An IBM System z9 mainframe.

Departmental computing

In the 1960s and 1970s more and more departments started to use cheaper and dedicated systems for specific purposes like process control and laboratory automation.

Supercomputer

A supercomputer is superficially similar to a mainframe, but is instead intended for extremely demanding computational tasks. As of November 2013, the fastest supercomputer in the world is the Tianhe-2, in Guangzhou, China.

The term supercomputer does not refer to a specific technology. Rather it indicates the fastest computers available at any given time. In mid 2011, the fastest supercomputers boasted speeds exceeding one petaflop, or 1000 trillion floating point operations per second. Super computers are fast but extremely costly so they are generally used by large organizations to execute computationally demanding tasks involving large data sets. Super computers typically run military and scientific applications. Although they cost millions of dollars, they are also being used for commercial applications where huge amounts of data must be analyzed. For example, large banks employ supercomputers to calculate the risks and returns of various investment strategies, and healthcare organizations use them to analyze giant databases of patient data to determine optimal treatments for various diseases and problems incurring to the country.

Hardware upgrade

When using computer hardware, an upgrade means adding new hardware to a computer that improves its performance, adds capacity or new features. For example, a user could perform a hardware upgrade to replace the hard drive with a SSD to get a boost in performance or increase the amount of files that may be stored. Also, the user could increase the RAM so the computer may run more smoothly. The user could add a USB 3.0 expansion card in order to fully use USB 3.0 devices. Performing such hardware upgrades may be necessary for older computers to meet a programs' system requirements.

Licensing & Attributions

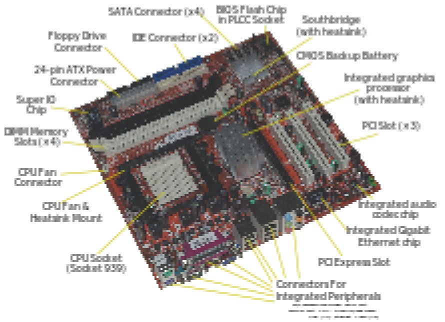
CC licensed content, Shared previously

- Computer Hardware. Provided by: Wikipedia. Located at: https://en.wikipedia.org/wiki/Computer_hardware. License: *CC BY-SA: Attribution-ShareAlike*

READING: MOTHERBOARD

A **motherboard** (sometimes alternatively known as the **mainboard**, **system board**, **planar board** or **logic board**, or colloquially, a **mobo**) is the main printed circuit board (PCB) found in computers and other expandable systems. It holds and allows communication between many of the crucial electronic components of a system, such as the central processing unit (CPU) and memory, and provides connectors for otherperipherals. Unlike a backplane, a motherboard contains significant sub-systems such as the processor and other components.

Motherboard specifically refers to a PCB with expansion capability and as the name suggests, this board is often referred to as the "mother" of all components attached to it, which often include sound cards, video cards, network cards, hard drives, or other forms of persistent storage; TV tuner cards, cards providing extra USB or FireWire slots and a variety of other custom components (the term *mainboard* is applied to devices with a single board and no additional expansions or capability, such as controlling boards in televisions, washing machines and other embedded systems).


SATA Connector (x4)
BIOS Flash Chip in PLCC Socket
Southbridge (with heatsink)
Floppy Drive Connector
IDE Connector (x2)
CMOS Backup Battery
24-pin ATX Power Connector
Integrated graphics processor (with heatsink)
Super IO Chip
DIMM Memory Slots (x4)
PCI Slot (x3)
CPU Fan Connector
CPU Fan & Heatsink Mount
Integrated audio codec chip
Integrated Gigabit Ethernet chip
CPU Socket (Socket 939)
PCI Express Slot
Connectors For Integrated Peripherals

Motherboard for an Acer desktop personal computer, showing the typical components and interfaces that are found on a motherboard. This model was made by Foxconn in 2007, and follows the ATX layout (known as the "form factor") usually employed for desktop computers. It is designed to work with AMD's Athlon 64 processor

History

Prior to the invention of the microprocessor, a digital computer consisted of multiple printed circuit boards in a card-cage case with components connected by a backplane, a set of interconnected sockets. In very old designs the wires were discrete connections between card connector pins, but printed circuit boards soon became the standard practice. The Central Processing Unit (CPU), memory, and peripheralswere housed on individual printed circuit boards, which were plugged into the backplate.

During the late 1980s and 1990s, it became economical to move an increasing number of peripheral functions onto the motherboard. In the late 1980s, personal computer motherboards began to include single ICs (also called Super I/O chips) capable of supporting a set of low-speed

Intel D945GCPE A microATX Motherboard LGA775 for Intel Pentium 4, D, XE, Dual-Core, Core 2 (circa 2007)

peripherals: keyboard, mouse, floppy disk drive, serial ports, and parallel ports. By the late-1990s, many personal computer motherboards supported a full range of audio, video, storage, and networking functions without the need for any expansion cards at all; higher-end systems for 3D gaming and computer graphics typically retained only the graphics card as a separate component.

The most popular computers such as the Apple II and IBM PC had published schematic diagrams and other documentation which permitted rapid reverse-engineering and third-party replacement motherboards. Usually intended for building new computers compatible with the exemplars, many motherboards offered additional performance or other features and were used to upgrade the manufacturer's original equipment.

Design

A motherboard provides the electrical connections by which the other components of the system communicate. Unlike a backplane, it also contains the central processing unit and hosts other subsystems and devices.

A typical desktop computer has its microprocessor, main memory, and other essential components connected to the motherboard. Other components such as external storage, controllers for video display and sound, and peripheral devices may be attached to the motherboard as plug-in cards or via cables, in modern computers it is increasingly common to integrate some of these peripherals into the motherboard itself.

The motherboard of a Samsung Galaxy SII; almost all functions of the device are integrated into a very small board

The Octek Jaguar V motherboard from 1993. This board has few onboard peripherals, as evidenced by the 6 slots provided for ISA cards and the lack of other built-in external interface connectors

An important component of a motherboard is the microprocessor's supporting chipset, which provides the supporting interfaces between the CPU and the various buses and external components. This chipset determines, to an extent, the features and capabilities of the motherboard.

Modern motherboards include:

- Sockets (or slots) in which one or more microprocessors may be installed. In the case of CPUs in ball grid array packages, such as the VIA C3, the CPU is directly soldered to the motherboard.
- Slots into which the system's main memory is to be installed (typically in the form of DIMM modules containing DRAM chips)
- A chipset which forms an interface between the CPU's front-side bus, main memory, and peripheral buses
- Non-volatile memory chips (usually Flash ROM in modern motherboards) containing the system's firmware or BIOS
- A clock generator which produces the system clock signal to synchronize the various components
- Slots for expansion cards (the interface to the system via the buses supported by the chipset)
- Power connectors, which receive electrical power from the computer power supply and distribute it to the CPU, chipset, main memory, and expansion cards. As of 2007, some graphics cards (e.g. GeForce 8 and Radeon R600) require more power than the motherboard can provide, and thus dedicated connectors have been introduced to attach them directly to the power supply.
- Connectors for hard drives, typically SATA only. Disk drives also connect to the power supply.

Additionally, nearly all motherboards include logic and connectors to support commonly used input devices, such as PS/2 connectors for a mouse and keyboard. Early personal computers such as the Apple II or IBM PC included only this minimal peripheral support on the motherboard. Occasionally video interface hardware was also integrated into the motherboard; for example, on the Apple II and rarely on IBM-compatible computers such as the IBM PC Jr. Additional peripherals such as disk controllers and serial ports were provided as expansion cards.

Given the high thermal design power of high-speed computer CPUs and components, modern motherboards nearly always include heat sinks and mounting points for fans to dissipate excess heat.

Form factor

Motherboards are produced in a variety of sizes and shapes called computer form factor, some of which are specific to individual computer manufacturers. However, the motherboards used in IBM-compatible systems are designed to fit various case sizes. As of 2007, most desktop computer motherboards use the ATX standard form factor — even those found in Macintosh and Sun computers, which have not been built from commodity components. A case's motherboard and PSU form factor must all match, though some smaller form factor motherboards of the same family will fit larger cases. For example, an ATX case will usually accommodate a microATX motherboard.

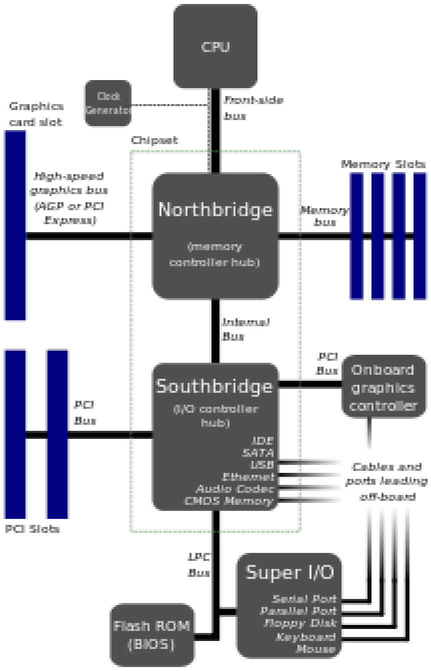
Laptop computers generally use highly integrated, miniaturized and customized motherboards. This is one of the reasons that laptop computers are difficult to upgrade and expensive to repair. Often the failure of one laptop component requires the replacement of the entire motherboard, which is usually more expensive than a desktop motherboard due to the large number of integrated components.

CPU sockets

A CPU socket (central processing unit) or slot is an electrical component that attaches to a Printed Circuit Board (PCB) and is designed to house a CPU (also called a microprocessor). It is a special type of integrated circuit socket designed for very high pin counts. A CPU socket provides many functions, including a physical structure to support the CPU, support for a heat sink, facilitating replacement (as well as reducing cost), and most importantly, forming an electrical interface both with the CPU and the PCB. CPU sockets on the motherboard can most often be found in most desktop and server computers (laptops typically use surface mount CPUs), particularly those based on the Intel x86 architecture. A CPU socket type and motherboard chipset must support the CPU series and speed.

Integrated peripherals

With the steadily declining costs and size of integrated circuits, it is now possible to include support for many peripherals on the motherboard. By combining many functions on one PCB, the physical size and total cost of the system may be reduced; highly integrated motherboards are thus especially popular in small form factor and budget computers.



- Disk controllers for a floppy disk drive, up to 2 PATA drives, and up to 6 SATA drives (including RAID 0/1 support)
- integrated graphics controller supporting 2D and 3D graphics, with VGA and TV output
- integrated sound card supporting 8-channel (7.1) audio and S/PDIF output
- Fast Ethernet network controller for 10/100 Mbit networking
- USB 2.0 controller supporting up to 12 USB ports
- IrDA controller for infrared data communication (e.g. with an IrDA-enabled cellular phone or printer)
- Temperature, voltage, and fan-speed sensors that allow software to monitor the health of computer components.

Block diagram of a modern motherboard, which supports many on-board peripheral functions as well as several expansion slots

Peripheral card slots

A typical motherboard will have a different number of connections depending on its standard and form factor.

A standard, modern ATX motherboard will typically have two or three PCI-Express 16x connection for a graphics card, one or two legacy PCI slots for various expansion cards, and one or two PCI-E 1x (which has superseded PCI). A standard EATX motherboard will have two to four PCI-E 16x connection for graphics cards, and a varying number of PCI and PCI-E 1x slots. It can sometimes also have a PCI-E 4x slot (will vary between brands and models).

Some motherboards have two or more PCI-E 16x slots, to allow more than 2 monitors without special hardware, or use a special graphics technology called SLI (for Nvidia) and Crossfire (for AMD). These allow 2 to 4 graphics cards to be linked together, to allow better performance in intensive graphical computing tasks, such as gaming, video editing, etc.

Temperature and reliability

A motherboard of a Vaio E series laptop

Motherboards are generally air cooled with heat sinks often mounted on larger chips, such as the Northbridge, in modern motherboards. Insufficient or improper cooling can cause damage to the internal components of the computer, or cause it to crash. Passive cooling, or a single fan mounted on the power supply, was sufficient for many desktop computer CPU’s until the late 1990s; since then, most have requiredCPU fans mounted on their heat sinks, due to rising clock speeds and power consumption. Most motherboards have connectors for additional case fans and integrated temperature sensors to detect motherboard and CPU temperatures and controllable fan connectors which the BIOS or operating system can use to regulate fan speed. Alternatively computers can use a water cooling system instead of many fans.

Some small form factor computers and home theater PCs designed for quiet and energy-efficient operation boast fan-less designs. This typically requires the use of a low-power CPU, as well as careful layout of the motherboard and other components to allow for heat sink placement.

A 2003 study found that some spurious computer crashes and general reliability issues, ranging from screen image distortions to I/O read/write errors, can be attributed not to software or peripheral hardware but to aging capacitors on PC motherboards. Ultimately this was shown to be the result of a faulty electrolyte formulation, an issue termed capacitor plague.

Motherboards use electrolytic capacitors to filter the DC power distributed around the board. These capacitors age at a temperature-dependent rate, as their water based electrolytes slowly evaporate. This can lead to loss of capacitance and subsequent motherboard malfunctions due to voltage instabilities. While most capacitors are rated for 2000 hours of operation at 105 °C (221 °F), their expected design life roughly doubles for every 10 °C (50 °F) below this. At 45 °C (113 °F) a lifetime of 15 years can be expected. This appears reasonable for a computer motherboard. However, many manufacturers deliver substandard capacitors, which significantly reduce life expectancy. Inadequate case cooling and elevated temperatures easily exacerbate this problem. It is possible, but time-consuming, to find and replace failed capacitors on personal computer motherboards.

Air pollution and reliability

High rates of motherboard failures in China and India appear to be due to “sulfurous air pollution produced by coal that’s burned to generate electricity. Air pollution corrodes the circuitry, according to Intel researchers.

Bootstrapping using the Basic input output system

Motherboards contain some non-volatile memory to initialize the system and load some startup software, usually an operating system, from some external peripheral device. Microcomputers such as the Apple II and IBM PC used ROM chips mounted in sockets on the motherboard. At power-up, the central processor would load its program counter with the address of the boot ROM and start executing instructions from the ROM. These instructions initialized and tested the system hardware, displayed system information on the screen, performed RAM checks, and then loaded an initial program from an external or peripheral device . If none was available, then the computer would perform tasks from other memory stores or display an error message, depending on the model and design of the computer and the ROM version. For example, both the Apple II and the original IBM PC had Microsoft Cassette BASIC in ROM and would start that if no program could be loaded from disk.

Most modern motherboard designs use a BIOS, stored in an EEPROM chip soldered to or socketed on the motherboard, to booting an operating system. Non-operating system boot programs are still supported on modern IBM PC-descended machines, but nowadays it is assumed that the boot program will be a complex operating system such as MS Windows NT or Linux. When power is first supplied to the motherboard, the BIOS firmware tests and configures memory, circuitry, and peripherals. This Power-On Self Test (POST) may include testing some of the following things:

- Video adapter
- Cards inserted into slots, such as conventional PCI
- Floppy drive
- Temperatures, voltages, and fan speeds for hardware monitoring
- CMOS used to store BIOS setup configuration
- Keyboard and Mouse
- Network controller
- Optical drives: CD-ROM or DVD-ROM
- SCSI hard drive
- IDE, EIDE, or SATA Hard disk
- Security devices, such as a fingerprint reader or the state of a latching switch to detect intrusion
- USB devices, such as a memory storage device

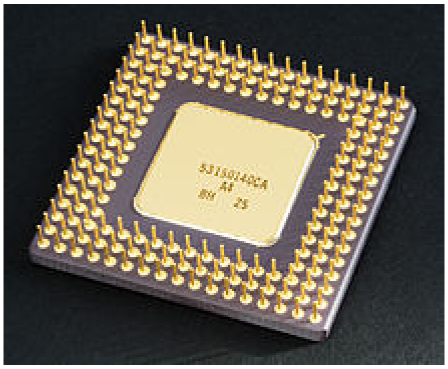
On recent motherboards, the BIOS may also patch the central processor microcode if the BIOS detects that the installed CPU is one for which errata have been published.

Licensing & Attributions

CC licensed content, Shared previously

- Motherboard. Provided by: Wikipedia. Located at: https://en.wikipedia.org/wiki/Motherboard. License: *CC BY-SA: Attribution-ShareAlike*

READING: THE CENTRAL PROCESSING UNIT



Bottom side of an Intel 80486DX2

An Intel 80486DX2 CPU, as seen from above

A **central processing unit** (**CPU**) is the electronic circuitry within a computer that carries out the instructions of a computer program by performing the basic arithmetic, logical, control and input/output (I/O) operations specified by the instructions. The term has been used in the computer industry at least since the early 1960s. Traditionally, the term "CPU" refers to a processor, more specifically to its processing unit and control unit (CU), distinguishing these core elements of a computer from external components such as main memory and I/O circuitry.

The form, design and implementation of CPUs have changed over the course of their history, but their fundamental operation remains almost unchanged. Principal components of a CPU include the arithmetic logic unit (ALU) that performs arithmetic and logic operations, processor registers that supply operands to the ALU and store the results of ALU operations, and a control unit that fetches instructions from memory and "executes" them by directing the coordinated operations of the ALU, registers and other components.

Most modern CPUs are microprocessors, meaning they are contained on a single integrated circuit (IC) chip. An IC that contains a CPU may also contain memory, peripheral interfaces, and other components of a computer; such integrated devices are variously calledmicrocontrollers or systems on a chip (SoC). Some computers employ a multi-core processor, which is a single chip containing two or more CPUs called "cores"; in that context, single chips are sometimes referred to as "sockets". Array processors or vector processors have multiple processors that operate in parallel, with no unit considered central.

History

EDVAC, one of the first stored-program computers

Computers such as the ENIAC had to be physically rewired to perform different tasks, which caused these machines to be called "fixed-program computers". Since the term "CPU" is generally defined as a device for software (computer program) execution, the earliest devices that could rightly be called CPUs came with the advent of the stored-program computer.

The idea of a stored-program computer was already present in the design of J. Presper Eckert and John William Mauchly's ENIAC, but was initially omitted so that it could be finished sooner. On June 30, 1945, before ENIAC was made, mathematician John von Neumann distributed the paper entitled *First Draft of a Report on the EDVAC*. It was the outline of a stored-program computer that would eventually be completed in August 1949. EDVAC was designed to perform a certain number of instructions (or operations) of various types. Significantly, the programs written for EDVAC were to be stored in high-speed computer memory rather than specified by the physical wiring of the computer. This overcame a severe limitation of ENIAC, which was the considerable time and effort required to reconfigure the computer to perform a new task. With von Neumann's design, the program that EDVAC ran could be changed simply by changing the contents of the memory. EDVAC, however, was not the first stored-program computer; the Manchester Small-Scale Experimental Machine, a small prototype stored-program computer, ran its first program on 21 June 1948 and the Manchester Mark 1 ran its first program during the night of 16–17 June 1949.

Early CPUs were custom designs used as part of a larger and sometimes distinctive computer. However, this method of designing custom CPUs for a particular application has largely given way to the development of multi-purpose processors produced in large quantities. This standardization began in the era of discrete transistor mainframes and minicomputers and has rapidly accelerated with the popularization of the integrated circuit (IC). The IC has allowed increasingly complex CPUs to be designed and manufactured to tolerances on the order of nanometers. Both the miniaturization and standardization of CPUs have increased the presence of digital devices in modern life far beyond the limited application of dedicated computing machines. Modern microprocessors appear in electronic devices ranging from automobiles to cellphones, and sometimes even in children’s toys.

While von Neumann is most often credited with the design of the stored-program computer because of his design of EDVAC, and the design became known as the von Neumann architecture, others before him, such as Konrad Zuse, had suggested and implemented similar ideas. The so-called Harvard architecture of the Harvard Mark I, which was completed before EDVAC, also utilized a stored-program design using punched paper tape rather than electronic memory. The key difference between the von Neumann and Harvard architectures is that the latter separates the storage and treatment of CPU instructions and data, while the former uses the same memory space for both. Most modern CPUs are primarily von Neumann in design, but CPUs with the Harvard architecture are seen as well, especially in embedded applications; for instance, the Atmel AVR microcontrollers are Harvard architecture processors.

Relays and vacuum tubes (thermionic tubes) were commonly used as switching elements; a useful computer requires thousands or tens of thousands of switching devices. The overall speed of a system is dependent on the speed of the switches. Tube computers like EDVAC tended to average eight hours between failures, whereas relay computers like the (slower, but earlier) Harvard Mark I failed very rarely. In the end, tube-based CPUs became dominant because the significant speed advantages afforded generally outweighed the reliability problems. Most of these early synchronous CPUs ran at low clock rates compared to modern microelectronic designs (see below for a discussion of clock rate). Clock signal frequencies ranging from 100 kHz to 4 MHz were very common at this time, limited largely by the speed of the switching devices they were built with.

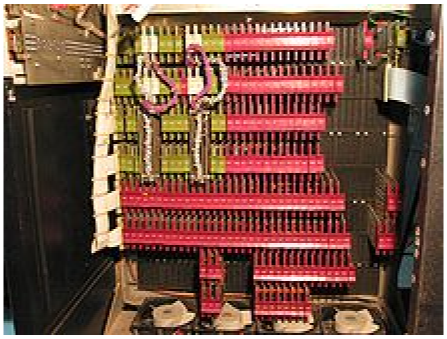
Transistor CPUs

The design complexity of CPUs increased as various technologies facilitated building smaller and more reliable electronic devices. The first such improvement came with the advent of the transistor. Transistorized CPUs during the 1950s and 1960s no longer had to be built out of bulky, unreliable, and fragile switching elements like vacuum tubes and relays. With this improvement more complex and reliable CPUs were built onto one or several printed circuit boards containing discrete (individual) components.

In 1964, IBM introduced its System/360 computer architecture that was used in a series of computers capable of running the same programs with different speed and performance. This was significant at a time when most electronic computers were incompatible with one another, even those made by the same manufacturer. To facilitate this improvement, IBM utilized the concept of a microprogram (often called “microcode”), which still sees widespread usage in modern CPUs. The System/360 architecture was so popular that it dominated the mainframe computer market for decades and left a legacy that is still continued by similar modern computers like the IBM zSeries. In 1965, Digital Equipment Corporation (DEC) introduced another influential computer aimed at the scientific and research markets, the PDP-8.

Transistor-based computers had several distinct advantages over their predecessors. Aside from facilitating increased reliability and lower power consumption, transistors also allowed CPUs to operate at much higher speeds because of the short switching time of a transistor in comparison to a tube or relay. Thanks to both the increased reliability as well as the dramatically increased speed of the switching elements (which were almost exclusively transistors by this time), CPU clock rates in the tens of megahertz were obtained during this period. Additionally while discrete transistor and IC CPUs were in heavy usage, new high-performance designs like SIMD (Single Instruction Multiple Data) vector processors began to appear. These early experimental designs later gave rise to the era of specialized supercomputers like those made by Cray Inc.

Small-scale integration CPUs



CPU, core memory, and external bus interface of a DEC PDP-8/I. Made of medium-scale integrated circuits.

During this period, a method of manufacturing many interconnected transistors in a compact space was developed. The integrated circuit (IC) allowed a large number of transistors to be manufactured on a single semiconductor-based die, or “chip”. At first only very basic non-specialized digital circuits such as NOR gates were miniaturized into ICs. CPUs based upon these “building block” ICs are generally referred to as “small-scale integration” (SSI) devices. SSI ICs, such as the ones used in the Apollo guidance computer, usually contained up to a few score transistors. To build an entire CPU out of SSI ICs required thousands of individual chips, but still consumed much less space and power than earlier discrete transistor designs.

IBM’s System/370 follow-on to the System/360 used SSI ICs rather than Solid Logic Technology discrete-transistor modules. DEC’s PDP-8/I and KI10 PDP-10 also switched from the individual transistors used by the PDP-8 and PDP-10 to SSI ICs, and their extremely popular PDP-11line was originally built with SSI ICs but was eventually implemented with LSI components once these became practical.

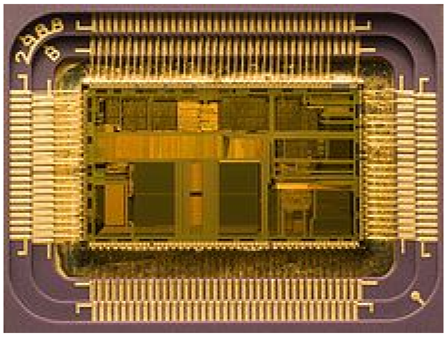
Large-scale integration CPUs

Lee Boysel published influential articles, including a 1967 “manifesto”, which described how to build the equivalent of a 32-bit mainframe computer from a relatively small number of large-scale integration circuits (LSI). At the time, the only way to build LSI chips, which are chips with a hundred or more gates, was to build them using a MOS process (i.e.,PMOS logic, NMOS logic, or CMOS logic). However, some companies continued to build processors out of bipolar chips because bipolar junction transistors were so much faster than MOS chips; for example, Datapoint built processors out of TTL chips until the early 1980s.

People building high-speed computers wanted them to be fast, so in the 1970s they built the CPUs from small-scale integration (SSI) and medium-scale integration (MSI) 7400 seriesTTL gates. At the time, MOS ICs were so slow that they were considered useful only in a few niche applications that required low power.

As the microelectronic technology advanced, an increasing number of transistors were placed on ICs, decreasing the quantity of individual ICs needed for a complete CPU. MSI and LSI ICs increased transistor counts to hundreds, and then thousands. By 1968, the number of ICs required to build a complete CPU had been reduced to 24 ICs of eight different types, with each IC containing roughly 1000 MOSFETs. In stark contrast with its SSI and MSI predecessors, the first LSI implementation of the PDP-11 contained a CPU composed of only four LSI integrated circuits.

Microprocessors



Die of an Intel 80486DX2microprocessor (actual size: 12×6.75 mm) in its packaging

Intel Core i5 CPU on a Vaio E series laptop motherboard (on the right, beneath the heat pipe)

In the 1970s the fundamental inventions by Federico Faggin (Silicon Gate MOS ICs with self-aligned gates along with his new random logic design methodology) changed the design and implementation of CPUs forever. Since the introduction of the first commercially available microprocessor (the Intel 4004) in 1970, and the first widely used microprocessor (the Intel 8080) in 1974, this class of CPUs has almost completely overtaken all other central processing unit implementation methods. Mainframe and minicomputer manufacturers of the time launched proprietary IC development programs to upgrade their older computer architectures, and eventually produced instruction set compatible microprocessors that were backward-compatible with their older hardware and software. Combined with the advent and eventual success of the ubiquitous personal computer, the term *CPU* is now applied almost exclusively to microprocessors. Several CPUs (denoted *cores*) can be combined in a single processing chip.

Previous generations of CPUs were implemented as discrete components and numerous small integrated circuits (ICs) on one or more circuit boards. Microprocessors, on the other hand, are CPUs manufactured on a very small number of ICs; usually just one. The overall smaller CPU size, as a result of being implemented on a single die, means faster switching time because of physical factors like decreased gate parasitic capacitance. This has allowed synchronous microprocessors to have clock rates ranging from tens of megahertz to several gigahertz. Additionally, as the ability to construct exceedingly small transistors on an IC has increased, the complexity and number of transistors in a single CPU has increased many fold. This widely observed trend is described by Moore's law, which has proven to be a fairly accurate predictor of the growth of CPU (and other IC) complexity.

While the complexity, size, construction, and general form of CPUs have changed enormously since 1950, it is notable that the basic design and function has not changed much at all. Almost all common CPUs today can be very accurately described as von Neumann stored-program machines. As the aforementioned Moore's law continues to hold true, concerns have arisen about the limits of integrated circuit transistor technology. Extreme miniaturization of electronic gates is causing the effects of phenomena like electromigration and subthreshold leakage to become much more significant. These newer concerns are among the many factors causing researchers to investigate new methods of computing such as the quantum computer, as well as to expand the usage of parallelism and other methods that extend the usefulness of the classical von Neumann model.

Operation

The fundamental operation of most CPUs, regardless of the physical form they take, is to execute a sequence of stored instructions that is called a program. The instructions to be executed are kept in some kind of computer memory. Nearly all CPUs follow the fetch, decode and execute steps in their operation, which are collectively known as the instruction cycle.

After the execution of an instruction, the entire process repeats, with the next instruction cycle normally fetching the next-in-sequence instruction because of the incremented value in the program counter. If a jump instruction was executed, the program counter will be modified to contain the address of the instruction that was jumped to and program execution continues normally. In more complex CPUs, multiple instructions can be fetched, decoded, and executed simultaneously. This section describes what is generally referred to as the "classic RISC pipeline", which is quite common among the simple CPUs used in many electronic devices (often called microcontroller). It largely ignores the important role of CPU cache, and therefore the access stage of the pipeline.

Some instructions manipulate the program counter rather than producing result data directly; such instructions are generally called "jumps" and facilitate program behavior like loops, conditional program execution (through the use of a conditional jump), and existence of functions. In some processors, some other instructions change the state of bits in a "flags" register. These flags can be used to influence how a program behaves, since they often indicate the outcome of various operations. For example, in such processors a "compare" instruction evaluates two values and sets or clears bits in the flags register to indicate which one is greater or whether they are equal; one of these flags could then be used by a later jump instruction to determine program flow.

Fetch

The first step, fetch, involves retrieving an instruction (which is represented by a number or sequence of numbers) from program memory. The instruction's location (address) in program memory is determined by a program counter (PC), which stores a number that identifies the address of the next instruction to be fetched. After an instruction is fetched, the PC is incremented by the length of the instruction so that it will contain the address of the next instruction in the sequence. Often, the instruction to be fetched must be retrieved from relatively slow memory, causing the CPU to stall while waiting for the instruction to be returned. This issue is largely addressed in modern processors by caches and pipeline architectures (see below).

Decode

The instruction that the CPU fetches from memory determines what the CPU will do. In the decode step, performed by the circuitry known as the instruction decoder, the instruction is converted into signals that control other parts of the CPU.

The way in which the instruction is interpreted is defined by the CPU's instruction set architecture (ISA). Often, one group of bits (that is, a "field") within the instruction, called the opcode, indicates which operation is to be performed, while the remaining fields usually provide supplemental information required for the operation, such as the operands. Those operands may be specified as a constant value (called an immediate value), or as the location of a value that may be a processor register or a memory address, as determined by some addressing mode.

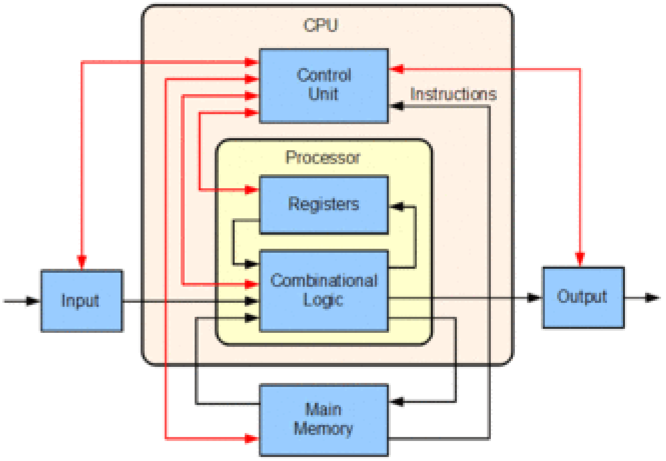
In some CPU designs the instruction decoder is implemented as a hardwired, unchangeable circuit. In others, a microprogram is used to translate instructions into sets of CPU configuration signals that are applied sequentially over multiple clock pulses. In some cases the memory that stores the microprogram is rewritable, making it possible to change the way in which the CPU decodes instructions.

Execute

After the fetch and decode steps, the execute step is performed. Depending on the CPU architecture, this may consist of a single action or a sequence of actions. During each action, various parts of the CPU are electrically connected so they can perform all or part of the desired operation and then the action is completed, typically in response to a clock pulse. Very often the results are written to an internal CPU register for quick access by subsequent instructions. In other cases results may be written to slower, but less expensive and higher capacity main memory.

For example, if an addition instruction is to be executed, the arithmetic logic unit (ALU) inputs are connected to a pair of operand sources (numbers to be summed), the ALU is configured to perform an addition operation so that the sum of its operand inputs will appear at its output, and the ALU output is connected to storage (e.g., a register or memory) that will receive the sum. When the clock pulse occurs, the sum will be transferred to storage and, if the resulting sum is too large (i.e., it is larger than the ALU's output word size), an arithmetic overflow flag will be set.

Structure and implementation



Block diagram of a basic uniprocessor-CPU computer. Black lines indicate data flow, whereas red lines indicate control flow; arrows indicate flow directions.

Hardwired into a CPU's circuitry is a set of basic operations it can perform, called an instruction set. Such operations may involve, for example, adding or subtracting two numbers, comparing two numbers, or jumping to a different part of a program. Each basic operation is represented by a particular combination of bits, known as the machine language opcode; while executing instructions in a machine language program, the CPU decides which operation to perform by "decoding" the opcode. A complete machine language instruction consists of an opcode and, in many cases, additional bits that specify arguments for the operation (for example, the numbers to be summed in the case of an addition operation). Going up the complexity scale, a machine language program is a collection of machine language instructions that the CPU executes.

The actual mathematical operation for each instruction is performed by a combinational logic circuit within the CPU's processor known as the arithmetic logic unit or ALU. In general, a CPU executes an instruction by fetching it from memory, using its ALU to perform an operation, and then storing the result to memory. Beside the instructions for integer mathematics and logic operations, various other machine instructions exist, such as those for loading data from memory and storing it back, branching operations, and mathematical operations on floating-point numbers performed by the CPU's floating-point unit (FPU).

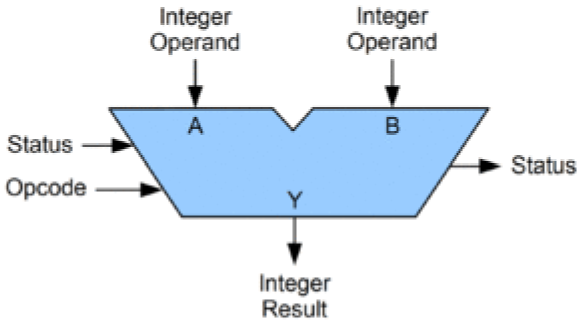
Control unit

The control unit of the CPU contains circuitry that uses electrical signals to direct the entire computer system to carry out stored program instructions. The control unit does not execute program instructions; rather, it directs other parts of the system to do so. The control unit communicates with both the ALU and memory.

Arithmetic logic unit

The arithmetic logic unit (ALU) is a digital circuit within the processor that performs integer arithmetic and bitwise logic operations. The inputs to the ALU are the data words to be operated on (called operands), status information from previous operations, and a code from the control unit indicating which operation to perform. Depending on the instruction being executed, the operands may come from internal CPU registers or external memory, or they may be constants generated by the ALU itself.

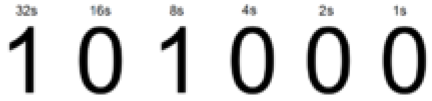
When all input signals have settled and propagated through the ALU circuitry, the result of the performed operation appears at the ALU's outputs. The result consists of both a data word, which may be stored in a register or memory, and status information that is typically stored in a special, internal CPU register reserved for this purpose.



Symbolic representation of an ALU and its input and output signals

Integer range

Every CPU represents numerical values in a specific way. For example, some early digital computers represented numbers as familiar decimal (base 10) numeral system values, and others have employed more unusual representations such as ternary (base three). Nearly all modern CPUs represent numbers in binary form, with each digit being represented by some two-valued physical quantity such as a “high” or “low” voltage.



A six-bit word containing the binary encoded representation of decimal value 40. Most modern CPUs employ word sizes that are a power of two, for example eight, 16, 32 or 64 bits.

Related to numeric representation is the size and precision of integer numbers that a CPU can represent. In the case of a binary CPU, this is measured by the number of bits (significant digits of a binary encoded integer) that the CPU can process in one operation, which is commonly called “word size”, “bit width”, “data path width”, “integer precision”, or “integer size”. A CPU’s integer size determines the range of integer values it can directly operate on. For example, an 8-bit CPU can directly manipulate integers represented by eight bits, which have a range of 256 (2^8) discrete integer values.

Integer range can also affect the number of memory locations the CPU can directly address (an address is an integer value representing a specific memory location). For example, if a binary CPU uses 32 bits to represent a memory address then it can directly address 2^{32} memory locations. To circumvent this limitation and for various other reasons, some CPUs use mechanisms (such as bank switching) that allow additional memory to be addressed.

CPUs with larger word sizes require more circuitry and consequently are physically larger, cost more, and consume more power (and therefore generate more heat). As a result, smaller 4- or 8-bit microcontrollers are commonly used in modern applications even though CPUs with much larger word sizes (such as 16, 32, 64, even 128-bit) are available. When higher performance is required, however, the benefits of a larger word size (larger data ranges and address spaces) may outweigh the disadvantages.

To gain some of the advantages afforded by both lower and higher bit lengths, many CPUs are designed with different bit widths for different portions of the device. For example, the IBM System/370 used a CPU that was primarily 32 bit, but it used 128-bit precision inside its floating point units to facilitate greater accuracy and range in floating point numbers. Many later CPU designs use similar mixed bit width, especially when the processor is meant for general-purpose usage where a reasonable balance of integer and floating point capability is required.

Clock rate

Most CPUs are synchronous circuits, which means they employ a clock signal to pace their sequential operations. The clock signal is produced by an external oscillator circuit that generates a consistent number of pulses each second in the form of a periodic square wave. The frequency of the clock pulses determines the rate at which a CPU executes instructions and, consequently, the faster the clock, the more instructions the CPU will execute each second.

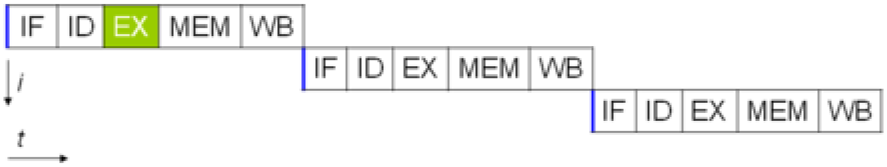
To ensure proper operation of the CPU, the clock period is longer than the maximum time needed for all signals to propagate (move) through the CPU. In setting the clock period to a value well above the worst-case propagation delay, it is possible to design the entire CPU and the way it moves data around the “edges” of the rising and falling clock signal. This has the advantage of simplifying the CPU significantly, both from a design perspective and a component-count perspective. However, it also carries the disadvantage that the entire CPU must wait on its slowest elements, even though some portions of it are much faster. This limitation has largely been compensated for by various methods of increasing CPU parallelism (see below).

However, architectural improvements alone do not solve all of the drawbacks of globally synchronous CPUs. For example, a clock signal is subject to the delays of any other electrical signal. Higher clock rates in increasingly complex CPUs make it more difficult to keep the clock signal in phase (synchronized) throughout the entire unit. This has led many modern CPUs to require multiple identical clock signals to be provided to avoid delaying a single signal significantly enough to cause the CPU to malfunction. Another major issue, as clock rates increase dramatically, is the amount of heat that is dissipated by the CPU. The constantly changing clock causes many components to switch regardless of whether they are being used at that time. In general, a component that is switching uses more energy than an element in a static state. Therefore, as clock rate increases, so does energy consumption, causing the CPU to require more heat dissipation in the form of CPU cooling solutions.

One method of dealing with the switching of unneeded components is called clock gating, which involves turning off the clock signal to unneeded components (effectively disabling them). However, this is often regarded as difficult to implement and therefore does not see common usage outside of very low-power designs. One notable recent CPU design that uses extensive clock gating is the IBM PowerPC-based Xenon used in the Xbox 360; that way, power requirements of the Xbox 360 are greatly reduced. Another method of addressing some of the problems with a global clock signal is the removal of the clock signal altogether. While removing the global clock signal makes the design process considerably more complex in many ways, asynchronous (or clockless) designs carry marked advantages in power consumption and heat dissipation in comparison with similar synchronous designs. While somewhat uncommon, entire asynchronous CPUs have been built without utilizing a global clock signal. Two notable examples of this are the ARM compliant AMULETand the MIPS R3000 compatible MiniMIPS.

Rather than totally removing the clock signal, some CPU designs allow certain portions of the device to be asynchronous, such as using asynchronous ALUs in conjunction with superscalar pipelining to achieve some arithmetic performance gains. While it is not altogether clear whether totally asynchronous designs can perform at a comparable or better level than their synchronous counterparts, it is evident that they do at least excel in simpler math operations. This, combined with their excellent power consumption and heat dissipation properties, makes them very suitable for embedded computers.

Parallelism



Model of a subscalar CPU, in which it takes fifteen clock cycles to complete three instructions.

The description of the basic operation of a CPU offered in the previous section describes the simplest form that a CPU can take. This type of CPU, usually referred to as *subscalar*, operates on and executes one instruction on one or two pieces of data at a time, that is less than one instruction per clock cycle—or more than one clock cycle per instruction (CPI > 1).

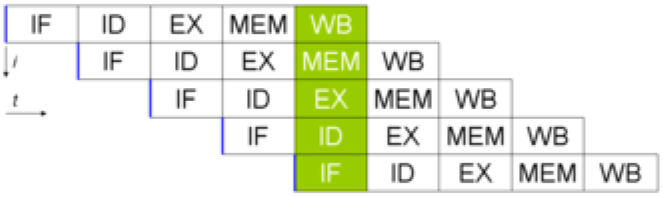
This process gives rise to an inherent inefficiency in subscalar CPUs. Since only one instruction is executed at a time, the entire CPU must wait for that instruction to complete before proceeding to the next instruction. As a result, the subscalar CPU gets “hung up” on instructions which take more than one clock cycle to complete execution. Even adding a second execution unit (see below) does not improve performance much; rather than one pathway being hung up, now two pathways are hung up and the number of unused transistors is increased. This design, wherein the CPU’s execution resources can operate on only one instruction at a time, can only possibly reach*scalar* performance (one instruction per clock cycle or CPI = 1). However, the performance is nearly always subscalar (less than one instruction per clock cycle or CPI > 1).

Attempts to achieve scalar and better performance have resulted in a variety of design methodologies that cause the CPU to behave less linearly and more in parallel. When referring to parallelism in CPUs, two terms are generally used to classify these design techniques:

- *instruction-level parallelism* (ILP), which seeks to increase the rate at which instructions are executed within a CPU (that is, to increase the utilization of on-die execution resources);
- *thread-level parallelism* (TLP), which purposes to increase the number of threads (effectively individual programs) that a CPU can execute simultaneously.

Each methodology differs both in the ways in which they are implemented, as well as the relative effectiveness they afford in increasing the CPU’s performance for an application.

Instruction-level parallelism

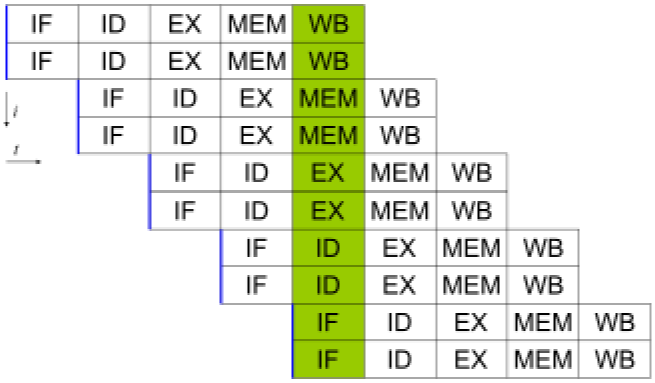


IF	ID	EX	MEM	WB				
	IF	ID	EX	MEM	WB			
		IF	ID	EX	MEM	WB		
			IF	ID	EX	MEM	WB	
				IF	ID	EX	MEM	WB

Basic five-stage pipeline. In the best case scenario, this pipeline can sustain a completion rate of one instruction per cycle.

One of the simplest methods used to accomplish increased parallelism is to begin the first steps of instruction fetching and decoding before the prior instruction finishes executing. This is the simplest form of a technique known as instruction pipelining, and is utilized in almost all modern general-purpose CPUs. Pipelining allows more than one instruction to be executed at any given time by breaking down the execution pathway into discrete stages. This separation can be compared to an assembly line, in which an instruction is made more complete at each stage until it exits the execution pipeline and is retired.

Pipelining does, however, introduce the possibility for a situation where the result of the previous operation is needed to complete the next operation; a condition often termed data dependency conflict. To cope with this, additional care must be taken to check for these sorts of conditions and delay a portion of the instruction pipeline if this occurs. Naturally, accomplishing this requires additional circuitry, so pipelined processors are more complex than subscalar ones (though not very significantly so). A pipelined processor can become very nearly scalar, inhibited only by pipeline stalls (an instruction spending more than one clock cycle in a stage).



IF	ID	EX	MEM	WB				
IF	ID	EX	MEM	WB				
	IF	ID	EX	MEM	WB			
	IF	ID	EX	MEM	WB			
		IF	ID	EX	MEM	WB		
		IF	ID	EX	MEM	WB		
			IF	ID	EX	MEM	WB	
			IF	ID	EX	MEM	WB	
				IF	ID	EX	MEM	WB
				IF	ID	EX	MEM	WB

A simple superscalar pipeline. By fetching and dispatching two instructions at a time, a maximum of two instructions per cycle can be completed.

Further improvement upon the idea of instruction pipelining led to the development of a method that decreases the idle time of CPU components even further. Designs that are said to be *superscalar* include a long instruction pipeline and multiple identical execution units. In a superscalar pipeline, multiple instructions are read and passed to a dispatcher, which decides whether or not the instructions can be executed in parallel (simultaneously). If so they are dispatched to available execution units, resulting in the ability for several instructions to be executed simultaneously. In general, the more instructions a superscalar CPU is able to dispatch simultaneously to waiting execution units, the more instructions will be completed in a given cycle.

Most of the difficulty in the design of a superscalar CPU architecture lies in creating an effective dispatcher. The dispatcher needs to be able to quickly and correctly determine whether instructions can be executed in parallel, as well as dispatch them in such a way as to keep as many execution units busy as possible. This requires that the instruction pipeline is filled as often as possible and gives rise to the need in superscalar architectures for significant amounts of CPU cache. It also makes hazard-avoiding techniques like branch prediction, speculative execution, and out-of-order execution crucial to maintaining high levels of performance. By attempting to predict which branch (or path) a conditional instruction will take, the CPU can minimize the number of times that the entire pipeline must wait until a conditional instruction is completed. Speculative execution often provides modest performance increases by executing portions of code that may not be needed after a conditional operation completes. Out-of-order execution somewhat rearranges the order in which instructions are executed to reduce delays due to data dependencies. Also in case of Single Instructions Multiple Data—a case when a lot of data from the same type has to be processed, modern processors can disable parts of the pipeline so that when a single instruction is executed many times, the CPU skips the fetch and decode phases and thus greatly increases performance on certain occasions, especially in highly monotonous program engines such as video creation software and photo processing.

In the case where a portion of the CPU is superscalar and part is not, the part which is not suffers a performance penalty due to scheduling stalls. The Intel P5 Pentium had two superscalar ALUs which could accept one instruction per clock each, but its FPU could not accept one instruction per clock. Thus the P5 was integer superscalar but not floating point superscalar. Intel's successor to the P5 architecture, P6, added superscalar capabilities to its floating point features, and therefore afforded a significant increase in floating point instruction performance.

Both simple pipelining and superscalar design increase a CPU's ILP by allowing a single processor to complete execution of instructions at rates surpassing one instruction per cycle (IPC). Most modern CPU designs are at least somewhat superscalar, and nearly all general purpose CPUs designed in the last decade are superscalar. In later years some of the emphasis in designing high-ILP computers has been moved out of the CPU's hardware and into its software interface, or ISA. The strategy of the very long instruction word (VLIW) causes some ILP to become implied directly by the software, reducing the amount of work the CPU must perform to boost ILP and thereby reducing the design's complexity.

Thread-level parallelism

Another strategy of achieving performance is to execute multiple programs or threads in parallel. This area of research is known as parallel computing. In Flynn's taxonomy, this strategy is known as Multiple instruction stream-Multiple data stream or MIMD.

One technology used for this purpose was multiprocessing (MP). The initial flavor of this technology is known as symmetric multiprocessing (SMP), where a small number of CPUs share a coherent view of their memory system. In this scheme, each CPU has additional hardware to maintain a constantly up-to-date view of memory. By avoiding stale views of memory, the CPUs can cooperate on the same program and programs can migrate from one CPU to another. To increase the number of cooperating CPUs beyond a handful, schemes such as non-uniform memory access (NUMA) and directory-based coherence protocols were introduced in the 1990s. SMP systems are limited to a small number of CPUs while NUMA systems have been built with thousands of processors. Initially, multiprocessing was built using multiple discrete CPUs and boards to implement the interconnect between the processors. When the processors and their interconnect are all implemented on a single silicon chip, the technology is known as a multi-core processor.

It was later recognized that finer-grain parallelism existed with a single program. A single program might have several threads (or functions) that could be executed separately or in parallel. Some of the earliest examples of this technology implemented input/output processing such as direct memory access as a separate thread from the computation thread. A more general approach to this technology was introduced in the 1970s when systems were designed to run multiple computation threads in parallel. This technology is known as multi-threading (MT). This approach is considered more cost-effective than multiprocessing, as only a small number of components within a CPU is replicated to support MT as opposed to the entire CPU in the case of MP. In MT, the execution units and the memory system including the caches are shared among multiple threads. The downside of MT is that the hardware support for multithreading is more visible to software than that of MP and thus supervisor software like operating systems have to undergo larger changes to support MT. One type of MT that was implemented is known as block multithreading, where one thread is executed until it is stalled waiting for data to return from external memory. In this scheme, the CPU would then quickly switch to another thread which is ready to run, the switch often done in one CPU clock cycle, such as the UltraSPARC Technology. Another type of MT is known as simultaneous multithreading, where instructions of multiple threads are executed in parallel within one CPU clock cycle.

For several decades from the 1970s to early 2000s, the focus in designing high performance general purpose CPUs was largely on achieving high ILP through technologies such as pipelining, caches, superscalar execution, out-of-order execution, etc. This trend culminated in large, power-hungry CPUs such as the Intel Pentium 4. By the early 2000s, CPU designers were thwarted from achieving higher performance from ILP techniques due to the growing disparity between CPU operating frequencies and main memory operating frequencies as well as escalating CPU power dissipation owing to more esoteric ILP techniques.

CPU designers then borrowed ideas from commercial computing markets such as transaction processing, where the aggregate performance of multiple programs, also known as throughput computing, was more important than the performance of a single thread or program.

This reversal of emphasis is evidenced by the proliferation of dual and multiple core CMP (chip-level multiprocessing) designs and notably, Intel's newer designs resembling its less superscalar P6 architecture. Late designs in several processor families exhibit CMP, including the x86-64 Opteron and Athlon 64 X2, the SPARC UltraSPARC T1, IBM POWER4 andPOWER5, as well as several video game console CPUs like the Xbox 360's triple-core PowerPC design, and the PS3's 7-core Cell microprocessor.

Data parallelism

A less common but increasingly important paradigm of CPUs (and indeed, computing in general) deals with data parallelism. The processors discussed earlier are all referred to as some type of scalar device. As the name implies, vector processors deal with multiple pieces of data in the context of one instruction. This contrasts with scalar processors, which deal with one piece of data for every instruction. Using Flynn's taxonomy, these two schemes of dealing with data are generally referred to as SIMD (single instruction, multiple data) and SISD (single instruction, single data), respectively. The great utility in creating CPUs that deal with vectors of data lies in optimizing tasks that tend to require the same operation (for example, a sum or a dot product) to be performed on a large set of data. Some classic examples of these types of tasks are multimedia applications (images, video, and sound), as well as many types of scientific and engineering tasks. Whereas a scalar CPU must complete the entire process of fetching, decoding, and executing each instruction and value in a set of data, a vector CPU can perform a single operation on a comparatively large set of data with one instruction. Of course, this is only possible when the application tends to require many steps which apply one operation to a large set of data.

Most early vector CPUs, such as the Cray-1, were associated almost exclusively with scientific research and cryptography applications. However, as multimedia has largely shifted to digital media, the need for some form of SIMD in general-purpose CPUs has become significant. Shortly after inclusion of floating point execution units started to become commonplace in general-purpose processors, specifications for and implementations of SIMD execution units also began to appear for general-purpose CPUs. Some of these early SIMD specifications like HP's Multimedia Acceleration eXtensions (MAX) and Intel's MMX were integer-only. This proved to be a significant impediment for some software developers, since many of the applications that benefit from SIMD primarily deal with floating point numbers. Progressively, these early designs were refined and remade into some of the common, modern SIMD specifications, which are usually associated with one ISA. Some notable modern examples are Intel's SSE and the PowerPC-related AltiVec (also known as VMX).

Performance

The *performance* or *speed* of a processor depends on, among many other factors, the clock rate (generally given in multiples of hertz) and the instructions per clock (IPC), which together are the factors for the instructions per second (IPS) that the CPU can perform. Many reported IPS values have represented "peak" execution rates on artificial instruction sequences with few branches, whereas realistic workloads consist of a mix of instructions and applications, some of which take longer to execute than others. The performance of the memory hierarchy also greatly affects processor performance, an issue barely considered in MIPS calculations. Because of these problems, various standardized tests, often called"benchmarks" for this purpose—such as SPECint—have been developed to attempt to measure the real effective performance in commonly used applications.

Processing performance of computers is increased by using multi-core processors, which essentially is plugging two or more individual processors (called *cores* in this sense) into one integrated circuit. Ideally, a dual core processor would be nearly twice as powerful as a single core processor. In practice, the performance gain is far smaller, only about 50%, due to imperfect software algorithms and implementation. Increasing the number of cores in a processor (i.e. dual-core, quad-core, etc.) increases the workload that can be handled. This means that the processor can now handle numerous asynchronous events, interrupts, etc. which can take a toll on the CPU when overwhelmed. These cores can be thought of as different floors in a processing plant, with each floor handling a different task. Sometimes, these cores will handle the same tasks as cores adjacent to them if a single core is not enough to handle the information.

Due to specific capabilities of modern CPUs, such as hyper-threading and uncore, which involve sharing of actual CPU resources while aiming at increased utilization, monitoring performance levels and hardware utilization gradually became a more complex task. As a response, some CPUs implement additional hardware logic that monitors actual utilization of various parts of a CPU and provides various counters accessible to software; an example is Intel's *Performance Counter Monitor* technology.

Licensing & Attributions

CC licensed content, Shared previously

- Central Processing Unit. **Provided by:** Wikipedia. **Located at:** https://en.wikipedia.org/wiki/Central_processing_unit. **License:** *CC BY-SA: Attribution-ShareAlike*

READING: CHIPSET

In a computer system, a **chipset** is a set of electronic components in an integrated circuit that manages the data flow between the processor, memory and peripherals. It is usually found on the motherboard. Chipsets are usually designed to work with a specific family of microprocessors. Because it controls communications between the processor and external devices, the chipset plays a crucial role in determining system performance.

Intel ICH7 Southbridge on Intel D945GCPE Desktop Board

Computers

In computing, the term *chipset* commonly refers to a set of specialized chips on a computer's motherboard or an expansion card. In personal computers, the first chipset for the IBM PC AT of 1984 was the NEAT chipset developed by Chips and Technologies for the Intel 80286 CPU.

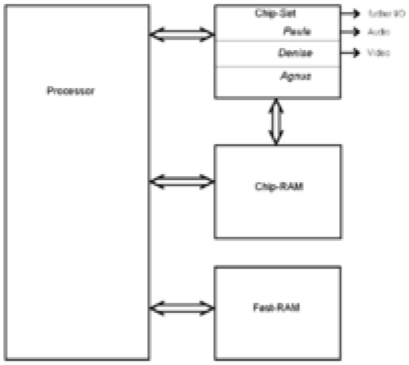


Diagram of Commodore Amiga’s Original Chip Set

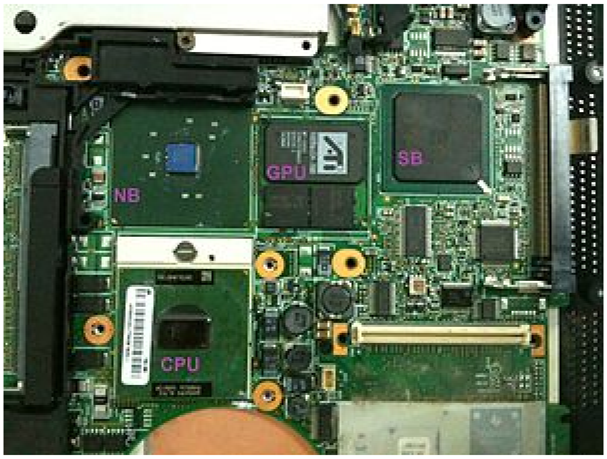
In home computers, game consoles and arcade-game hardware of the 1980s and 1990s, the term chipset was used for the custom audio and graphics chips. Examples include the Commodore Amiga’s Original Chip Set or SEGA’s System 16 chipset.

Based on Intel Pentium-class microprocessors, the term *chipset* often refers to a specific pair of chips on the motherboard: the *northbridge* and the *southbridge*. The northbridge links the CPU to very high-speed devices, especially RAM and graphics controllers, and the southbridge connects to lower-speed peripheral buses (such as PCIor ISA). In many modern chipsets, the southbridge contains some on-chip integrated peripherals, such as Ethernet, USB, and audio devices.

Motherboards and their chipsets often come from different manufacturers. As of 2015, manufacturers of chipsets for x86 motherboards include AMD, Broadcom, Intel, NVIDIA,SiS and VIA Technologies. Apple computers and Unix workstations have traditionally used custom-designed chipsets. Some server manufacturers also develop custom chipsets for their products.

In the 1980s Chips and Technologies pioneered the manufacturing of chipsets for PC-compatible computers. Computer systems produced since then often share commonly used chipsets, even across widely disparate computing specialties. For example, the NCR 53C9x, a low-cost chipset implementing a SCSI interface to storage devices, could be found inUnix machines such as the MIPS Magnum, embedded devices, and personal computers.

Move toward processor integration in PCs



A part of an IBM T42 laptop motherboard. CPU: Central processing unit. NB: Northbridge. GPU: Graphics processing unit. SB: Southbridge.

Traditionally in x86 computers, the processor’s primary connection to the rest of the machine is through the motherboard chipset’s northbridge. The northbridge is directly responsible for communications with high-speed devices (system memory and primary expansion buses, such as PCIe, AGP and PCI cards, being common examples) and conversely any system communication back to the processor. This connection between the processor and northbridge is traditionally known as the front side bus (FSB). Requests to resources not directly controlled by the northbridge are offloaded to the southbridge, with the northbridge being an intermediary between the processor and the southbridge. The southbridge traditionally handles “everything else”, generally lower-speed peripherals and board functions (the largest being hard disk and storage connectivity) such as USB, parallel and serial communications. The connection between the northbridge and southbridge does not have a common name, but is usually a high-speed interconnect proprietary to the chipset vendor.

Before 2003, any interaction between a CPU and main memory or an expansion device such as a graphics card(s)—whether AGP, PCI or integrated into the motherboard—was directly controlled by the northbridge IC on behalf of the processor. This made processor performance highly dependent on the system chipset, especially the northbridge’s memory performance and ability to shuttle this information back to the processor. In 2003, however, AMD’s introduction of the Athlon 64-bit series of processors changed this. The Athlon64 marked the introduction of an integrated memory controller being incorporated into the processor itself thus allowing the processor to directly access and handle memory, negating the need for a traditional northbridge to do so. Intel followed suit in 2008 with the release of its Core i series CPUs and the X58 platform. In newer processors integration has further increased, primarily inclusion of the system’s primary PCIe controller and integrated graphics directly on the CPU itself. As fewer functions are left un-handled by the processor, chipset vendors have condensed the remaining northbridge and southbridge functions into a single chip. Intel’s version of this is the

“Platform Controller Hub” (PCH), effectively an enhanced southbridge for the remaining peripherals as traditional northbridge duties, such as memory controller, expansion bus (PCIe) interface (though the chipset often contains secondary PCIe connections), and even on-board video controller, are integrated into the CPU itself. However, the Platform Controller Hub was integrated into the processor for certain models of Intel’s Skylake processors.

Licensing & Attributions

CC licensed content, Shared previously

- Chipset. Provided by: Wikipedia. Located at: https://en.wikipedia.org/wiki/Chipset. License: *CC BY-SA: Attribution-ShareAlike*

READING: MEMORY MANAGEMENT

https://learn.saylor.org/course/view.php?id=94&sectionid=972

Licensing & Attributions

Public domain content

- License: *Public Domain: No Known Copyright*

READING: RANDOM ACCESS MEMORY

Random-access memory (RAM /ræm/) is a form of computer data storage. A random-access memory device allows data items to be accessed (read or written) in almost the same amount of time irrespective of the physical location of data inside the memory. In contrast, with other direct-access data storage media such as hard disks, CD-RWs, DVD-RWs and the older drum memory, the time required to read and write data items varies significantly depending on their physical locations on the recording medium, due to mechanical limitations such as media rotation speeds and arm movement delays.

Example of writable volatile random-access memory: Synchronous Dynamic RAM modules, primarily used as main memory in personal computers, workstations, and servers.

Today, random-access memory takes the form of integrated circuits. RAM is normally associated with volatile types of memory (such as DRAM memory modules), where stored information is lost if power is removed, although many efforts have been made to develop non-volatile RAM chips. Other types of non-volatile memory exist that allow random access for read operations, but either do not allow write operations or have limitations on them. These include most types of ROM and a type of flash memory called *NOR-Flash*.

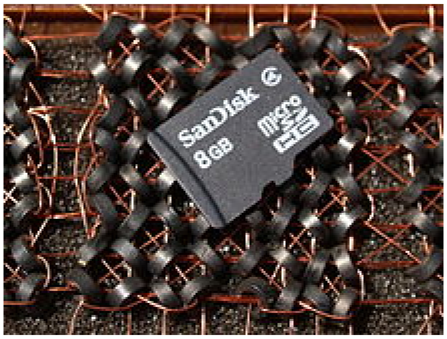
Integrated-circuit RAM chips came into the market in the late 1960s, with the first commercially available DRAM chip, the Intel 1103, introduced in October 1970.

History

These IBM tabulating machines from the 1930s used mechanical counters to store information

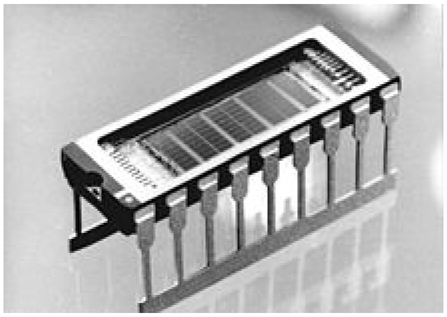
Early computers used relays, mechanical counters or delay lines for main memory functions. Ultrasonic delay lines could only reproduce data in the order it was written. Drum memory could be expanded at relatively low cost but efficient retrieval of memory items required knowledge of the physical layout of the drum to optimize speed. Latches built out of vacuum tube triodes, and later, out of discrete transistors, were used for smaller and faster memories such as registers. Such registers were relatively large and too costly to use for large amounts of data; generally only a few dozen or few hundred bits of such memory could be provided.

The first practical form of random-access memory was the Williams tube starting in 1947. It stored data as electrically charged spots on the face of a cathode ray tube. Since the electron beam of the CRT could read and write the spots on the tube in any order, memory was random access. The capacity of the Williams tube was a few hundred to around a thousand bits, but it was much smaller, faster, and more power-efficient than using individual vacuum tube latches. Developed at the University of Manchester in England, the Williams tube provided the medium on which the first electronically stored-memory program was implemented in the Manchester Small-Scale Experimental Machine (SSEM) computer, which first successfully ran a program on 21 June 1948. In fact, rather than the Williams tube memory being designed for the SSEM, the SSEM was a testbed to demonstrate the reliability of the memory.



A portion of a core memory with a modern flash RAM SD card on top

Magnetic-core memory was invented in 1947 and developed up until the mid-1970s. It became a widespread form of random-access memory, relying on an array of magnetized rings. By changing the sense of each ring’s magnetization, data could be stored with one bit stored per ring. Since every ring had a combination of address wires to select and read or write it, access to any memory location in any sequence was possible.



1 Megabit chip – one of the last models developed by VEB Carl Zeiss Jena in 1989

Magnetic core memory was the standard form of memory system until displaced by solid-state memory in integrated circuits, starting in the early 1970s. Robert H. Dennard invented dynamic random-access memory (DRAM) in 1968; this allowed replacement of a 4 or 6-transistor latch circuit by a single transistor for each memory bit, greatly increasing memory density at the cost of volatility. Data was stored in the tiny capacitance of each transistor, and had to be periodically refreshed every few milliseconds before the charge could leak away.

Prior to the development of integrated read-only memory (ROM) circuits, *permanent* (or *read-only*) random-access memory was often constructed using diode matrices driven by address decoders, or specially wound core rope memory planes.

Types of RAM

The two main forms of modern RAM are static RAM (SRAM) and dynamic RAM (DRAM). In SRAM, a bit of data is stored using the state of a six transistor memory cell. This form of RAM is more expensive to produce, but is generally faster and requires less power than DRAM and, in modern computers, is often used as cache memory for the CPU. DRAM stores a bit of data using a transistor and capacitor pair, which together comprise a DRAM memory cell. The capacitor holds a high or low charge (1 or 0, respectively), and the transistor acts as a switch that lets the control circuitry on the chip read the capacitor's state of charge or change it. As this form of memory is less expensive to produce than static RAM, it is the predominant form of computer memory used in modern computers.

Both static and dynamic RAM are considered *volatile*, as their state is lost or reset when power is removed from the system. By contrast, read-only memory (ROM) stores data by permanently enabling or disabling selected transistors, such that the memory cannot be altered. Writeable variants of ROM (such as EEPROM and flash memory) share properties of both ROM and RAM, enabling data topersist without power and to be updated without requiring special equipment. These persistent forms of semiconductor ROM include USB flash drives, memory cards for cameras and portable devices, etc. ECC memory (which can be either SRAM or DRAM) includes special circuitry to detect and/or correct random faults (memory errors) in the stored data, using parity bits or error correction code.

In general, the term *RAM* refers solely to solid-state memory devices (either DRAM or SRAM), and more specifically the main memory in most computers. In optical storage, the termDVD-RAM is somewhat of a misnomer since, unlike CD-RW or DVD-RW it does not need to be erased before reuse. Nevertheless, a DVD-RAM behaves much like a hard disc drive if somewhat slower.

Memory hierarchy

One can read and over-write data in RAM. Many computer systems have a memory hierarchy consisting of processor registers, on-die SRAM caches, external caches, DRAM, pagingsystems and virtual memory or swap space on a hard drive. This entire pool of memory may be referred to as "RAM" by many developers, even though the various subsystems can have very different access times, violating the original concept behind the *random access* term in RAM. Even within a hierarchy level such as DRAM, the specific row, column, bank,rank, channel, or interleave organization of the components make the access time variable, although not to the extent that rotating storage media or a tape is variable. The overall goal of using a memory hierarchy is to obtain the higher possible average access performance while minimizing the total cost of the entire memory system (generally, the memory hierarchy follows the access time with the fast CPU registers at the top and the slow hard drive at the bottom).

In many modern personal computers, the RAM comes in an easily upgraded form of modules called memory modules or DRAM modules about the size of a few sticks of chewing gum. These can quickly be replaced should they become damaged or when changing needs demand more storage capacity. As suggested above, smaller amounts of RAM (mostly SRAM) are also integrated in the CPU and other ICs on the motherboard, as well as in hard-drives, CD-ROMs, and several other parts of the computer system.

Other uses of RAM

In addition to serving as temporary storage and working space for the operating system and applications, RAM is used in numerous other ways.

Virtual memory

Most modern operating systems employ a method of extending RAM capacity, known as "virtual memory". A portion of the computer's hard drive is set aside for a *paging file* or a*scratch partition*, and the combination of physical RAM and the paging file form the system's total memory. (For example, if a computer has 2 GB of RAM and a 1 GB page file, the operating system has 3 GB total memory available to it.) When the system runs low on physical memory, it can "swap" portions of RAM to the paging file to make room for new data, as well as to read previously swapped information back into RAM. Excessive use of this mechanism results in thrashing and generally hampers overall system performance, mainly because hard drives are far slower than RAM.

RAM disk

Software can “partition” a portion of a computer’s RAM, allowing it to act as a much faster hard drive that is called a RAM disk. A RAM disk loses the stored data when the computer is shut down, unless memory is arranged to have a standby battery source.

Shadow RAM

Sometimes, the contents of a relatively slow ROM chip are copied to read/write memory to allow for shorter access times. The ROM chip is then disabled while the initialized memory locations are switched in on the same block of addresses (often write-protected). This process, sometimes called *shadowing*, is fairly common in both computers and embedded systems.

As a common example, the BIOS in typical personal computers often has an option called “use shadow BIOS” or similar. When enabled, functions relying on data from the BIOS’s ROM will instead use DRAM locations (most can also toggle shadowing of video card ROM or other ROM sections). Depending on the system, this may not result in increased performance, and may cause incompatibilities. For example, some hardware may be inaccessible to the operating system if shadow RAM is used. On some systems the benefit may be hypothetical because the BIOS is not used after booting in favor of direct hardware access. Free memory is reduced by the size of the shadowed ROMs.

Recent developments

Several new types of *non-volatile* RAM, which will preserve data while powered down, are under development. The technologies used include carbon nanotubes and approaches utilizing Tunnel magnetoresistance. Amongst the 1st generation MRAM, a 128 KiB (128×2^{10} bytes) chip was manufactured with 0.18 μm technology in the summer of 2003. In June 2004, Infineon Technologies unveiled a 16 MiB (16×2^{20} bytes) prototype again based on 0.18 μm technology. There are two 2nd generation techniques currently in development: thermal-assisted switching (TAS) which is being developed by Crocus Technology, and spin-transfer torque (STT) on which Crocus, Hynix, IBM, and several other companies are working. Nantero built a functioning carbon nanotube memory prototype 10 GiB (10×2^{30} bytes) array in 2004. Whether some of these technologies will be able to eventually take a significant market share from either DRAM, SRAM, or flash-memory technology, however, remains to be seen.

Since 2006, “solid-state drives” (based on flash memory) with capacities exceeding 256 gigabytes and performance far exceeding traditional disks have become available. This development has started to blur the definition between traditional random-access memory and “disks”, dramatically reducing the difference in performance.

Some kinds of random-access memory, such as “EcoRAM”, are specifically designed for server farms, where low power consumption is more important than speed.

Memory wall

The “memory wall” is the growing disparity of speed between CPU and memory outside the CPU chip. An important reason for this disparity is the limited communication bandwidth beyond chip boundaries. From 1986 to 2000, CPU speed improved at an annual rate of 55% while memory speed only improved at 10%. Given these trends, it was expected that memory latency would become an overwhelming bottleneck in computer performance.

CPU speed improvements slowed significantly partly due to major physical barriers and partly because current CPU designs have already hit the memory wall in some sense. Intel summarized these causes in a 2005 document.

> “First of all, as chip geometries shrink and clock frequencies rise, the transistor leakage current increases, leading to excess power consumption and heat… Secondly, the advantages of higher clock speeds are in part negated by memory latency, since memory access times have not been able to keep pace with increasing clock frequencies. Third, for certain applications, traditional serial architectures are becoming less efficient as processors get faster (due to the so-called Von Neumann bottleneck), further undercutting any gains that frequency increases might otherwise buy. In addition, partly due to limitations in the means of producing inductance within solid state devices, resistance-capacitance (RC) delays in signal transmission are growing as feature sizes shrink, imposing an additional bottleneck that frequency increases don’t address.”

The RC delays in signal transmission were also noted in Clock Rate versus IPC: The End of the Road for Conventional Microarchitectures which projects a maximum of 12.5% average annual CPU performance improvement between 2000 and 2014.

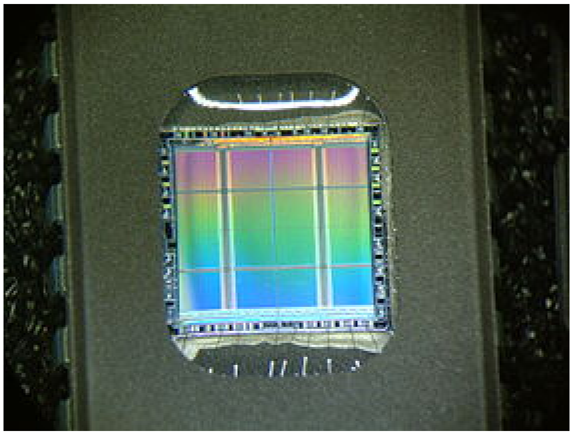
A different concept is the processor-memory performance gap, which can be addressed by 3D computer chips that reduce the distance between the logic and memory aspects that are further apart in a 2D chip. Memory subsystem design requires a focus on the gap, which is widening over time. The main method of bridging the gap is the use of caches; small amounts of high-speed memory that houses recent operations and instructions nearby the processor, speeding up the execution of those operations or instructions in cases where they are called upon frequently. Multiple levels of caching have been developed in order to deal with the widening of the gap, and the performance of high-speed modern computers are reliant on evolving caching techniques. These can prevent the loss of performance that the processor has, as it takes less time to perform the computation it has been initiated to complete. There can be up to a 53% difference between the growth in speed of processor speeds and the lagging speed of main memory access.

Licensing & Attributions

CC licensed content, Shared previously

- Random access memory. **Provided by:** Wikipedia. **Located at:** https://en.wikipedia.org/wiki/Random-access_memory. **License:** *CC BY-SA: Attribution-ShareAlike*

READING: READ-ONLY MEMORY



Read-only memory (ROM) is a class of storage medium used in computers and other electronic devices. Data stored in ROM can only be modified slowly, with difficulty, or not at all, so it is mainly used to distribute firmware (software that is very closely tied to specific hardware, and unlikely to need frequent updates).

Strictly, *read-only memory* refers to memory that is hard-wired, such as diode matrix and the later mask ROM. Although discrete circuits can be altered (in principle), integrated circuits (ICs) cannot and are useless if the data is bad. The fact that such memory can never be changed is a large drawback; more recently, *ROM* commonly refers to memory that is read-only in normal operation, while reserving the fact of some possible way to change it.

Other types of non-volatile memory such as erasable programmable read only memory (EPROM) and electrically erasable programmable read-only memory (EEPROM or Flash ROM) are sometimes referred to, in an abbreviated way, as “read-only memory” (ROM); although these types of memory can be erased and re-programmed multiple times, writing to this memory takes longer and may require different procedures than reading the memory. When used in this less precise way, “ROM” indicates a*non-volatile* memory which serves functions typically provided by mask ROM, such as storage of program code and nonvolatile data.

History

Read-only memory was used for Jacquard looms.

The simplest type of solid-state ROM is as old as the semiconductor technology itself. Combinational logic gates can be joined manually to map n-bit address input onto arbitrary values of m-bit data output (a look-up table). With the invention of the integrated circuit camemask ROM. Mask ROM consists of a grid of word lines (the address input) and bit lines (the data output), selectively joined together with transistor switches, and can represent an arbitrary look-up table with a regular physical layout and predictable propagation delay.

Many game consoles use interchangeable ROM cartridges, allowing for one system to play multiple games.

In mask ROM, the data is physically encoded in the circuit, so it can only be programmed during fabrication. This leads to a number of serious disadvantages:

1. It is only economical to buy mask ROM in large quantities, since users must contract with a foundry to produce a custom design.
2. The turnaround time between completing the design for a mask ROM and receiving the finished product is long, for the same reason.
3. Mask ROM is impractical for R&D work since designers frequently need to modify the contents of memory as they refine a design.
4. If a product is shipped with faulty mask ROM, the only way to fix it is to recall the product and physically replace the ROM in every unit shipped.

Subsequent developments have addressed these shortcomings. PROM, invented in 1956, allowed users to program its contents exactly once by physically altering its structure with the application of high-voltage pulses. This addressed problems 1 and 2 above, since a company can simply order a large batch of fresh PROM chips and program them with the desired contents at its designers' convenience. The 1971 invention of EPROM essentially solved problem 3, since EPROM (unlike PROM) can be repeatedly reset to its unprogrammed state by exposure to strong ultraviolet light. EEPROM, invented in 1983, went a long way to solving problem 4, since an EEPROM can be programmed in-place if the containing device provides a means to receive the program contents from an external source (for example, a personal computer via aserial cable). Flash memory, invented at Toshiba in the mid-1980s, and commercialized in the early 1990s, is a form of EEPROM that makes very efficient use of chip area and can be erased and reprogrammed thousands of times without damage.

All of these technologies improved the flexibility of ROM, but at a significant cost-per-chip, so that in large quantities mask ROM would remain an economical choice for many years. (Decreasing cost of reprogrammable devices had almost eliminated the market for mask ROM by the year 2000.) Rewriteable technologies were envisioned as replacements for mask ROM.

The most recent development is NAND flash, also invented at Toshiba. Its designers explicitly broke from past practice, stating plainly that “the aim of NAND Flash is to replace hard disks,” rather than the traditional use of ROM as a form of non-volatile primary storage. As of 2007, NAND has partially achieved this goal by offering throughput comparable to hard disks, higher tolerance of physical shock, extreme miniaturization (in the form of USB flash drives and tiny microSD memory cards, for example), and much lower power consumption.

Use for storing programs

Every stored-program computer may use a form of non-volatile storage (that is, storage that retains its data when power is removed) to store the initial program that runs when the computer is powered on or otherwise begins execution (a process known as bootstrapping, often abbreviated to “booting” or “booting up”). Likewise, every non-trivial computer needs some form of mutable memory to record changes in its state as it executes.

Forms of read-only memory were employed as non-volatile storage for programs in most early stored-program computers, such as ENIAC after 1948. (Until then it was not a stored-program computer as every program had to be manually wired into the machine, which could take days to weeks.) Read-only memory was simpler to implement since it needed only a mechanism to read stored values, and not to change them in-place, and thus could be implemented with very crude electromechanical devices (see historical examples below). With the advent of integrated circuits in the 1960s, both ROM and its mutable counterpart static RAM were implemented as arrays of transistors in silicon chips; however, a ROM memory cell could be implemented using fewer transistors than an SRAM memory cell, since the latter needs a latch (comprising 5-20 transistors) to retain its contents, while a ROM cell might consist of the absence (logical 0) or presence (logical 1) of one transistor connecting a bit line to a word line. Consequently, ROM could be implemented at a lower cost-per-bit than RAM for many years.

Most home computers of the 1980s stored a BASIC interpreter or operating system in ROM as other forms of non-volatile storage such as magnetic disk drives were too costly. For example, the Commodore 64 included 64 KB of RAM and 20 KB of ROM contained a BASIC interpreter and the “KERNAL” of its operating system. Later home or office computers such as the IBM PC XT often included magnetic disk drives, and larger amounts of RAM, allowing them to load their operating systems from disk into RAM, with only a minimal hardware initialization core and bootloader remaining in ROM (known as the BIOS in IBM-compatible computers). This arrangement allowed for a more complex and easily upgradeable operating system.

In modern PCs, “ROM” (or flash) is used to store the basic bootstrapping firmware for the main processor, as well as the various firmware needed to internally control self-contained devices such as graphic cards, hard disks, DVD drives, TFT screens, etc., in the system. Today, many of these “read-only” memories – especially the BIOS – are often replaced with Flash memory (see below), to permit in-place reprogramming should the need for a firmware upgrade arise. However, simple and mature sub-systems (such as the keyboard or some communication controllers in the integrated circuits on the main board, for example) may employ mask ROM or OTP (one-time programmable).

ROM and successor technologies such as flash are prevalent in embedded systems. These are in everything from industrial robots to home appliances and consumer electronics(MP3 players, set-top boxes, etc.) all of which are designed for specific functions, but are based on general-purpose microprocessors. With software usually tightly coupled to hardware, program changes are rarely needed in such devices (which typically lack hard disks for reasons of cost, size, or power consumption). As of 2008, most products use Flash rather than mask ROM, and many provide some means for connecting to a PC for firmware updates; for example, a digital audio player might be updated to support a new file format. Some hobbyists have taken advantage of this flexibility to reprogram consumer products for new purposes; for example, the iPodLinux and OpenWrt projects have enabled users to run full-featured Linux distributions on their MP3 players and wireless routers, respectively.

ROM is also useful for binary storage of cryptographic data, as it makes them difficult to replace, which may be desirable in order to enhance information security.

Use for storing data

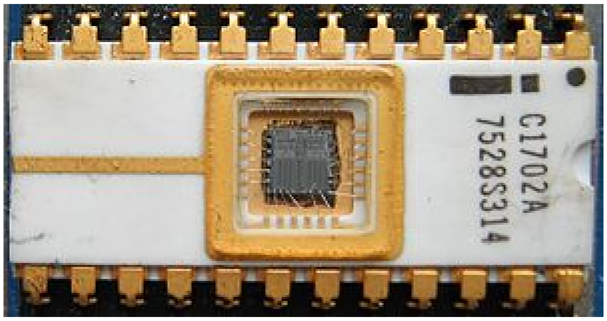
Since ROM (at least in hard-wired mask form) cannot be modified, it is really only suitable for storing data which is not expected to need modification for the life of the device. To that end, ROM has been used in many computers to store look-up tables for the evaluation of mathematical and logical functions (for example, a floating-point unit might tabulate the sine function in order to facilitate faster computation). This was especially effective when CPUs were slow and ROM was cheap compared to RAM.

Notably, the display adapters of early personal computers stored tables of bitmapped font characters in ROM. This usually meant that the text display font could not be changed interactively. This was the case for both the CGA and MDA adapters available with the IBM PC XT.

The use of ROM to store such small amounts of data has disappeared almost completely in modern general-purpose computers. However, Flash ROM has taken over a new role as a medium for mass storage or secondary storage of files.

Types

Semiconductor based



The first EPROM, an Intel 1702, with the die and wire bonds clearly visible through the erase window.

Classic *mask-programmed ROM* chips are integrated circuits that physically encode the data to be stored, and thus it is impossible to change their contents after fabrication. Other types of non-volatile solid-state memory permit some degree of modification:

- *Programmable read-only memory* (PROM), or *one-time programmable ROM* (OTP), can be written to or *programmed* via a special device called a *PROM programmer*. Typically, this device uses high voltages to permanently destroy or create internal links (fuses or antifuses) within the chip. Consequently, a PROM can only be programmed once.
- *Erasable programmable read-only memory* (EPROM) can be erased by exposure to strong ultraviolet light (typically for 10 minutes or longer), then rewritten with a process that again needs higher than usual voltage applied. Repeated exposure to UV light will eventually wear out an EPROM, but the *endurance* of most EPROM chips exceeds 1000 cycles of erasing and reprogramming. EPROM chip packages can often be identified by the prominent quartz “window” which allows UV light to enter. After programming, the window is typically covered with a label to prevent accidental erasure. Some EPROM chips are factory-erased before they are packaged, and include no window; these are effectively PROM.
- *Electrically erasable programmable read-only memory* (EEPROM) is based on a similar semiconductor structure to EPROM, but allows its entire contents (or selected *banks*) to be electrically erased, then rewritten electrically, so that they need not be removed from the computer (or camera, MP3 player, etc.). Writing or *flashing* an EEPROM is much slower (milliseconds per bit) than reading from a ROM or writing to a RAM (nanoseconds in both cases).
    - *Electrically alterable read-only memory* (EAROM) is a type of EEPROM that can be modified one bit at a time. Writing is a very slow process and again needs higher voltage (usually around 12 V) than is used for read access. EAROMs are intended for applications that require infrequent and only partial rewriting. EAROM may be used as non-volatile storage for critical system setup information; in many applications, EAROM has been supplanted by CMOS RAM supplied by mains power and backed-up with a lithium battery.
    - *Flash memory* (or simply *flash*) is a modern type of EEPROM invented in 1984. Flash memory can be erased and rewritten faster than ordinary EEPROM, and newer designs feature very high endurance (exceeding 1,000,000 cycles). Modern NAND flash makes efficient use of silicon chip area, resulting in individual ICs with a capacity as high as 32 GB as of 2007; this feature, along with its endurance and physical durability, has allowed NAND flash to replace magnetic in some applications (such as USB flash drives). Flash memory is sometimes called *flash ROM* or *flash EEPROM* when used as a replacement for older ROM types, but not in applications that take advantage of its ability to be modified quickly and frequently.

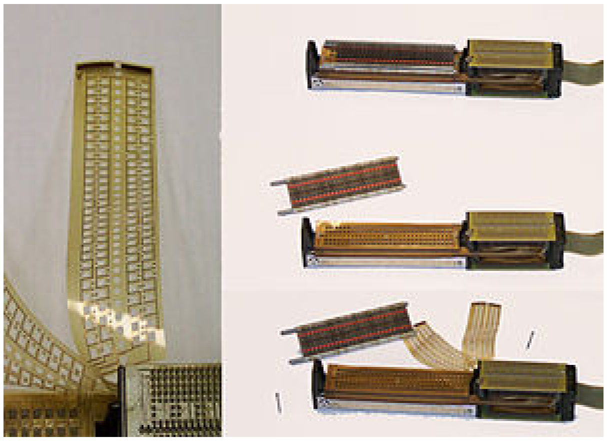
By applying write protection, some types of reprogrammable ROMs may temporarily become read-only memory.

Other technologies

There are other types of non-volatile memory which are not based on solid-state IC technology, including:

- Optical storage media, such CD-ROM which is read-only (analogous to masked ROM). CD-R is Write Once Read Many (analogous to PROM), while CD-RW supports erase-rewrite cycles (analogous to EEPROM); both are designed for backwards-compatibility with CD-ROM.

Historical examples



Transformer matrix ROM (TROS), from the IBM System 360/20

- Diode matrix ROM, used in small amounts in many computers in the 1960s as well as electronic desk calculators and keyboard encoders for terminals. This ROM was programmed by installing discrete semiconductor diodes at selected locations between a matrix of *word line traces* and *bit line traces* on a printed circuit board.
- Resistor, capacitor, or transformer matrix ROM, used in many computers until the 1970s. Like diode matrix ROM, it was programmed by placing components at selected locations between a matrix of *word lines* and *bit lines*. ENIAC's Function Tables were resistor matrix ROM, programmed by manually setting rotary switches. Various models of the IBM System/360and complex peripheral devices stored their microcode in either capacitor (called *BCROS* for *balanced capacitor read-only storage* on the 360/50 and 360/65, or *CCROS* for *charged capacitor read-only storage* on the 360/30) or transformer (called*TROS* for *transformer read-only storage* on the 360/20, 360/40 and others) matrix ROM.
- Core rope, a form of transformer matrix ROM technology used where size and weight were critical. This was used inNASA/MIT's Apollo Spacecraft Computers, DEC's PDP-8 computers, and other places. This type of ROM was programmed by hand by weaving "word line wires" inside or outside of ferrite transformer cores.
- Dimond Ring stores, in which wires are threaded through a sequence of large ferrite rings that function only as sensing devices. These were used in TXE telephone exchanges.
- The perforated metal character mask ("stencil") in Charactron cathode ray tubes, which was used as ROM to shape a wide electron beam to form a selected character shape on the screen either for display or a scanned electron beam to form a selected character shape as an overlay on a video signal.

Speed

Reading

Although the relative speed of RAM vs. ROM has varied over time, as of 2007 large RAM chips can be read faster than most ROMs. For this reason (and to allow uniform access), ROM content is sometimes copied to RAM or shadowed before its first use, and subsequently read from RAM.

Writing

For those types of ROM that can be electrically modified, writing speed is always much slower than reading speed, and it may need unusually high voltage, the movement of jumper plugs to apply write-enable signals, and special lock/unlock command codes. Modern NAND Flash achieves the highest write speeds of any rewritable ROM technology, with speeds as high as 15 MB/s (or 70 ns/bit), by allowing (needing) large blocks of memory cells to be written simultaneously.

Endurance and data retention

Because they are written by forcing electrons through a layer of electrical insulation onto a floating transistor gate, rewriteable ROMs can withstand only a limited number of write and erase cycles before the insulation is permanently damaged. In the earliest EAROMs, this might occur after as few as 1,000 write cycles, while in modern Flash EEPROM the endurance may exceed 1,000,000, but it is by no means infinite. This limited endurance, as well as the higher cost per bit, means that Flash-based storage is unlikely to completely supplant magnetic disk drives in the near future.

The timespan over which a ROM remains accurately readable is not limited by write cycling. The **data retention** of EPROM, EAROM, EEPROM, and Flash *may* be limited by charge leaking from the floating gates of the memory cell transistors. Leakage is accelerated by high temperatures or radiation. Masked ROM and fuse/antifuse PROM do not suffer from this effect, as their data retention depends on physical rather than electrical permanence of the integrated circuit (although *fuse re-growth* was once a problem in some systems).

Content images

The contents of ROM chips in video game console cartridges can be extracted with special software or hardware devices. The resultant memory dump files are known as **ROM images**, and can be used to produce duplicate cartridges, or in console emulators. The term originated when most console games were distributed on cartridges containing ROM chips, but achieved such widespread usage that it is still applied to images of newer games distributed on CD-ROMs or other optical media.

ROM images of commercial games usually contain copyrighted software. The unauthorized copying and distribution of copyrighted software is usually a violation of copyright laws (in some jurisdictions, duplication of ROM cartridges for backup purposes may be considered fair use). Nevertheless, there is a thriving community engaged in the illegal distribution and trading of such software and abandonware. In such circles, the term “ROM images” is sometimes shortened simply to “ROMs” or sometimes changed to “romz” to highlight the connection with “warez”.

Licensing & Attributions

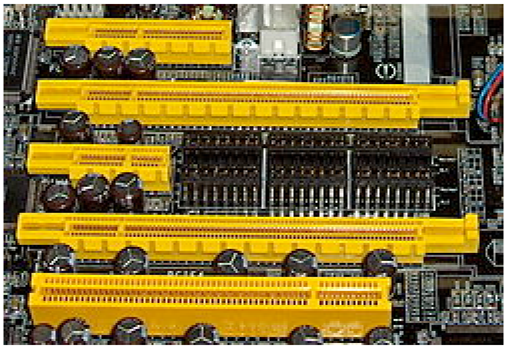
CC licensed content, Shared previously

- Read-only memory. **Provided by:** Wikipedia. **Located at:** https://en.wikipedia.org/wiki/Read-only_memory. **License:** *CC BY-SA: Attribution-ShareAlike*

READING: BUS

In computer architecture, a **bus** (related to the Latin “omnibus”, meaning “for all”) is a communication system that transfers data between components inside a computer, or between computers. This expression covers all related hardware components (wire, optical fiber, etc.) and software, including communication protocols.

Early computer buses were parallel electrical wires with multiple connections, but the term is now used for any physical arrangement that provides the same logical functionality as a parallel electrical bus. Modern computer buses can use both parallel and bit serial connections, and can be wired in either a multidrop (electrical parallel) or daisy chain topology, or connected by switched hubs, as in the case of USB.



4 PCI Express bus card slots (from top to bottom: x4, x16, x1 and x16), compared to a 32-bit conventional PCI bus card slot (very bottom)

Background and nomenclature

Computer systems generally consist of three main parts: the central processing unit (CPU) that processes data, memory that holds the programs and data to be processed, and I/O (input/output) devices as peripherals that communicate with the outside world. An early computer might use a hand-wired CPU of vacuum tubes, a magnetic drum for main memory, and a punch tape and printer for reading and writing data. In a modern system we might find a multi-core CPU, DDR3 SDRAM for memory, a hard drive for secondary storage, agraphics card and LCD display as a display system, a mouse and keyboard for interaction, and a Wi-Fi connection for networking. In both examples, computer buses of one form or another move data between all of these devices.

In most traditional computer architectures, the CPU and main memory tend to be tightly coupled. A microprocessor conventionally is a single chip which has a number of electrical connections on its pins that can be used to select an “address” in the main memory and another set of pins to read and write the data stored at that location. In most cases, the CPU and memory share signalling characteristics and operate in synchrony. The bus connecting the CPU and memory is one of the defining characteristics of the system, and often referred to simply as the system bus.

It is possible to allow peripherals to communicate with memory in the same fashion, attaching adaptors in the form of expansion cards directly to the system bus. This is commonly accomplished through some sort of standardized electrical connector, several of these forming the expansion bus or local bus. However, as the performance differences between the CPU and peripherals varies widely, some solution is generally needed to ensure that peripherals do not slow overall system performance. Many CPUs feature a second set of pins similar to those for communicating with memory, but able to operate at very different speeds and using different protocols. Others use smart controllers to place the data directly in memory, a concept known as direct memory access. Most modern systems combine both solutions, where appropriate.

As the number of potential peripherals grew, using an expansion card for every peripheral became increasingly untenable. This has led to the introduction of bus systems designed specifically to support multiple peripherals. Common examples are the SATA ports in modern computers, which allow a number of hard drives to be connected without the need for a card. However, these high-performance systems are generally too expensive to implement in low-end devices, like a mouse. This has led to the parallel development of a number of low-performance bus systems for these solutions, the most common example being Universal Serial Bus. All such examples may be referred to as peripheral buses, although this terminology is not universal.

In modern systems the performance difference between the CPU and main memory has grown so great that increasing amounts of high-speed memory is built directly into the CPU, known as a cache. In such systems, CPUs communicate using high-performance buses that operate at speeds much greater than memory, and communicate with memory using protocols similar to those used solely for peripherals in the past. These system buses are also used to communicate with most (or all) other peripherals, through adaptors, which in turn talk to other peripherals and controllers. Such systems are architecturally more similar to multicomputers, communicating over a bus rather than a network. In these cases, expansion buses are entirely separate and no longer share any architecture with their host CPU (and may in fact support many different CPUs, as is the case with PCI). What would have formerly been a system bus is now often known as a front-side bus.

Given these changes, the classical terms “system”, “expansion” and “peripheral” no longer have the same connotations. Other common categorization systems are based on the buses primary role, connecting devices internally or externally, PCI vs. SCSI for instance. However, many common modern bus systems can be used for both; SATA and the associated eSATA are one example of a system that would formerly be described as internal, while in certain automotive applications use the primarily external IEEE 1394 in a fashion more similar to a system bus. Other examples, like InfiniBand and I²C were designed from the start to be used both internally and externally.

Internal bus

The internal bus, also known as internal data bus, memory bus, system bus or Front-Side-Bus, connects all the internal components of a computer, such as CPU and memory, to the motherboard. Internal data buses are also referred to as a local bus, because they are intended to connect to local devices. This bus is typically rather quick and is independent of the rest of the computer operations.

External bus

The external bus, or expansion bus, is made up of the electronic pathways that connect the different external devices, such as printer etc., to the computer.

Implementation details

Buses can be parallel buses, which carry data words in parallel on multiple wires, or serial buses, which carry data in bit-serial form. The addition of extra power and control connections, differential drivers, and data connections in each direction usually means that most serial buses have more conductors than the minimum of one used in 1-Wire andUNI/O. As data rates increase, the problems of timing skew, power consumption, electromagnetic interference and crosstalk across parallel buses become more and more difficult to circumvent. One partial solution to this problem has been to double pump the bus. Often, a serial bus can be operated at higher overall data rates than a parallel bus, despite having fewer electrical connections, because a serial bus inherently has no timing skew or crosstalk. USB, FireWire, and Serial ATA are examples of this. Multidrop connections do not work well for fast serial buses, so most modern serial buses use daisy-chain or hub designs.

Network connections such as Ethernet are not generally regarded as buses, although the difference is largely conceptual rather than practical. An attribute generally used to characterize a bus is that power is provided by the bus for the connected hardware. This emphasizes the busbar origins of bus architecture as supplying switched or distributed power. This excludes, as buses, schemes such as serial RS-232, parallel Centronics, IEEE 1284 interfaces and Ethernet, since these devices also needed separate power supplies.Universal Serial Bus devices may use the bus supplied power, but often use a separate power source. This distinction is exemplified by a telephone system with a connected modem, where the RJ11 connection and associated modulated signalling scheme is not considered a bus, and is analogous to an Ethernet connection. A phone line connection scheme is not considered to be a bus with respect to signals, but the Central Office uses buses with cross-bar switches for connections between phones.

However, this distinction—that power is provided by the bus—is not the case in many avionic systems, where data connections such as ARINC 429, ARINC 629, MIL-STD-1553B(STANAG 3838), and EFABus (STANAG 3910) are commonly referred to as “data buses” or, sometimes, “databuses”. Such avionic data buses are usually characterized by having several equipments or Line Replaceable Items/Units (LRI/LRUs) connected to a common, shared media. They may, as with ARINC 429, be simplex, i.e. have a single source LRI/LRU or, as with ARINC 629, MIL-STD-1553B, and STANAG 3910, be duplex, allow all the connected LRI/LRUs to act, at different times (half duplex), as transmitters and receivers of data.

History

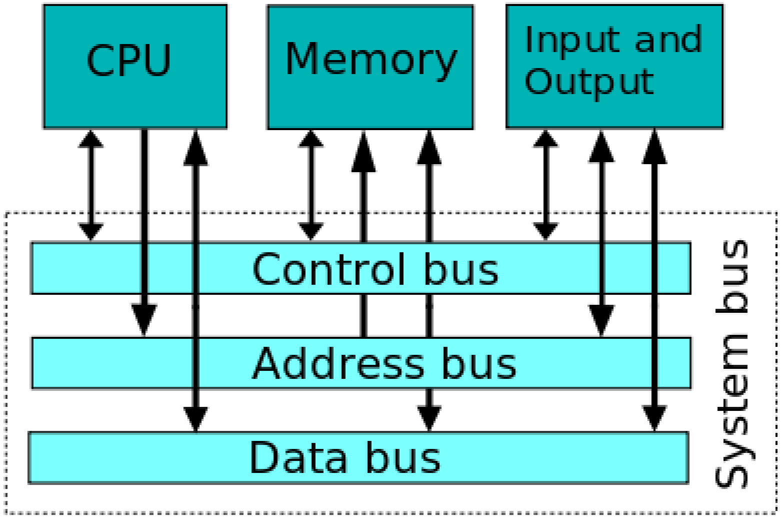
Over time, several groups of people worked on various computer bus standards, including the IEEE Bus Architecture Standards Committee (BASC), the IEEE “Superbus” study group, the open microprocessor initiative (OMI), the open microsystems initiative (OMI), the “Gang of Nine” that developed EISA, etc.

First generation

Early computer buses were bundles of wire that attached computer memory and peripherals. Anecdotally termed the “*digit trunk*“, they were named after electrical power buses, or busbars. Almost always, there was one bus for memory, and one or more separate buses for peripherals. These were accessed by separate instructions, with completely different timings and protocols.

One of the first complications was the use of interrupts. Early computer programs performed I/O by waiting in a loop for the peripheral to become ready. This was a waste of time for programs that had other tasks to do. Also, if the program attempted to perform those other tasks, it might take too long for the program to check again, resulting in loss of data. Engineers thus arranged for the peripherals to interrupt the CPU. The interrupts had to be prioritized, because the CPU can only execute code for one peripheral at a time, and some devices are more time-critical than others.

High-end systems introduced the idea of channel controllers, which were essentially small computers dedicated to handling the input and output of a given bus. IBM introduced these on the IBM 709 in 1958, and they became a common feature of their platforms. Other high-performance vendors like Control Data Corporation implemented similar designs. Generally, the channel controllers would do their best to run all of the bus operations internally, moving data when the CPU was known to be busy elsewhere if possible, and only using interrupts when necessary. This greatly reduced CPU load, and provided better overall system performance.



Single system bus

To provide modularity, memory and I/O buses can be combined into a unified system bus. In this case, a single mechanical and electrical system can be used to connect together many of the system components, or in some cases, all of them.

Later computer programs began to share memory common to several CPUs. Access to this memory bus had to be prioritized, as well. The simple way to prioritize interrupts or bus access was with a daisy chain. In this case signals will naturally flow through the bus in physical or logical order, eliminating the need for complex scheduling.

Minis and micros

Digital Equipment Corporation (DEC) further reduced cost for mass-produced minicomputers, and mapped peripherals into the memory bus, so that the input and output devices appeared to be memory locations. This was implemented in the Unibus of the PDP-11 around 1969.

Early microcomputer bus systems were essentially a passive backplane connected directly or through buffer amplifiers to the pins of the CPU. Memory and other devices would be added to the bus using the same address and data pins as the CPU itself used, connected in parallel. Communication was controlled by the CPU, which had read and written data from the devices as if they are blocks of memory, using the same instructions, all timed by a central clock controlling the speed of the CPU. Still, devices interrupted the CPU by signaling on separate CPU pins. For instance, a disk drive controller would signal the CPU that new data was ready to be read, at which point the CPU would move the data by reading the “memory location” that corresponded to the disk drive. Almost all early microcomputers were built in this fashion, starting with the S-100 bus in the Altair 8800 computer system.

In some instances, most notably in the IBM PC, although similar physical architecture can be employed, instructions to access peripherals (`in` and `out`) and memory (`mov` and others) have not been made uniform at all, and still generate distinct CPU signals, that could be used to implement a separate I/O bus.

These simple bus systems had a serious drawback when used for general-purpose computers. All the equipment on the bus has to talk at the same speed, as it shared a single clock.

Increasing the speed of the CPU becomes harder, because the speed of all the devices must increase as well. When it is not practical or economical to have all devices as fast as the CPU, the CPU must either enter a wait state, or work at a slower clock frequency temporarily, to talk to other devices in the computer. While acceptable in embedded systems, this problem was not tolerated for long in general-purpose, user-expandable computers.

Such bus systems are also difficult to configure when constructed from common off-the-shelf equipment. Typically each added expansion card requires many jumpers in order to set memory addresses, I/O addresses, interrupt priorities, and interrupt numbers.

Second generation

“Second generation” bus systems like NuBus addressed some of these problems. They typically separated the computer into two “worlds”, the CPU and memory on one side, and the various devices on the other. A *bus controller* accepted data from the CPU side to be moved to the peripherals side, thus shifting the communications protocol burden from the CPU itself. This allowed the CPU and memory side to evolve separately from the device bus, or just “bus”. Devices on the bus could talk to each other with no CPU intervention. This led to much better “real world” performance, but also required the cards to be much more complex. These buses also often addressed speed issues by being “bigger” in terms of the size of the data path, moving from 8-bit parallel buses in the first generation, to 16 or 32-bit in the second, as well as adding software setup (now standardised as Plug-n-play) to supplant or replace the jumpers.

However these newer systems shared one quality with their earlier cousins, in that everyone on the bus had to talk at the same speed. While the CPU was now isolated and could increase speed, CPUs and memory continued to increase in speed much faster than the buses they talked to. The result was that the bus speeds were now very much slower than what a modern system needed, and the machines were left starved for data. A particularly common example of this problem was that video cards quickly outran even the newer bus systems like PCI, and computers began to include AGP just to drive the video card. By 2004 AGP was outgrown again by high-end video cards and other peripherals and has been replaced by the new PCI Express bus.

An increasing number of external devices started employing their own bus systems as well. When disk drives were first introduced, they would be added to the machine with a card plugged into the bus, which is why computers have so many slots on the bus. But through the 1980s and 1990s, new systems like SCSI and IDE were introduced to serve this need, leaving most slots in modern systems empty. Today there are likely to be about five different buses in the typical machine, supporting various devices.

Third generation

“Third generation” buses have been emerging into the market since about 2001, including HyperTransport and InfiniBand. They also tend to be very flexible in terms of their physical connections, allowing them to be used both as internal buses, as well as connecting different machines together. This can lead to complex problems when trying to service different requests, so much of the work on these systems concerns software design, as opposed to the hardware itself. In general, these third generation buses tend to look more like a network than the original concept of a bus, with a higher protocol overhead needed than early systems, while also allowing multiple devices to use the bus at once.

Buses such as Wishbone have been developed by the open source hardware movement in an attempt to further remove legal and patent constraints from computer design.

Licensing & Attributions

CC licensed content, Shared previously

- Bus (computing). **Provided by:** Wikipedia. **Located at:** https://en.wikipedia.org/wiki/Bus_(computing). **License:** *CC BY-SA: Attribution-ShareAlike*

MODULE 3: SYSTEM SOFTWARE

READING: INTRO TO OPERATING SYSTEMS

https://learn.saylor.org/course/view.php?id=94&sectionid=967

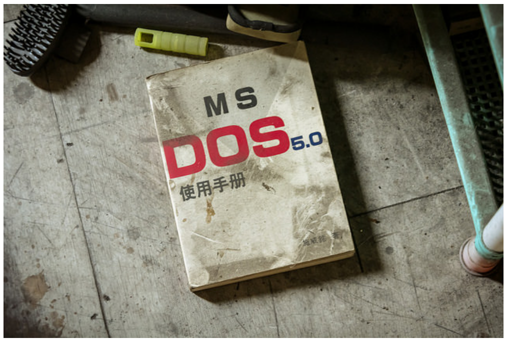
Licensing & Attributions

Public domain content

- License: *Public Domain: No Known Copyright*

READING: OPERATING SYSTEMS

Introduction



The PC of a business end-user will quite commonly provide a number of different application programs, but no program development tools. It is also possible (although unlikely) that the PC of a programmer could provide development tools only, and no other commercial applications. But the one thing that all these machines would definitely have is an operating system (OS). The operating system is an essential piece of software that resides on virtually every computer. It allows the computer to run a number of different applications apparently simultaneously, and shares resources such as printers between a number of different users, while at the same time performing those functions that are critical to the correct operating of the computer system, regardless of which application is running.

Different types of computers may require different operating systems. Most mainframe computers have a proprietary operating system, such as IBM's MVS or Siemen's BS2000. At the opposite end of the market, we find that the leading operating system for the IBM compatible Personal Computers is Microsoft Windows (having all but replaced its predecessor MS-DOS) although Mac-OS and Unix variants such as Linux also have strong followings.

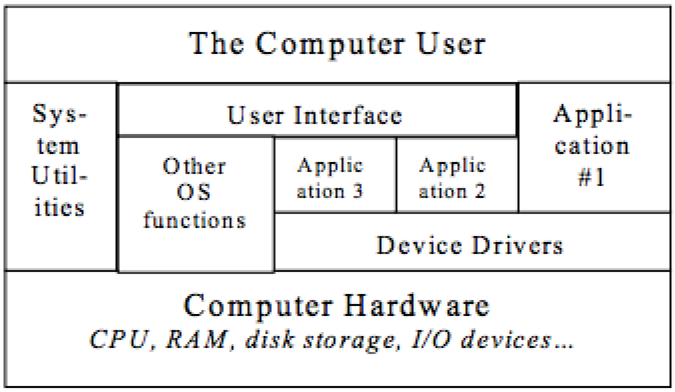


Figure 5-4: Some operating system components

Today's operating systems are true marvels of engineering and rate among the most complex artifacts designed by humans. Their complexity stems from the increasing number of functions that an operating system is required to handle.

Multitasking

Today's computers handle many applications at the same time: you could be downloading a file from the Internet at the same time as you update a spreadsheet that is embedded within a word-processing document (while at the same time your virus scanner is running in the background). Each application needs to share the CPU, main memory and all peripheral devices. The OS sees to it that all conflicts are resolved and each application gets a fair share of the common resources, often in accordance with a certain priority schedule. An important task, therefore, is that of the proper scheduling of different tasks. The OS also ensures that illbehaving applications do not impact on the integrity of the other applications and their data. (By the way, when a single application program runs multiple tasks simultaneously, this is known as multithreading.)

Multi-user management

Barring the personal computer, computers have more than a single user at any one time – a typical mainframe may have to handle many thousands of simultaneous users. Each of the users may operate a different application (see multitasking) but will not be prepared to wait for other users to finish their tasks. So the OS sees to it that each user gets a fair share of the CPU of the mainframe, by slicing each second in tiny fractions and allocating them to the different users. In addition, each user may have a different profile in terms of access to certain peripheral devices and data. This requires sophisticated security and data management.

Memory management

Apart from the CPU, primary memory is often the most expensive and scarcest component of a computer system. Efficient memory management is a key component of any OS, especially in multitasking and multi-user environment. The likelihood that there is sufficient memory to load all the data and software for all users simultaneously is extremely small. In practice, a technique of virtual memory is used whereby the available amount of main memory is extended by using secondary storage devices, usually the hard disk. During a typical program execution, not all program instructions and data are required at once. The OS will swap out all unused software and data segments onto disk, until they are actually required.

Secondary storage management

All applications require disk storage – if only to store the application software itself. The OS provides the common access routines to the secondary disk storage devices. It will keep track of data and program file names, physical locations and perform common functions such as the copying, erasing or backing up of data. It also makes the differences in physical hardware as transparent as possible: a data file will appear the same to an application whether the data is being loaded from a diskette, a hard disk or an optical CD-ROM.

Peripheral handler

Apart from managing the memory and secondary storage devices, the OS is usually also responsible for the handling of other peripheral devices. Again, a major objective is to relieve the application from the responsibility of having to cater for a multitude of different possible input and output devices. Thanks to the OS, a program (developer) does not need to cater for the differences between a trackball, mouse or other pointing device. The OS will also ensure that your document will be printed correctly, regardless of whether you have a colour inkjet, a laser or lowly dot-matrix printer attached. Equally, individual programs no longer have to worry about the resolution or capability of your computer monitor. The critical element is again that the applications do not have to worry about the actual hardware connected to the computer system. This is usually achieved by means of device drivers, small software routines that become part of the operating system and help it to interface correctly with specific hardware devices. The device drivers for popular or standard devices are usually already included in the operating system. More esoteric or very new hardware devices will come with a separate disk containing the proper device drivers. These then have to be installed as part of the operating system at the same time as connecting the hardware.

Communications Management

Communications software allows a computer to recognise the presence of other machines on a network, and to grant or restrict access to local files. It also manages network traffic, by monitoring communications lines and diagnosing problems, as well as sending and receiving data to and from remote devices. While PC operating systems include some basic networking and communications facilities, network operating systems (NOS) such as Novell and Unix provide sophisticated management tools for large networks.

User interface

Until the late 1980s, operating systems focussed on the efficient management of resources, tasks and secondary storage. The popularity of personal computers meant that a whole class of less-technical end-users entered the world of computing. This pressured the IT industry into developing a user-friendlier graphical user-interface and the user interface is quickly becoming a significant part of microcomputer operating systems. Different types of user interface have already been discussed earlier in this chapter.

System Utilities

As a logical extension of the operating system, additional system utilities are often required to assist with the proper and optimal use of the computer. Some system utilities may be provided along with the operating system, while others are marketed by independent third-party vendors. A number of system utilities focuses on improving the functionality of the operating system (e.g. better hard disk management) while others provide additional functions (e.g. anti-virus or encryption software). More examples of system utilities are editors, additional security utilities, performance monitors, file viewers, data compression software etc.

Licensing & Attributions

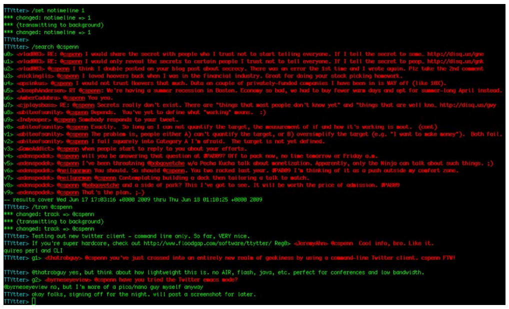
CC licensed content, Shared previously

- Discovering Information Systems Section 5.4. **Authored by**: Jean-Paul Van Belle, Mike Eccles, and Jane Nash. **Located at**: https://archive.org/details/ost-computer-science-discoveringinformationsystems. **License**: *CC BY-NC-ND: Attribution-NonCommercial-NoDerivatives*
- Abandoned MS DOS 5.0 operating system manual. **Authored by**: Alexander Synaptic. **Located at**: https://www.flickr.com/photos/synapticism/15836248264. **License**: *CC BY-NC: Attribution-NonCommercial*

READING: SYSTEM SOFTWARE

Introduction

System software (systems software) is computer software designed to operate and control the computer hardware and to provide a platform for running application software. System software can be separated into two different categories, operating systems and utility software.



```
TTYtter> /set notimeline 1
*** changed: notimeline => 1
*** (transmitting to background)
*** changed: notimeline => 1
TTYtter>
TTYtter> /search @cspenn
u0> <vlad003> RE: @cspenn I would share the secret with people who I trust not to start telling everyone. If I tell the secret to some… http://disq.us/gne
u1> <vlad003> RE: @cspenn I would only reveal the secrets to certain people I trust not to tell everyone. If I tell the secret to peop… http://disq.us/gnk
u2> <vlad003> @cspenn I think I double posted on your blog post about secrecy. There was an error the 1st time and I wrote again. Plz take the 2nd comment
u3> <nickinglis> @cspenn I loved hoovers back when I was in the financial industry. Great for doing your stock picking homework.
u4> <apsinkus> @cspenn I would not trust Hoovers that much. Data on couple of privately-funded companies I have been in is WAY off (like 10X).
u5> <JosephAndersen> RT @cspenn: We're having a summer recession in Boston. Economy so bad, we had to buy fewer warm days and opt for summer-long April instead.
u6> <AmberCadabra> @cspenn Yea yea.
u7> <cjplaysbass> RE: @cspenn Secrets really don't exist. There are "things that most people don't know yet" and "things that are well kno… http://disq.us/gwy
u8> <abiteofsanity> @cspenn Depends.  You've yet to define what "working" means.  :)
u9> <Indyooper> @cspenn Somebody responds to your tweet.
v0> <abiteofsanity> @cspenn Exactly.  So long as I can not quantify the target, the measurement of if and how it's working is moot.  (cont)
v1> <abiteofsanity> @cspenn The problem is, people either A) can't quantify the target, or B) oversimplify the target (e.g. "I want to make money").  Both fail.
v2> <abiteofsanity> @cspenn I fall squarely into Category A I'm afraid.  The target is not yet defined.
v3> <GameAddict> @cspenn when people start to reply to you about your efforts.
v4> <edenspodek> @cspenn will you be answering that question at #PAB09? Off to pack now, no time tomorrow or Friday a.m.
v5> <edenspodek> @cspenn I've been threatning @bobgoyetche w/a Pecha Kucha talk about monetization. Apparently, only the Ninja can talk about such things. ;)
v6> <edenspodek> @neilgorman You should. So should @cspenn. You two rocked last year. #PAB09 I'm thinking of it as a push outside my comfort zone.
v7> <edenspodek> @neilgorman @cspenn Contemplating building a deck then tailoring a talk to match.
v8> <edenspodek> @cspenn @bobgoyetche and a side of pork? This I've got to see. It will be worth the price of admission. #PAB09
v9> <edenspodek> @cspenn That's the plan. ;-)
-- results cover Wed Jun 17 17:03:16 +0000 2009 thru Thu Jun 18 01:10:25 +0000 2009
TTYtter> /tron @cspenn
*** changed: track => @cspenn
*** (transmitting to background)
*** changed: track => @cspenn
TTYtter> Testing out new twitter client - command line only. So far, VERY nice.
TTYtter> If you're super hardcore, check out http://www.floodgap.com/software/ttytter/ Reg0> <JeremyAhn> @cspenn  Cool info, bro. Like it.
quires perl and CLI
TTYtter> g1> <thatrobguy> @cspenn you've just crossed into an entirely new realm of geekiness by using a command-line Twitter client. cspenn FTW!

TTYtter> @thatrobguy yes, but think about how lightweight this is. no AIR, flash, java, etc. perfect for conferences and low bandwidth.
TTYtter> g2> <byrneseyeview> @cspenn have you tried the Twitter emacs mode?
@byrneseyeview no, but I'm more of a pico/nano guy myself anyway
TTYtter> okay folks, signing off for the night. will post a screenshot for later.
TTYtter> 
```

- The *operating system* (prominent examples being z/OS, Microsoft Windows, Mac OS X and Linux), allows the parts of a computer to work together by performing tasks like transferring data between memory and disks or rendering output onto a display device. It also provides a platform to run high-level system software and application software.
    - A *kernel* is the core part of the operating system that defines an API for applications programs (including some system software) and an interface to device drivers.
        - Device drivers such as computer BIOS and device firmware provide basic functionality to operate and control the hardware connected to or built into the computer.
    - A *user interface* "allows users to interact with a computer." Since the 1980s the graphical user interface (GUI) has been perhaps the most common user interface technology. The command-line interface is still a commonly used alternative.
- *Utility software* helps to analyze, configure, optimize and maintain the computer, such as virus protection.

In some publications, the term *system software* also includes software development tools (like a compiler, linker or debugger).

In contrast to system software, software that allows users to do things like create text documents, play games, listen to music, or web browsers to surf the web are called application software. The line where the distinction should be drawn isn't always clear. Most operating systems bundle such software. Such software is not considered *system software* when it can be uninstalled without affecting the functioning of other software. Exceptions could be e.g. web browsers such as Internet Explorer where Microsoft argued in court that it was system software that could not be uninstalled. Later examples are Chrome OS and Firefox OS where the browser functions as the only user interface *and* the only way to run programs (and other web browser can not be installed in their place), then they can well be argued to *be* (part of) the operating system and then system software.

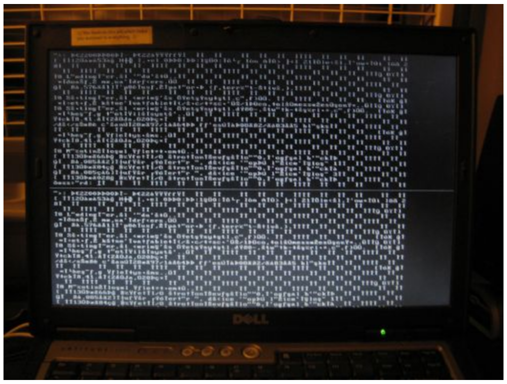
Licensing & Attributions

CC licensed content, Shared previously

- System Software. **Located at:** https://en.wikipedia.org/wiki/System_software. **License:** *CC BY-NC-SA: Attribution-NonCommercial-ShareAlike*
- Twitter in a command line interface. **Authored by:** Christopher Penn. **Located at:** https://www.flickr.com/photos/financialaidpodcast/3637854482/. **License:** *CC BY-NC-ND: Attribution-NonCommercial-NoDerivatives*

READING: BOOTING

Introduction



In computing, booting (or booting up) is the initialization of a computerized system. The system can be a computer or a computer appliance. The booting process can be “hard”, after electrical power to the CPU is switched from off to on (in order to diagnose particular hardware errors), or “soft,” when thosepower-on self-tests (POST) can be avoided. Soft booting can be initiated by hardware such as a button press, or by software command. Booting is complete when the normal, operative, runtime environment is attained.

A boot loader is a computer program that loads an operating system or some other system software for the computer after completion of the power-on self-tests; it is the loader for the operating system itself, which has its own loader for loading ordinary user programs and libraries. Within the hard reboot process, it runs after completion of the self-tests, then loads and runs the software. A boot loader is loaded into main memory from persistent memory, such as a hard disk drive or, in some older computers, from a medium such as punched cards, punched tape, or magnetic tape. The boot loader then loads and executes the processes that finalize the boot. Like POST processes, the boot loader code comes from a “hard-wired” and persistent location; if that location is too limited for some reason, that primary boot loader calls a second-stage boot loader or a secondary program loader.

On modern general purpose computers, the boot up process can take tens of seconds, and typically involves performing a power-on self-test, locating and initializing peripheral devices, and then finding, loading and starting an operating system. The process of hibernating or sleeping does not involve booting. Minimally, some embedded systems do not require a noticeable boot sequence to begin functioning and when turned on may simply run operational programs that are stored in ROM. All computing systems are state machines, and a reboot may be the only method to return to a designated zero-state from an unintended, locked state.

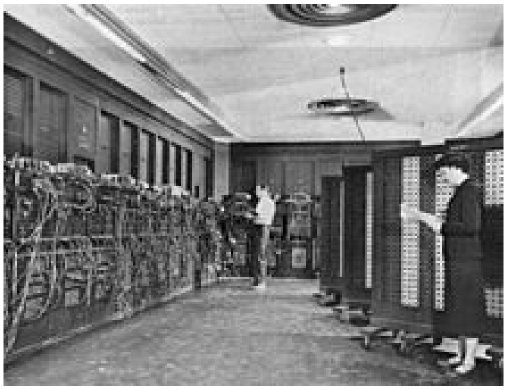
Boot is short for *bootstrap* or *bootstrap load* and derives from the phrase *to pull oneself up by one’s bootstraps*. The usage calls attention to the requirement that, if most software is loaded onto a computer by other software already running on the computer, some mechanism must exist to load the initial software onto the computer. Early computers used a variety of ad-hoc methods to get a small program into memory to solve this problem. The invention ofread-only memory (ROM) of various types solved this paradox by allowing computers to be shipped with a start up program that could not be erased. Growth in the capacity of ROM has allowed ever more elaborate start up procedures to be implemented.

History

There are many different methods available to load a short initial program into a computer. These methods reach from simple, physical input to removable media that can hold more complex programs.

Pre integrated-circuit-ROM examples

Early computers



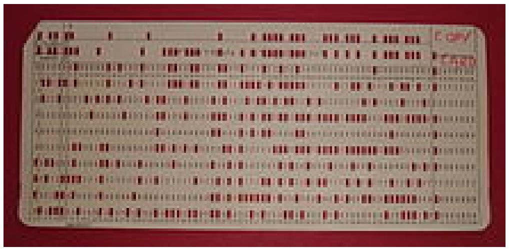
Switches and cables used to program ENIAC (1946)

Early computers in the 1940s and 1950s were one-of-a-kind engineering efforts that could take weeks to program and program loading was one of many problems that had to be solved. An early computer, ENIAC, had no “program” stored in memory, but was set up for each problem by a configuration of interconnecting cables. Bootstrapping did not apply to ENIAC, whose hardware configuration was ready for solving problems as soon as power was applied.

In 1960, the Ballistic Missile Early Warning System Display Information Processor (DIP) in Colorado Springs—before the NORAD facility was built in the underground Cheyenne Mountain Complex—ran only one program, which carried its own startup code. The program was stored as a bit image on a continuously running magnetic drum, and loaded in a fraction of a second. Core memory was probably cleared manually via the maintenance console, and startup from when power was fully up was very fast, only a few seconds. In its general design, the DIP compared roughly with aDEC PDP-8.

First commercial computers

The first programmable computers for commercial sale, such as the UNIVAC I and the IBM 701 included features to make their operation simpler. They typically included instructions that performed a complete input or output operation. The same hardware logic could be used to load the contents of a punch card or other input media that contained a bootstrap program by pressing a single button. This booting concept was called a variety of names for IBMcomputers of the 1950s and early 1960s, but IBM used the term “Initial Program Load” with the IBM 7030 Stretch and later used it for their mainframe lines, starting with the System/360 in 1964.



Initial program load punched card for the IBM 1130 (1965)

The IBM 701 computer (1952–1956) had a “Load” button that initiated reading of the first 36-bit word into main memory from a punched card in a card reader, a magnetic tape in a tape drive, or a magnetic drum unit, depending on the position of the Load Selector switch. The left 18-bit half-word was then executed as an instruction, which usually read additional words into memory. The loaded boot program was then executed, which, in turn, loaded a larger program from that medium into memory without further help from the human operator. The term “boot” has been used in this sense since at least 1958.

Other IBM computers of that era had similar features. For example, the IBM 1401 system (c. 1958) used a card reader to load a program from a punched card. The 80 characters stored in the punched card were read into memory locations 001 to 080, then the computer would branch to memory location 001 to read its first stored instruction. This instruction was always the same: move the information in these first 80 memory locations to an assembly area where the information in punched cards 2, 3, 4, and so on, could be combined to form the stored program. Once this information was moved to the assembly area, the machine would branch to an instruction in location 080 (read a card) and the next card would be read and its information processed.

IBM System/3 console from the 1970s. Program load selector switch is lower left; Program load switch is lower right.

Another example was the IBM 650 (1953), a decimal machine, which had a group of ten 10-position switches on its operator panel which were addressable as a memory word (address 8000) and could be executed as an instruction. Thus setting the switches to 7004000400 and pressing the appropriate button would read the first card in the card reader into memory (op code 70), starting at address 400 and then jump to 400 to begin executing the program on that card.

IBM's competitors also offered single button program load.

- The CDC 6600 (c. 1964) had a *dead start* panel with 144 toggle switches; the dead start switch entered 12 words from the toggle switches to the memory of *peripheral processor* (*PP*) 0 and initiated the load sequence. PP 0 loaded the necessary code into its own memory and then initialized the other PPs.
- The GE 645 (c. 1965) had a "SYSTEM BOOTLOAD" button that, when pressed, caused one of the I/O controllers to load a 64-word program into memory from a diode read-only memory and deliver an interrupt to cause that program to start running.
- The first model of the PDP-10 had a "READ IN" button that, when pressed, reset the processor and started an I/O operation on a device specified by switches on the control panel, reading in a 36-bit word giving a target address and count for subsequent word reads; when the read completed, the processor started executing the code read in by jumping to the last word read in.

A noteworthy variation of this is found on the Burroughs B1700 where there is neither a bootstrap ROM nor a hardwired IPL operation. Instead, after the system is reset it reads and executes opcodes sequentially from a tape drive mounted on the front panel; this sets up a boot loader in RAM which is then executed. However, since this makes few assumptions about the system it can equally well be used to load diagnostic (Maintenance Test Routine) tapes which display an intelligible code on the front panel even in cases of gross CPU failure.

IBM System/360 and successors

In the IBM System/360 and its successors, including the current z/Architecture machines, the boot process is known as *Initial Program Load* (IPL).

IBM coined this term for the 7030 (Stretch), revived it for the design of the System/360, and continues to use it in those environments today. In the System/360 processors, an IPL is initiated by the computer operator by selecting the three hexadecimal digit device address (CUU; C=I/O Channel address, UU=Control unit and Device address) followed by pressing the *LOAD* button. On most System/370 and some later systems, the functions of the switches and the LOAD button are simulated using selectable areas on the screen of a graphics console, often an IBM 2250-like device or an IBM 3270-like device. For example, on the System/370 Model 158, the keyboard sequence 0-7-X (zero, seven and X, in that order) results in an IPL from the device address which was keyed into the input area. Amdahl 470V/6 and related CPUs supported four hexadecimal digits on those CPUs which had the optional second channel unit installed, for a total of 32 channels. Later, IBM would also support more than 16 channels.

The IPL function in the System/360 and its successors, and its compatibles such as Amdahl's, reads 24 bytes from an operator-specified device into main storage starting at real address zero. The second and third groups of eight bytes are treated as Channel Command Words (CCWs) to continue loading the startup program (the first CCW is always simulated by the CPU and consists of a Read IPL command, 02h, with command chaining and suppress incorrect length indication being enforced). When the I/O channel commands are complete, the first group of eight bytes is then loaded into the processor's Program Status Word (PSW) and the startup program begins execution at the location designated by that PSW. The IPL device is usually a disk drive, hence the special significance of the 02h read-type command, but exactly the same procedure is also used to IPL from other input-type devices, such as tape drives, or even card readers, in a device-independent manner, allowing, for example, the installation of an operating system on a brand-new computer from an OS initial distribution magnetic tape. For disk controllers, the 02h command also causes the selected device to seek to cylinder 0000h, head 0000h, simulating a Seek cylinder and head command, 07h, and to search for record 01h, simulating a Search ID Equal command, 31h; seeks and searches are not simulated by tape and card controllers, as for these device classes an 02h command is simply a sequential read command, not a Read IPL command.

The disk, tape or card deck must contain a special program to load the actual operating system into main storage, and for this specific purpose "IPL Text" is placed on the disk by the stand-alone DASDI (Direct Access Storage Device Initialization) program or an equivalent program running under an operating system, e.g., ICKDSF, but IPL-able tapes and card decks are usually distributed with this "IPL Text" already present.

Minicomputers

Minicomputers, starting with the Digital Equipment Corporation (DEC) PDP-5 and PDP-8 (1965) simplified design by using the CPU to assist input and output operations. This saved cost but made booting more complicated than pressing a single button. Minicomputers typically had some way to *toggle in* short programs by manipulating an array of switches on the front panel. Since the early minicomputers used magnetic core memory, which did not lose its information when power was off, these bootstrap loaders would remain in place unless they were erased. Erasure sometimes happened accidentally when a program bug caused a loop that overwrote all of memory.

PDP-8/E front panel showing the switches used to load the bootstrap program.

Other minicomputers with such simple form of booting include Hewlett-Packard's HP 2100 series (mid-1960s), the original Data General Nova (1969), and DEC's PDP-11 (1970).

DEC later added an optional diode matrix read-only memory for the PDP-11 that stored a bootstrap program of up to 32 words (64 bytes). It consisted of a printed circuit card, the M792, that plugged into the Unibus and held a 32 by 16 array of semiconductor diodes. With all 512 diodes in place, the memory contained all "one" bits; the card was programmed by cutting off each diode whose bit was to be "zero." DEC also sold versions of the card, the BM792-Yx series, pre-programmed for many standard input devices by simply omitting the unneeded diodes.

Following the older approach, the earlier PDP-1 has a hardware loader, such that an operator need only push the "load" switch to instruct the paper tapereader to load a program directly into core memory. The Data General Supernova used front panel switches to cause the computer to automatically load instructions into memory from a device specified by the front panel's data switches, and then jump to loaded code; the Nova 800 and 1200 had a switch that loaded a program into main memory from a special read-only memory and jumped to it.

Early minicomputer boot loader examples

In a minicomputer with a paper tape reader, the first program to run in the boot process, the boot loader, would read into core memory either the second-stage boot loader (often called a *Binary Loader*) that could read paper tape with checksum or the operating system from an outside storage medium. Pseudocode for the boot loader might be as simple as the following eight instructions:

```
1. Set the P register to 9
2. Check paper tape reader ready
3. If not ready, jump to 2
4. Read a byte from paper tape reader to accumulator
5. Store accumulator to address in P register
6. If end of tape, jump to 9
7. Increment the P register
```

8. Jump to 2

A related example is based on a loader for a Nicolet Instrument Corporation minicomputer of the 1970s, using the paper tape reader-punch unit on a Teletype Model 33 ASR teleprinter. The bytes of its second-stage loader are read from paper tape in reverse order.

1. Set the P register to 106
2. Check paper tape reader ready
3. If not ready, jump to 2
4. Read a byte from paper tape reader to accumulator
5. Store accumulator to address in P register
6. Decrement the P register
7. Jump to 2

The length of the second stage loader is such that the final byte overwrites location 7. After the instruction in location 6 executes, location 7 starts the second stage loader executing. The second stage loader then waits for the much longer tape containing the operating system to be placed in the tape reader. The difference between the boot loader and second stage loader is the addition of checking code to trap paper tape read errors, a frequent occurrence with relatively low-cost, “part-time-duty” hardware, such as the Teletype Model 33 ASR. (Friden Flexowriters were far more reliable, but also comparatively costly.)

Booting the first microcomputers

The earliest microcomputers, such as the Altair 8800 and an even earlier, similar machine (based on the Intel 8008 CPU) had no bootstrapping hardware as such. When started, the CPU would see memory that would contain executable code containing only binary zeros—memory was cleared by resetting when powering up. The front panels of these machines carried toggle switches, one switch per bit of the computer memory word. Simple additions to the hardware permitted one memory location at a time to be loaded from those switches to store bootstrap code. Meanwhile, the CPU was kept from attempting to execute memory content. Once correctly loaded, the CPU was enabled to execute the bootstrapping code. This process was tedious and had to be error-free.

Integrated circuit read-only memory era

The boot process was revolutionized by the introduction of integrated circuit read-only memory (ROM), with its many variants, including mask-programmed ROMs, programmable ROMs (PROM), erasable programmable ROMs(EPROM), and flash memory. These allowed firmware boot programs to be shipped installed on the computer.

Typically, every microprocessor will, after a reset or power-on condition, perform a start-up process that usually takes the form of “begin execution of the code that is found starting at a specific address” or “look for a multibyte code at a specific address and jump to the indicated location to begin execution.” A system built using that microprocessor will have the permanent ROM occupying these special locations so that the system always begins operating without operator assistance. For example, Intel x86 processors always start by running the instructions beginning at FFFF:0000, while for the MOS 6502 processor, initialization begins by reading a two-byte vector address at $FFFD (MS byte) and $FFFC (LS byte) and jumping to that location to run the bootstrap code.

An Intel 2708 EPROM “chip” on a circuit board.

Apple Inc.'s first computer, the Apple 1 introduced in 1976, featured PROM chips that eliminated the need for a front panel for the boot process. According to Apple's ad announcing it "No More Switches, No More Lights … the firmware in PROMS enables you to enter, display and debug programs (all in hex) from the keyboard."

Due to the expense of read-only memory at the time, the Apple II series booted its disk operating systems using a series of very small incremental steps, each passing control onward to the next phase of the gradually more complex boot process. (See Apple DOS: Boot loader). Because so little of the disk operating system relied on ROM, the hardware was also extremely flexible and supported a wide range of customized disk copy protection mechanisms. (See Software Cracking: History.)

Some operating systems, most notably pre-1995 Macintosh systems from Apple, are so closely interwoven with their hardware that it is impossible to natively boot an operating system other than the standard one. This is the opposite extreme of the scenario using switches mentioned above; it is highly inflexible but relatively error-proof and foolproof as long as all hardware is working normally. A common solution in such situations is to design a boot loader that works as a program belonging to the standard OS that hijacks the system and loads the alternative OS. This technique was used by Apple for its A/UX Unix implementation and copied by various freeware operating systems and BeOS Personal Edition 5.

Some machines, like the Atari ST microcomputer, were "instant-on", with the operating system executing from a ROM. Retrieval of the OS from secondary or tertiary store was thus eliminated as one of the characteristic operations for bootstrapping. To allow system customizations, accessories, and other support software to be loaded automatically, the Atari's floppy drive was read for additional components during the boot process. There was a timeout delay that provided time to manually insert a floppy as the system searched for the extra components. This could be avoided by inserting a blank disk. The Atari ST hardware was also designed so the cartridge slot could provide native program execution for gaming purposes as a holdover from Atari's legacy making electronic games; by inserting the Spectre GCR cartridge with the Macintosh system ROM in the game slot and turning the Atari on, it could "natively boot" the Macintosh operating system rather than Atari's own TOS system.

The IBM Personal Computer included ROM-based firmware called the BIOS; one of the functions of that firmware was to perform a power-on self test when the machine was powered up, and then to read software from a boot device and execute it. Firmware compatible with the BIOS on the IBM Personal Computer is used in IBM PC compatible computers. The Extensible Firmware Interface was developed by Intel, originally for Itanium-based machines, and later also used as an alternative to the BIOS in x86-based machines, including Apple Macs using Intel processors.

Unix workstations originally had vendor-specific ROM-based firmware. Sun Microsystems later developed OpenBoot, later known as Open Firmware, which incorporated a Forth interpreter, with much of the firmware being written in Forth. It was standardized by the IEEE as IEEE standard 1275-1994; firmware that implements that standard was used in PowerPC-based Macs and some other PowerPC-based machines, as well as Sun's own SPARC-based computers. The Advanced RISC Computing specification defined another firmware standard, which was implemented on some MIPS-based and Alpha-based machines and the SGI Visual Workstation x86-based workstations.

Modern boot loaders

When a computer is turned off, its software—including operating systems, application code, and data—remains stored on nonvolatile data storage devices such as hard disk drives, CDs, DVDs, flash memory cards (SD cards, for example), USB flash drives, and floppy disks. When the computer is powered on, it typically does not have an operating system or its loader in random access memory (RAM). The computer first executes a relatively small program stored in read-only memory (ROM) along with a small amount of needed data, to access the nonvolatile device or devices from which the operating system programs and data can be loaded into RAM.

The small program that starts this sequence is known as a *bootstrap loader*, *bootstrap* or *boot loader*. This small program's only job is to load other data and programs which are then executed from RAM. Often, multiple-stage boot loaders are used, during which several programs of increasing complexity load one after the other in a process of chain loading.

Some computer systems, upon receiving a boot signal from a human operator or a peripheral device, may load a very small number of fixed instructions into memory at a specific location, initialize at least one CPU, and then point the CPU to the instructions and start their execution. These instructions typically start an input operation from some peripheral device (which may be switch-selectable by the operator). Other systems may send hardware commands directly to peripheral devices or I/O controllers that cause an extremely simple input operation (such as “read sector zero of the system device into memory starting at location 1000”) to be carried out, effectively loading a small number of boot loader instructions into memory; a completion signal from the I/O device may then be used to start execution of the instructions by the CPU.

Smaller computers often use less flexible but more automatic boot loader mechanisms to ensure that the computer starts quickly and with a predetermined software configuration. In many desktop computers, for example, the bootstrapping process begins with the CPU executing software contained in ROM (for example, the BIOS of an IBM PC) at a predefined address (some CPUs, including the Intel x86 series are designed to execute this software after reset without outside help). This software contains rudimentary functionality to search for devices eligible to participate in booting, and load a small program from a special section (most commonly the boot sector) of the most promising device, typically starting at a fixed entry point such as the start of the sector.

Boot loaders may face peculiar constraints, especially in size; for instance, on the IBM PC and compatibles, a boot sector should typically work in only 32 KB (later relaxed to 64 KB) of system memory and not use instructions not supported by the original 8088/8086 processors. The first stage of boot loaders located on fixed disks and removable drives must fit into the first 446 bytes of the Master Boot Record in order to leave room for the default 64-byte partition table with four partition entries and the two-byte boot signature, which the BIOS requires for a proper boot loader — or even less, when additional features like more than four partition entries (up to 16 with 16 bytes each), a disk signature (6 bytes), a disk timestamp (6 bytes), an Advanced Active Partition(18 bytes) or special multi-boot loaders have to be supported as well in some environments. In floppy and superfloppy Volume Boot Records, up to 59 bytes are occupied for the Extended BIOS Parameter Block on FAT12 and FAT16 volumes since DOS 4.0, whereas the FAT32 EBPB introduced with DOS 7.1 requires even 71 bytes, leaving only 441 bytes for the boot loader when assuming a sector size of 512 bytes. Microsoft boot sectors therefore traditionally imposed certain restrictions on the boot process, for example, the boot file had to be located at a fixed position in the root directory of the file system and stored as consecutive sectors, conditions taken care of by the `SYS` command and slightly relaxed in later versions of DOS. The boot loader was then able to load the first three sectors of the file into memory, which happened to contain another embedded boot loader able to load the remainder of the file into memory. When they added LBA and FAT32 support, they even switched to a two-sector boot loader using 386 instructions. At the same time other vendors managed to squeeze much more functionality into a single boot sector without relaxing the original constraints on the only minimal available memory and processor support. For example, DR-DOS boot sectors are able to locate the boot file in the FAT12, FAT16 and FAT32 file system, and load it into memory as a whole via CHS or LBA, even if the file is not stored in a fixed location and in consecutive sectors.

Second-stage boot loader

Second-stage boot loaders, such as GNU GRUB, BOOTMGR, Syslinux, NTLDR or BootX, are not themselves operating systems, but are able to load an operating system properly and transfer execution to it; the operating system subsequently initializes itself and may load extra device drivers. The second-stage boot loader does not need drivers for its own operation, but may instead use generic storage access methods provided by system firmware such as the BIOS or Open Firmware, though typically with restricted hardware functionality and lower performance.

Many boot loaders (like GNU GRUB, Windows’s BOOTMGR, and Windows NT/2000/XP’s NTLDR) can be configured to give the user multiple booting choices. These choices can include different operating systems (for dual or multi-booting from different partitions or drives), different versions of the same operating system (in case a new version has unexpected problems), different operating system loading options (e.g., booting into a rescue or safe mode), and some standalone programs that can function without an operating system, such as memory testers (e.g., memtest86+) or even games (see List of PC Booter games). Some boot loaders can also load other boot loaders; for example, GRUB loads BOOTMGR instead of loading Windows directly. Usually a default choice is preselected with a time delay during which a user can press a key to change the choice; after this delay, the default choice is automatically run so normal booting can occur without interaction.

The boot process can be considered complete when the computer is ready to interact with the user, or the operating system is capable of running system programs or application programs. Typical modern personal computers boot in about one minute, of which about 15 seconds are taken by a power-on self-test(POST) and a preliminary boot loader, and the rest by loading the operating system and other software. Time spent after the operating system loading can be considerably shortened to as little as 3 seconds[24] by bringing the system up with all cores at once, as with coreboot. Large servers may take several minutes to boot and start all their services.

Many embedded systems must boot immediately. For example, waiting a minute for a digital television or a GPS navigation device to start is generally unacceptable. Therefore, such devices have software systems in ROM or flash memory so the device can begin functioning immediately; little or no loading is necessary, because the loading can be precomputed and stored on the ROM when the device is made.

Large and complex systems may have boot procedures that proceed in multiple phases until finally the operating system and other programs are loaded and ready to execute. Because operating systems are designed as if they never start or stop, a boot loader might load the operating system, configure itself as a mere process within that system, and then irrevocably transfer control to the operating system. The boot loader then terminates normally as any other process would.

Network booting

Most computers are also capable of booting over a computer network. In this scenario, the operating system is stored on the disk of a server, and certain parts of it are transferred to the client using a simple protocol such as the Trivial File Transfer Protocol (TFTP). After these parts have been transferred, the operating system takes over the control of the booting process.

As with the second-stage boot loader, network booting begins by using generic network access methods provided by the network interface's boot ROM, which typically contains a Preboot Execution Environment (PXE) image. No drivers are required, but the system functionality is limited until the operating system kernel and drivers are transferred and started. As a result, once the ROM-based booting has completed it is entirely possible to network boot into an operating system that itself does not have the ability to use the network interface.

Personal computers (PC)

Boot devices

The boot device is the device from which the operating system is loaded. A modern PC BIOS supports booting from various devices, typically a local hard disk drive via the Master Boot Record (MBR) (and of several MS-DOS partitionso n such a disk, or GPT through GRUB 2), an optical disc drive (using El Torito), a USB mass storage device (FTL-based flash drive, SD card, or multi-media card slot; hard disk drive, optical disc drive, etc.), or a network interface card (using PXE). Older, less common BIOS-bootable devices include floppy disk drives, SCSI devices, Zip drives, and LS-120 drives.

Windows To Go bootable flash drive, a Live USB example

Typically, the BIOS will allow the user to configure a *boot order*. If the boot order is set to "first, the DVD drive; second, the hard disk drive", then the BIOS will try to boot from the DVD drive, and if this fails (e.g. because there is no DVD in the drive), it will try to boot from the local hard drive.

For example, on a PC with Windows XP installed on the hard drive, the user could set the boot order to the one given above, and then insert a Linux Live CD in order to try out Linux without having to install an operating system onto the hard drive. This is an example of dual booting, in which the user chooses which operating system to start after the computer has performed its Power-on self-test (POST). In this example of dual booting, the user chooses by inserting or removing the CD from the computer, but it is more common to choose which operating system to boot by selecting from a BIOS or UEFI boot menu, by using the computer keyboard; the boot menu is typically entered by pressing `Delete` or `F11` keys during the POST.

Several devices are available that enable the user to *quick-boot* into what is usually a variant of Linux for various simple tasks such as Internet access; examples are Splashtop and Latitude ON.

Boot sequence

A hex dump of FreeBSD's boot0 MBR

Upon starting, an IBM-compatible personal computer's x86 CPU executes, in real mode, the instruction located at reset vector (the physical memory address FFFF0h on 16-bit x86 processors and FFFFFFF0h on 32-bit and 64-bit x86 processors), usually pointing to the BIOS entry point inside the ROM. This memory location typically contains a jump instruction that transfers execution to the location of the BIOS start-up program. This program runs a power-on self-test (POST) to check and initialize required devices such as DRAM and the PCI bus (including running embedded ROMs). The most complicated step is setting up DRAM over SPI, made more difficult by the fact that at this point memory is very limited.

After initializing required hardware, the BIOS goes through a pre-configured list of non-volatile storage devices ("boot device sequence") until it finds one that is bootable. A bootable device is defined as one that can be read from, and where the last two bytes of the first sector contain the little-endian word AA55h, found as byte sequence 55h, AAh on disk (also known as the MBR boot signature), or where it is otherwise established that the code inside the sector is executable on x86 PCs.

Coreboot splits the initialization and boot services into distinct parts, supporting "payloads" such as SeaBIOS,TianoCore, GRUB, and Linux directly (from flash).

Once the BIOS has found a bootable device it loads the boot sector to linear address 7C00h (usually segment:offset0000h:7C00h, but some BIOSes erroneously use 07C0h:0000h) and transfers execution to the boot code. In the case of a hard disk, this is referred to as the Master Boot Record (MBR) and is by definition not operating-system specific. The conventional MBR code checks the MBR's partition table for a partition set as *bootable* (the one with *active* flag set). If an active partition is found, the MBR code loads the boot sector code from that partition, known as Volume Boot Record (VBR), and executes it.

The VBR is often operating-system specific; however, in most operating systems its main function is to load and execute the operating system kernel, which continues startup.

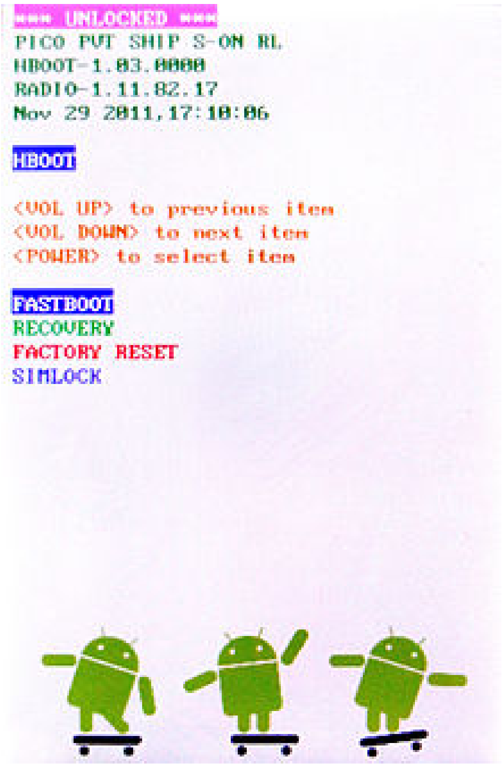
If there is no active partition, or the active partition's boot sector is invalid, the MBR may load a secondary boot loader which will select a partition (often via user input) and load its boot sector, which usually loads the corresponding operating system kernel. In some cases, the MBR may also attempt to load secondary boot loaders before trying to boot the active partition. If all else fails, it should issue an INT 18h BIOS interrupt call (followed by an INT 19h just in case INT 18h would return) in order to give back control to the BIOS, which would then attempt to boot off other devices, attempt a remote boot via network or invoke ROM BASIC.

Some systems (particularly newer Macintoshes and new editions of Microsoft Windows) use Intel's EFI. Also coreboot allows a computer to boot without having the firmware/BIOS constantly running in system management mode. 16-bit BIOS interfaces are required by certain x86 operating systems, such as DOS and Windows 3.1/95/98 (and all when not booted via UEFI). However, most boot loaders retain 16-bit BIOS call support.

Other kinds of boot sequences

Some modern CPUs and microcontrollers (for example, TI OMAP) or sometimes even DSPs may have boot ROM with boot code integrated directly into their silicon, so such a processor could perform quite a sophisticated boot sequence on its own and load boot programs from various sources like NAND flash, SD or MMC card and so on. It is hard to hardwire all the required logic for handling such devices, so an integrated boot ROM is used instead in such scenarios. Boot ROM usage enables more flexible boot sequences than hardwired logic could provide. For example, the boot ROM could try to perform boot from multiple boot sources. Also, a boot ROM is often able to load a boot loader or diagnostic program via serial interfaces like UART, SPI, USB and so on. This feature is often used for system recovery purposes when for some reasons usual boot software in non-volatile memory got erased, and it could also be used for initial non-volatile memory programming when there is clean non-volatile memory installed and hence no software available in the system yet.

Some embedded system designs may also include an intermediary boot sequence step in form of additional code that gets loaded into system RAM by the integrated boot ROM. Additional code loaded that way usually serves as a way for overcoming platform limitations, such as small amounts of RAM, so a dedicated primary boot loader, such as Das U-Boot, can be loaded as the next step in system's boot sequence. The additional code and boot sequence step are usually referred to as *secondary program loader* (SPL).



```
*** UNLOCKED ***
PICO PVT SHIP S-ON RL
HBOOT-1.03.0000
RADIO-1.11.02.17
Nov 29 2011,17:10:06

HBOOT

<VOL UP> to previous item
<VOL DOWN> to next item
<POWER> to select item

FASTBOOT
RECOVERY
FACTORY RESET
SIMLOCK
```

An unlocked Android bootloader, showing additional available options

It is also possible to take control of a system by using a hardware debug interface such as JTAG. Such an interface may be used to write the boot loader program into bootable non-volatile memory (e.g. flash) by instructing the processor core to perform the necessary actions to program non-volatile memory. Alternatively, the debug interface may be used to upload some diagnostic or boot code into RAM, and then to start the processor core and instruct it to execute the uploaded code. This allows, for example, the recovery of embedded systems where no software remains on any supported boot device, and where the processor does not have any integrated boot ROM. JTAG is a standard and popular interface; many CPUs, microcontrollers and other devices are manufactured with JTAG interfaces (as of 2009).

Some microcontrollers provide special hardware interfaces which cannot be used to take arbitrary control of a system or directly run code, but instead they allow the insertion of boot code into bootable non-volatile memory (like flash memory) via simple protocols. Then at the manufacturing phase, such interfaces are used to inject boot code (and possibly other code) into non-volatile memory. After system reset, the microcontroller begins to execute code programmed into its non-volatile memory, just like usual processors are using ROMs for booting. Most notably this technique is used by Atmel AVR microcontrollers, and by others as well. In many cases such interfaces are implemented by hardwired logic. In other cases such interfaces could be created by software running in integrated on-chip boot ROM from GPIO pins.

Most digital signal processors have a serial mode boot, and a parallel mode boot, such as the host port interface (HPI boot)

In case of DSPs there is often a second microprocessor or microcontroller present in the system design, and this is responsible for overall system behavior, interrupt handling, dealing with external events, user interface, etc. while the DSP is dedicated to signal processing tasks only. In such systems the DSP could be booted by another processor which is sometimes referred as the *host processor* (giving name to a Host Port). Such a processor is also sometimes referred as the *master*, since it usually boots first from its own memories and then controls overall system behavior, including booting of the DSP, and then further controlling the DSP's behavior. The DSP often lacks its own boot memories and relies on the host processor to supply the required code instead. The most notable systems with such a design are cell phones, modems, audio and video players and so on, where a DSP and a CPU/microcontroller are co-existing.

Many FPGA chips load their configuration from an external serial EEPROM (“configuration ROM”) on power-up.

Licensing & Attributions

CC licensed content, Shared previously

- Booting. **Provided by:** Wikipedia. **Located at:** https://en.wikipedia.org/wiki/Booting. **License:** *CC BY-SA: Attribution-ShareAlike*
- Booting up. **Authored by:** Nick Johnson. **Located at:** https://www.flickr.com/photos/npj/4836485286/. **License:** *CC BY: Attribution*

READING: OPERATING SYSTEM

Introduction

An **operating system** (**OS**) is software that manages computer hardware and software resources and provides common services for computer programs. The operating system is an essential component of the system software in a computer system. Application programs usually require an operating system to function.

Time-sharing operating systems schedule tasks for efficient use of the system and may also include accounting software for cost allocation of processor time, mass storage, printing, and other resources.

For hardware functions such as input and output and memory allocation, the operating system acts as an intermediary between programs and the computer hardware, although the application code is usually executed directly by the hardware and frequently makes system calls to an OS function or be interrupted by it. Operating systems are found on many devices that contain a computer—from cellular phones and video game consoles to web servers and supercomputers.

Examples of popular modern operating systems include Android, BlackBerry, BSD, Chrome OS, iOS, Linux, OS X, QNX, Microsoft Windows, Windows Phone, and z/OS. The first eight of these examples share roots inUNIX. Popular hard real-time operating systems include FreeRTOS, Micrium and VxWorks.

Types of operating systems

Single- and multi-tasking

A single-tasking system can only run one program at a time, while a multi-tasking operating system allows more than one program to be running in concurrency. This is achieved by time-sharing, dividing the available processor time between multiple processes which are each interrupted repeatedly in time-slices by a task scheduling subsystem of the operating system. Multi-tasking may be characterized in pre-emptive and co-operative types. In pre-emptive multitasking, the operating system slices the CPU time and dedicates a slot to each of the programs. Unix-like operating systems, e.g., Solaris, Linux, as well as AmigaOS support pre-emptive multitasking. Cooperative multitasking is achieved by relying on each process to provide time to the other processes in a defined manner. 16-bit versions of Microsoft Windows used cooperative multi-tasking. 32-bit versions of both Windows NT and Win9x, used pre-emptive multi-tasking.

Single- and multi-user

Single-user operating systems have no facilities to distinguish users, but may allow multiple programs to run in tandem. A multi-user operating system extends the basic concept of multi-tasking with facilities that identify processes and resources, such as disk space, belonging to multiple users, and the system permits multiple users to interact with the system at the same time. Time-sharing operating systems schedule tasks for efficient use of the system and may also include accounting software for cost allocation of processor time, mass storage, printing, and other resources to multiple users.

Distributed

A distributed operating system manages a group of distinct computers and makes them appear to be a single computer. The development of networked computers that could be linked and communicate with each other gave rise to distributed computing. Distributed computations are carried out on more than one machine. When computers in a group work in cooperation, they form a distributed system.

Templated

In an OS, distributed and cloud computing context, templating refers to creating a single virtual machine image as a guest operating system, then saving it as a tool for multiple running virtual machines (Gagne, 2012, p. 716). The technique is used both in virtualization and cloud computing management, and is common in large server warehouses.

Embedded

Embedded operating systems are designed to be used in embedded computer systems. They are designed to operate on small machines like PDAs with less autonomy. They are able to operate with a limited number of resources. They are very compact and extremely efficient by design. Windows CE and Minix 3 are some examples of embedded operating systems.

Real-time

A real-time operating system is an operating system that guaranties to process events or data within a certain short amount of time. A real-time operating system may be single- or multi-tasking, but when multitasking, it uses specialized scheduling algorithms so that a deterministic nature of behavior is achieved. An event-driven system switches between tasks based on their priorities or external events while time-sharing operating systems switch tasks based on clock interrupts.

History

Early computers were built to perform a series of single tasks, like a calculator. Basic operating system features were developed in the 1950s, such as resident monitor functions that could automatically run different programs in succession to speed up processing. Operating systems did not exist in their modern and more complex forms until the early 1960s. Hardware features were added, that enabled use of runtime libraries, interrupts, and parallel processing. When personal computers became popular in the 1980s, operating systems were made for them similar in concept to those used on larger computers.

In the 1940s, the earliest electronic digital systems had no operating systems. Electronic systems of this time were programmed on rows of mechanical switches or by jumper wires on plug boards. These were special-purpose systems that, for example, generated ballistics tables for the military or controlled the printing of payroll checks from data on punched paper cards. After programmable general purpose computers were invented, machine languages (consisting of strings of the binary digits 0 and 1 on punched paper tape) were introduced that sped up the programming process (Stern, 1981).

In the early 1950s, a computer could execute only one program at a time. Each user had sole use of the computer for a limited period of time and would arrive at a scheduled time with program and data on punched paper cards and/or punched tape. The program would be loaded into the machine, and the machine would be set to work until the program completed or crashed. Programs could generally be debugged via a front panel using toggle switches and panel lights. It is said that Alan Turing was a master of this on the early Manchester Mark 1 machine, and he was already deriving the primitive conception of an operating system from the principles of the Universal Turing machine.

Later machines came with libraries of programs, which would be linked to a user's program to assist in operations such as input and output and generating computer code from human-readable symbolic code. This was the genesis of the modern-day operating system. However, machines still ran a single job at a time. At Cambridge University in England the job queue was at one time a washing line from which tapes were hung with different colored clothes-pegs to indicate job-priority.

An improvement was the Atlas Supervisor introduced with the Manchester Atlas commissioned in 1962, 'considered by many to be the first recognizable modern operating system'. Brinch Hansen described it as "the most significant breakthrough in the history of operating systems."

OS/360 was used on most IBM mainframe computers beginning in 1966, including computers used by the Apollo program.

Mainframes

Through the 1950s, many major features were pioneered in the field of operating systems, including batch processing, input/output interrupt, buffering,multitasking, spooling, runtime libraries, link-loading, and programs for sorting records in files. These features were included or not included in application software at the option of application programmers, rather than in a separate operating system used by all applications. In 1959, the SHARE Operating System was released as an integrated utility for the IBM 704, and later in the 709 and 7090 mainframes, although it was quickly supplanted by IBSYS/IBJOB on the 709, 7090 and 7094.

During the 1960s, IBM's OS/360 introduced the concept of a single OS spanning an entire product line, which was crucial for the success of the System/360 machines. IBM's current mainframe operating systems are distant descendants of this original system and applications written for OS/360 can still be run on modern machines.

OS/360 also pioneered the concept that the operating system keeps track of all of the system resources that are used, including program and data space allocation in main memory and file space in secondary storage, and file locking during update. When the process is terminated for any reason, all of these resources are re-claimed by the operating system.

The alternative CP-67 system for the S/360-67 started a whole line of IBM operating systems focused on the concept of virtual machines. Other operating systems used on IBM S/360 series mainframes included systems developed by IBM: COS/360 (Compatibility Operating System), DOS/360 (Disk Operating System), TSS/360 (Time Sharing System), TOS/360 (Tape Operating System), BOS/360 (Basic Operating System), and ACP (Airline Control Program), as well as a few non-IBM systems: MTS (Michigan Terminal System), MUSIC (Multi-User System for Interactive Computing), and ORVYL (Stanford Timesharing System).

Control Data Corporation developed the SCOPE operating system in the 1960s, for batch processing. In cooperation with the University of Minnesota, the Kronos and later the NOS operating systems were developed during the 1970s, which supported simultaneous batch and timesharing use. Like many commercial timesharing systems, its interface was an extension of the Dartmouth BASIC operating systems, one of the pioneering efforts in timesharing and programming languages. In the late 1970s, Control Data and the University of Illinois developed the PLATO operating system, which used plasma panel displays and long-distance time sharing networks. Plato was remarkably innovative for its time, featuring real-time chat, and multi-user graphical games.

In 1961, Burroughs Corporation introduced the B5000 with the MCP, (Master Control Program) operating system. The B5000 was a stack machine designed to exclusively support high-level languages with no machine language or assembler, and indeed the MCP was the first OS to be written exclusively in a high-level language – ESPOL, a dialect of ALGOL. MCP also introduced many other ground-breaking innovations, such as being the first commercial implementation of virtual memory. During development of the AS400, IBM made an approach to Burroughs to license MCP to run on the AS400 hardware. This proposal was declined by Burroughs management to protect its existing hardware production. MCP is still in use today in the Unisys ClearPath/MCP line of computers.

UNIVAC, the first commercial computer manufacturer, produced a series of EXEC operating systems. Like all early main-frame systems, this batch-oriented system managed magnetic drums, disks, card readers and line printers. In the 1970s, UNIVAC produced the Real-Time Basic (RTB) system to support large-scale time sharing, also patterned after the Dartmouth BC system.

General Electric and MIT developed General Electric Comprehensive Operating Supervisor (GECOS), which introduced the concept of ringed security privilege levels. After acquisition by Honeywell it was renamed General Comprehensive Operating System (GCOS).

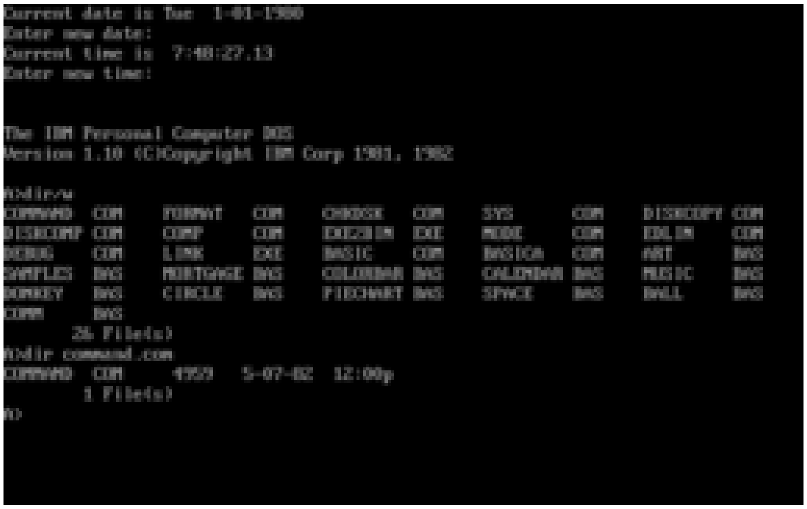
Digital Equipment Corporation developed many operating systems for its various computer lines, including TOPS-10 and TOPS-20 time sharing systems for the 36-bit PDP-10 class systems. Before the widespread use of UNIX, TOPS-10 was a particularly popular system in universities, and in the early ARPANET community.

From the late 1960s through the late 1970s, several hardware capabilities evolved that allowed similar or ported software to run on more than one system. Early systems had utilized microprogramming to implement features on their systems in order to permit different underlying computer architectures to appear to be the same as others in a series. In fact, most 360s after the 360/40 (except the 360/165 and 360/168) were microprogrammed implementations.

The enormous investment in software for these systems made since the 1960s caused most of the original computer manufacturers to continue to develop compatible operating systems along with the hardware. Notable supported mainframe operating systems include:

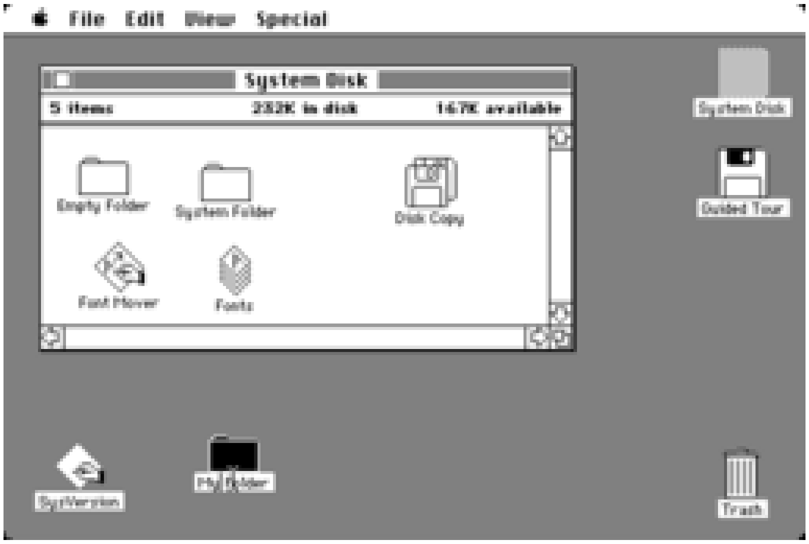
- Burroughs MCP – B5000, 1961 to Unisys Clearpath/MCP, present
- IBM OS/360 – IBM System/360, 1966 to IBM z/OS, present
- IBM CP-67 – IBM System/360, 1967 to IBM z/VM
- UNIVAC EXEC 8 – UNIVAC 1108, 1967, to OS 2200 Unisys Clearpath Dorado, present

Microcomputers



```
Current date is Tue  1-01-1980
Enter new date:
Current time is  7:48:27.13
Enter new time:


The IBM Personal Computer DOS
Version 1.10 (C)Copyright IBM Corp 1981, 1982

A>dir/w
COMMAND  COM    FORMAT   COM    CHKDSK   COM    SYS      COM    DISKCOPY COM
DISKCOMP COM    COMP     COM    EXE2BIN  EXE    MODE     COM    EDLIN    COM
DEBUG    COM    LINK     EXE    BASIC    COM    BASICA   COM    ART      BAS
SAMPLES  BAS    MORTGAGE BAS    COLORBAR BAS    CALENDAR BAS    MUSIC    BAS
DONKEY   BAS    CIRCLE   BAS    PIECHART BAS    SPACE    BAS    BALL     BAS
COMM     BAS
       26 File(s)
A>dir command.com
COMMAND  COM     4959   5-07-82  12:00p
        1 File(s)
A>
```

PC DOS was an early personal computer OS that featured a command line interface.



Mac OS by Apple Computer became the first widespread OS to feature a graphical user interface. Many of its features such as windows and icons would later become commonplace in GUIs.

The first microcomputers did not have the capacity or need for the elaborate operating systems that had been developed for mainframes and minis; minimalistic operating systems were developed, often loaded from ROM and known as *monitors*. One notable early disk operating system was CP/M, which was supported on many early microcomputers and was closely imitated by Microsoft's MS-DOS, which became widely popular as the operating system chosen for the IBM PC (IBM's version of it was called IBM DOS or PC DOS). In the 1980s, Apple Computer Inc. (now Apple Inc.) abandoned its popular Apple II series of microcomputers to introduce theApple Macintosh computer with an innovative Graphical User Interface (GUI) to the Mac OS.

The introduction of the Intel 80386 CPU chip with 32-bit architecture and paging capabilities, provided personal computers with the ability to run multitasking operating systems like those of earlier minicomputers and mainframes. Microsoft responded to this progress by hiring Dave Cutler, who had developed the VMS operating system for Digital Equipment Corporation. He would lead the development of the Windows NT operating system, which continues to serve as the basis for Microsoft's operating systems line. Steve Jobs, a co-founder of Apple Inc., started NeXT Computer Inc., which developed the NEXTSTEP operating system. NEXTSTEP would later be acquired by Apple Inc. and used, along with code from FreeBSD as the core of Mac OS X.

The GNU Project was started by activist and programmer Richard Stallman with the goal of creating a complete free software replacement to the proprietary UNIX operating system. While the project was highly successful in duplicating the functionality of various parts of UNIX, development of the GNU Hurd kernel proved to be unproductive. In 1991, Finnish computer science student Linus Torvalds, with cooperation from volunteers collaborating over the Internet, released the first version of the Linux kernel. It was soon merged with the GNU user space components and system software to form a complete operating system. Since then, the combination of the two major components has usually been referred to as simply "Linux" by the software industry, a naming convention that Stallman and the Free Software Foundation remain opposed to, preferring the name GNU/Linux. The Berkeley Software Distribution, known as BSD, is the UNIX derivative distributed by the University of California, Berkeley, starting in the 1970s. Freely distributed and ported to many minicomputers, it eventually also gained a following for use on PCs, mainly as FreeBSD, NetBSD and OpenBSD.

Examples of operating systems

Unix and Unix-like operating systems

Evolution of Unix systems

Unix was originally written in assembly language. Ken Thompson wrote B, mainly based on BCPL, based on his experience in the MULTICS project. B was replaced by C, and Unix, rewritten in C, developed into a large, complex family of inter-related operating systems which have been influential in every modern operating system (see History).

The *Unix-like* family is a diverse group of operating systems, with several major sub-categories including System V, BSD, and Linux. The name “UNIX” is a trademark of The Open Group which licenses it for use with any operating system that has been shown to conform to their definitions. “UNIX-like” is commonly used to refer to the large set of operating systems which resemble the original UNIX.

Unix-like systems run on a wide variety of computer architectures. They are used heavily for servers in business, as well as workstations in academic and engineering environments. Free UNIX variants, such as Linux and BSD, are popular in these areas.

Four operating systems are certified by The Open Group (holder of the Unix trademark) as Unix. HP’s HP-UX and IBM’s AIX are both descendants of the original System V Unix and are designed to run only on their respective vendor’s hardware. In contrast, Sun Microsystems’s Solaris Operating System can run on multiple types of hardware, including x86 and Sparc servers, and PCs. Apple’s OS X, a replacement for Apple’s earlier (non-Unix) Mac OS, is a hybrid kernel-based BSD variant derived from NeXTSTEP, Mach, and FreeBSD.

Unix interoperability was sought by establishing the POSIX standard. The POSIX standard can be applied to any operating system, although it was originally created for various Unix variants.

BSD and its descendants

The first server for the World Wide Webran on NeXTSTEP, based on BSD

A subgroup of the Unix family is the Berkeley Software Distribution family, which includes FreeBSD, NetBSD, and OpenBSD. These operating systems are most commonly found on webservers, although they can also function as a personal computer OS. The Internet owes much of its existence to BSD, as many of the protocols now commonly used by computers to connect, send and receive data over a network were widely implemented and refined in BSD. The World Wide Web was also first demonstrated on a number of computers running an OS based on BSD called NextStep.

BSD has its roots in Unix. In 1974, University of California, Berkeley installed its first Unix system. Over time, students and staff in the computer science department there began adding new programs to make things easier, such as text editors. When Berkeley received new VAX computers in 1978 with Unix installed, the school's undergraduates modified Unix even more in order to take advantage of the computer's hardware possibilities. The Defense Advanced Research Projects Agency of the US Department of Defense took interest, and decided to fund the project. Many schools, corporations, and government organizations took notice and started to use Berkeley's version of Unix instead of the official one distributed by AT&T.

Steve Jobs, upon leaving Apple Inc. in 1985, formed NeXT Inc., a company that manufactured high-end computers running on a variation of BSD calledNeXTSTEP. One of these computers was used by Tim Berners-Lee as the first webserver to create the World Wide Web.

Developers like Keith Bostic encouraged the project to replace any non-free code that originated with Bell Labs. Once this was done, however, AT&T sued. Eventually, after two years of legal disputes, the BSD project came out ahead and spawned a number of free derivatives, such as FreeBSD and NetBSD.

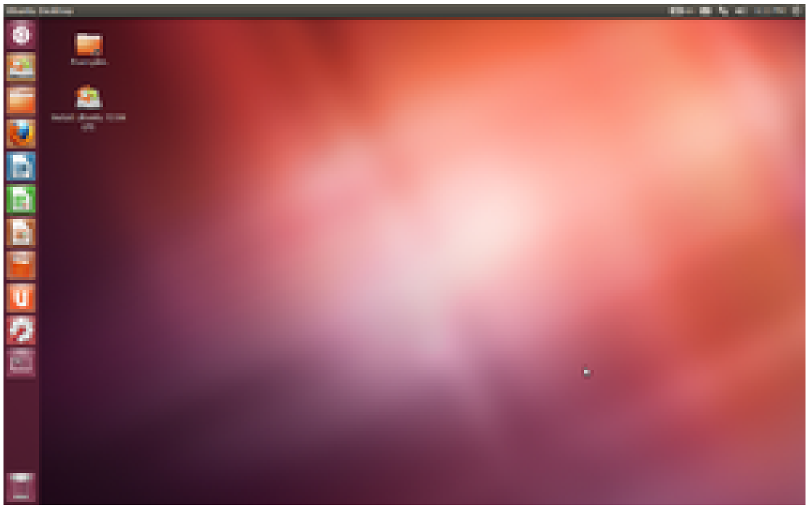
OS X

The standard user interface of OS X

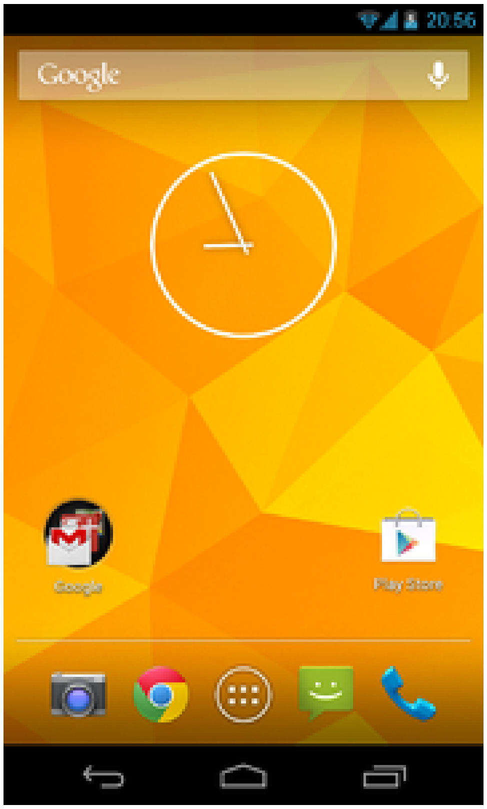
OS X (formerly "Mac OS X") is a line of open core graphical operating systems developed, marketed, and sold by Apple Inc., the latest of which is pre-loaded on all currently shipping Macintosh computers. OS X is the successor to the original Mac OS, which had been Apple's primary operating system since 1984. Unlike its predecessor, OS X is a UNIX operating system built on technology that had been developed at NeXT through the second half of the 1980s and up until Apple purchased the company in early 1997. The operating system was first released in 1999 as Mac OS X Server 1.0, with a desktop-oriented version (Mac OS X v10.0 "Cheetah") following in March 2001. Since then, six more distinct "client" and "server" editions of OS X have been released, until the two were merged in OS X 10.7 "Lion". Releases of OS X v10.0 through v10.8 are named after big cats. Starting with v10.9, "Mavericks", OS X versions are named after inspirational places inCalifornia. OS X 10.10 "Yosemite", the most recent version, was announced and released on 2 June 2014 at the WWDC 2014.

Prior to its merging with OS X, the server edition – OS X Server – was architecturally identical to its desktop counterpart and usually ran on Apple's line of Macintosh server hardware. OS X Server included work group management and administration software tools that provide simplified access to key network services, including a mail transfer agent, a Samba server, an LDAP server, a domain name server, and others. With Mac OS X v10.7 Lion, all server aspects of Mac OS X Server have been integrated into the client version and the product re-branded as "OS X" (dropping "Mac" from the name). The server tools are now offered as an application.

Linux



Ubuntu, desktop Linux distribution



Android, a popular mobile operating system using a modified version of the Linux kernel

The Linux kernel originated in 1991 as a side project of Linus Torvalds, while a university student in Finland. He posted information about his project on a newsgroup for computer students and programmers, and received support and assistance from volunteers who succeeded in creating a complete and functional kernel.

Linux is Unix-like, but was developed without any Unix code, unlike BSD and its variants. Because of its open license model, the Linux kernel code is available for study and modification, which resulted in its use on a wide range of computing machinery from supercomputers to smart-watches. Although estimates suggest that Linux is used on only 1.82% of all personal computers, it has been widely adopted for use in servers and embedded systems such as cell phones. Linux has superseded Unix on many platforms and is used on the ten most powerful supercomputers in the world. The Linux kernel is used in some popular distributions, such as Red Hat, Debian, Ubuntu, Linux Mint and Google's Android.

Google Chromium OS

Chromium is an operating system based on the Linux kernel and designed by Google. Since Chromium OS targets computer users who spend most of their time on the Internet, it is mainly a web browser with limited ability to run local applications, though it has a built-in file manager and media player. Instead, it relies on Internet applications (or Web apps) used in the web browser to accomplish tasks such as word processing. Chromium OS differs from Chrome OS in that Chromium is open-source and used primarily by developers whereas Chrome OS is the operating system shipped out in Chromebooks.

Microsoft Windows

Microsoft Windows is a family of proprietary operating systems designed by Microsoft Corporation and primarily targeted to Intel architecture based computers, with an estimated 88.9 percent total usage share on Web connected computers. The newest version is Windows 8.1 for workstations and Windows Server 2012 R2 for servers. Windows 7 recently overtook Windows XP as most used OS.

Microsoft Windows originated in 1985, as an operating environment running on top of MS-DOS, which was the standard operating system shipped on most Intel architecture personal computers at the time. In 1995, Windows 95 was released which only used MS-DOS as a bootstrap. For backwards compatibility, Win9x could run real-mode MS-DOS and 16 bits Windows 3.x drivers. Windows ME, released in 2000, was the last version in the Win9x family. Later versions have all been based on the Windows NT kernel. Current client versions of Windows run on IA-32, x86-64 and 32-bit ARM microprocessors. In addition Itanium is still supported in older server version Windows Server 2008 R2. In the past, Windows NT supported additional architectures.

Server editions of Windows are widely used. In recent years, Microsoft has expended significant capital in an effort to promote the use of Windows as a server operating system. However, Windows' usage on servers is not as widespread as on personal computers, as Windows competes against Linux and BSD for server market share. The first PC that used windows operating system was the IBM Personal System/2.

Other

There have been many operating systems that were significant in their day but are no longer so, such as AmigaOS; OS/2 from IBM and Microsoft; Mac OS, the non-Unix precursor to Apple's Mac OS X; BeOS; XTS-300; RISC OS; MorphOS; Haiku; BareMetal and FreeMint. Some are still used in niche markets and continue to be developed as minority platforms for enthusiast communities and specialist applications. OpenVMS, formerly from DEC, is still under active development by Hewlett-Packard. Yet other operating systems are used almost exclusively in academia, for operating systems education or to do research on operating system concepts. A typical example of a system that fulfills both roles is MINIX, while for example Singularity is used purely for research.

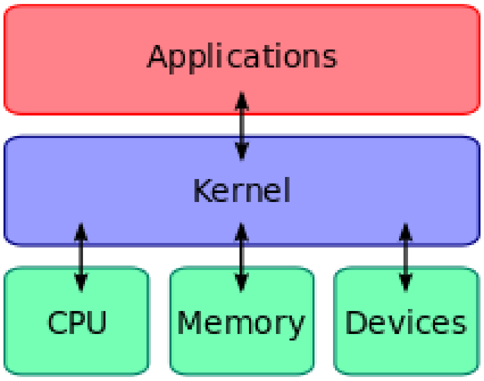
Other operating systems have failed to win significant market share, but have introduced innovations that have influenced mainstream operating systems, not least Bell Labs' Plan 9.

Components

The components of an operating system all exist in order to make the different parts of a computer work together. All user software needs to go through the operating system in order to use any of the hardware, whether it be as simple as a mouse or keyboard or as complex as an Internet component.

Kernel

With the aid of the firmware and device drivers, the kernel provides the most basic level of control over all of the computer's hardware devices. It manages memory access for programs in the RAM, it determines which programs get access to which hardware resources, it sets up or resets the CPU's operating states for optimal operation at all times, and it organizes the data for long-term non-volatile storage with file systems on such media as disks, tapes, flash memory, etc.



A kernel connects the application software to the hardware of a computer.

Program execution

The operating system provides an interface between an application program and the computer hardware, so that an application program can interact with the hardware only by obeying rules and procedures programmed into the operating system. The operating system is also a set of services which simplify development and execution of application programs. Executing an application program involves the creation of a process by the operating system kernel which assigns memory space and other resources, establishes a priority for the process in multi-tasking systems, loads program binary code into memory, and initiates execution of the application program which then interacts with the user and with hardware devices.

Interrupts

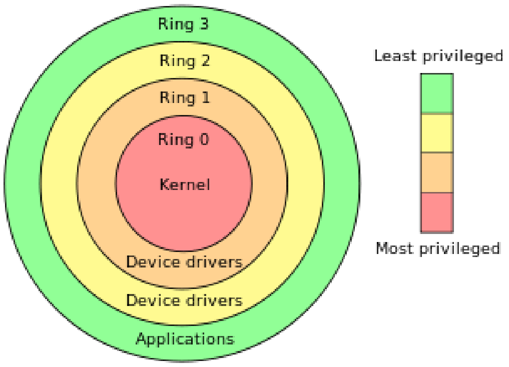
Interrupts are central to operating systems, as they provide an efficient way for the operating system to interact with and react to its environment. The alternative—having the operating system "watch" the various sources of input for events (polling) that require action—can be found in older systems with very small stacks (50 or 60 bytes) but is unusual in modern systems with large stacks. Interrupt-based programming is directly supported by most modern CPUs. Interrupts provide a computer with a way of automatically saving local register contexts, and running specific code in response to events. Even very basic computers support hardware interrupts, and allow the programmer to specify code which may be run when that event takes place.

When an interrupt is received, the computer's hardware automatically suspends whatever program is currently running, saves its status, and runs computer code previously associated with the interrupt; this is analogous to placing a bookmark in a book in response to a phone call. In modern operating systems, interrupts are handled by the operating system's kernel. Interrupts may come from either the computer's hardware or the running program.

When a hardware device triggers an interrupt, the operating system's kernel decides how to deal with this event, generally by running some processing code. The amount of code being run depends on the priority of the interrupt (for example: a person usually responds to a smoke detector alarm before answering the phone). The processing of hardware interrupts is a task that is usually delegated to software called a device driver, which may be part of the operating system's kernel, part of another program, or both. Device drivers may then relay information to a running program by various means.

A program may also trigger an interrupt to the operating system. If a program wishes to access hardware, for example, it may interrupt the operating system's kernel, which causes control to be passed back to the kernel. The kernel then processes the request. If a program wishes additional resources (or wishes to shed resources) such as memory, it triggers an interrupt to get the kernel's attention.

Modes



Privilege rings for the x86 available in protected mode. Operating systems determine which processes run in each mode.

Modern CPUs support multiple modes of operation. CPUs with this capability use at least two modes:protected mode and supervisor mode. The supervisor mode is used by the operating system's kernel for low level tasks that need unrestricted access to hardware, such as controlling how memory is written and erased, and communication with devices like graphics cards. Protected mode, in contrast, is used for almost everything else. Applications operate within protected mode, and can only use hardware by communicating with the kernel, which controls everything in supervisor mode. CPUs might have other modes similar to protected mode as well, such as the virtual modes in order to emulate older processor types, such as 16-bit processors on a 32-bit one, or 32-bit processors on a 64-bit one.

When a computer first starts up, it is automatically running in supervisor mode. The first few programs to run on the computer, being the BIOS or EFI, bootloader, and the operating system have unlimited access to hardware – and this is required because, by definition, initializing a protected environment can only be done outside of one. However, when the operating system passes control to another program, it can place the CPU into protected mode.

In protected mode, programs may have access to a more limited set of the CPU's instructions. A user program may leave protected mode only by triggering an interrupt, causing control to be passed back to the kernel. In this way the operating system can maintain exclusive control over things like access to hardware and memory.

The term "protected mode resource" generally refers to one or more CPU registers, which contain information that the running program isn't allowed to alter. Attempts to alter these resources generally causes a switch to supervisor mode, where the operating system can deal with the illegal operation the program was attempting (for example, by killing the program).

Memory management

Among other things, a multi-programming operating system kernel must be responsible for managing all system memory which is currently in use by programs. This ensures that a program does not interfere with memory already in use by another program. Since programs time share, each program must have independent access to memory.

Cooperative memory management, used by many early operating systems, assumes that all programs make voluntary use of the kernel's memory manager, and do not exceed their allocated memory. This system of memory management is almost never seen any more, since programs often contain bugs which can cause them to exceed their allocated memory. If a program fails, it may cause memory used by one or more other programs to be affected or overwritten. Malicious programs or viruses may purposefully alter another program's memory, or may affect the operation of the operating system itself. With cooperative memory management, it takes only one misbehaved program to crash the system.

Memory protection enables the kernel to limit a process' access to the computer's memory. Various methods of memory protection exist, including memory segmentation and paging. All methods require some level of hardware support (such as the 80286 MMU), which doesn't exist in all computers.

In both segmentation and paging, certain protected mode registers specify to the CPU what memory address it should allow a running program to access. Attempts to access other addresses trigger an interrupt which cause the CPU to re-enter supervisor mode, placing the kernel in charge. This is called asegmentation violation or Seg-V for short, and since it is both difficult to assign a meaningful result to such an operation, and because it is usually a sign of a misbehaving program, the kernel generally resorts to terminating the offending program, and reports the error.

Windows versions 3.1 through ME had some level of memory protection, but programs could easily circumvent the need to use it. A general protection fault would be produced, indicating a segmentation violation had occurred; however, the system would often crash anyway.

Virtual memory

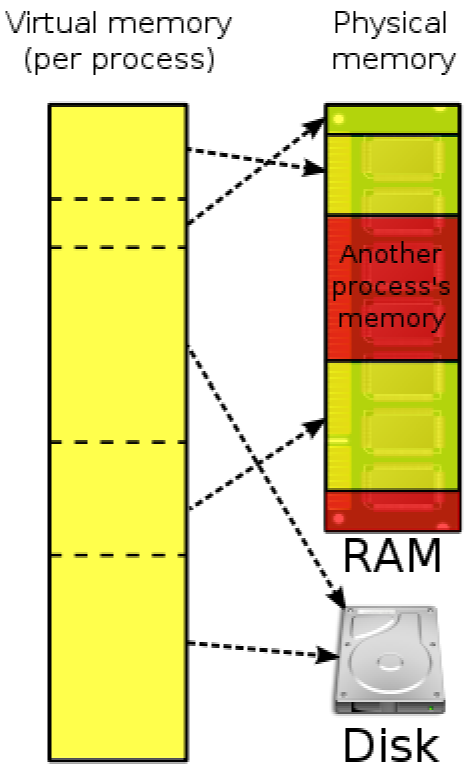
The use of virtual memory addressing (such as paging or segmentation) means that the kernel can choose what memory each program may use at any given time, allowing the operating system to use the same memory locations for multiple tasks.

If a program tries to access memory that isn't in its current range of accessible memory, but nonetheless has been allocated to it, the kernel is interrupted in the same way as it would if the program were to exceed its allocated memory. (See section on memory management.) Under UNIX this kind of interrupt is referred to as a page fault.

When the kernel detects a page fault it generally adjusts the virtual memory range of the program which triggered it, granting it access to the memory requested. This gives the kernel discretionary power over where a particular application's memory is stored, or even whether or not it has actually been allocated yet.

In modern operating systems, memory which is accessed less frequently can be temporarily stored on disk or other media to make that space available for use by other programs. This is called swapping, as an area of memory can be used by multiple programs, and what that memory area contains can be swapped or exchanged on demand.

"Virtual memory" provides the programmer or the user with the perception that there is a much larger amount of RAM in the computer than is really there.



Many operating systems can "trick" programs into using memory scattered around the hard disk and RAM as if it is one continuous chunk of memory, called virtual memory.

Multitasking

Multitasking refers to the running of multiple independent computer programs on the same computer; giving the appearance that it is performing the tasks at the same time. Since most computers can do at most one or two things at one time, this is generally done via time-sharing, which means that each program uses a share of the computer's time to execute.

An operating system kernel contains a scheduling program which determines how much time each process spends executing, and in which order execution control should be passed to programs. Control is passed to a process by the kernel, which allows the program access to the CPU and memory. Later, control is returned to the kernel through some mechanism, so that another program may be allowed to use the CPU. This so-called passing of control between the kernel and applications is called a context switch.

An early model which governed the allocation of time to programs was called cooperative multitasking. In this model, when control is passed to a program by the kernel, it may execute for as long as it wants before explicitly returning control to the kernel. This means that a malicious or malfunctioning program may not only prevent any other programs from using the CPU, but it can hang the entire system if it enters an infinite loop.

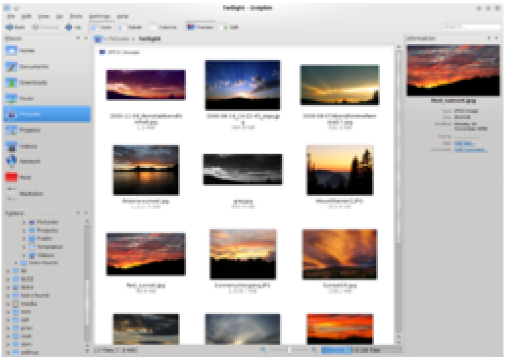
Modern operating systems extend the concepts of application preemption to device drivers and kernel code, so that the operating system has preemptive control over internal run-times as well.

The philosophy governing preemptive multitasking is that of ensuring that all programs are given regular time on the CPU. This implies that all programs must be limited in how much time they are allowed to spend on the CPU without being interrupted. To accomplish this, modern operating system kernels make use of a timed interrupt. A protected mode timer is set by the kernel which triggers a return to supervisor mode after the specified time has elapsed. (See above sections on Interrupts and Dual Mode Operation.)

On many single user operating systems cooperative multitasking is perfectly adequate, as home computers generally run a small number of well tested programs. The AmigaOS is an exception, having pre-emptive multitasking from its very first version. Windows NT was the first version of Microsoft Windows which enforced preemptive multitasking, but it didn't reach the home user market until Windows XP (since Windows NT was targeted at professionals).

Disk access and file systems

Access to data stored on disks is a central feature of all operating systems. Computers store data on disks using files, which are structured in specific ways in order to allow for faster access, higher reliability, and to make better use out of the drive's available space. The specific way in which files are stored on a disk is called a file system, and enables files to have names and attributes. It also allows them to be stored in a hierarchy of directories or folders arranged in a directory tree.



Filesystems allow users and programs to organize and sort files on a computer, often through the use of directories (or "folders")

Early operating systems generally supported a single type of disk drive and only one kind of file system. Early file systems were limited in their capacity, speed, and in the kinds of file names and directory structures they could use. These limitations often reflected limitations in the operating systems they were designed for, making it very difficult for an operating system to support more than one file system.

While many simpler operating systems support a limited range of options for accessing storage systems, operating systems like UNIX and Linux support a technology known as a virtual file system or VFS. An operating system such as UNIX supports a wide array of storage devices, regardless of their design or file systems, allowing them to be accessed through a common application programming interface (API). This makes it unnecessary for programs to have any knowledge about the device they are accessing. A VFS allows the operating system to provide programs with access to an unlimited number of devices with an infinite variety of file systems installed on them, through the use of specific device drivers and file system drivers.

A connected storage device, such as a hard drive, is accessed through a device driver. The device driver understands the specific language of the drive and is able to translate that language into a standard language used by the operating system to access all disk drives. On UNIX, this is the language of block devices.

When the kernel has an appropriate device driver in place, it can then access the contents of the disk drive in raw format, which may contain one or more file systems. A file system driver is used to translate the commands used to access each specific file system into a standard set of commands that the operating system can use to talk to all file systems. Programs can then deal with these file systems on the basis of filenames, and directories/folders, contained within a hierarchical structure. They can create, delete, open, and close files, as well as gather various information about them, including access permissions, size, free space, and creation and modification dates.

Various differences between file systems make supporting all file systems difficult. Allowed characters in file names, case sensitivity, and the presence of various kinds of file attributes makes the implementation of a single interface for every file system a daunting task. Operating systems tend to recommend using (and so support natively) file systems specifically designed for them; for example, NTFS in Windows and ext3 and ReiserFS in Linux. However, in practice, third party drives are usually available to give support for the most widely used file systems in most general-purpose operating systems (for example, NTFS is available in Linux through NTFS-3g, and ext2/3 and ReiserFS are available in Windows through third-party software).

Support for file systems is highly varied among modern operating systems, although there are several common file systems which almost all operating systems include support and drivers for. Operating systems vary on file system support and on the disk formats they may be installed on. Under Windows, each file system is usually limited in application to certain media; for example, CDs must use ISO 9660 or UDF, and as of Windows Vista, NTFS is the only file system which the operating system can be installed on. It is possible to install Linux onto many types of file systems. Unlike other operating systems, Linux and UNIX allow any file system to be used regardless of the media it is stored in, whether it is a hard drive, a disc (CD, DVD…), a USB flash drive, or even contained within a file located on another file system.

Device drivers

A device driver is a specific type of computer software developed to allow interaction with hardware devices. Typically this constitutes an interface for communicating with the device, through the specific computer bus or communications subsystem that the hardware is connected to, providing commands to and/or receiving data from the device, and on the other end, the requisite interfaces to the operating system and software applications. It is a specialized hardware-dependent computer program which is also operating system specific that enables another program, typically an operating system or applications software package or computer program running under the operating system kernel, to interact transparently with a hardware device, and usually provides the requisite interrupt handling necessary for any necessary asynchronous time-dependent hardware interfacing needs.

The key design goal of device drivers is abstraction. Every model of hardware (even within the same class of device) is different. Newer models also are released by manufacturers that provide more reliable or better performance and these newer models are often controlled differently. Computers and their operating systems cannot be expected to know how to control every device, both now and in the future. To solve this problem, operating systems essentially dictate how every type of device should be controlled. The function of the device driver is then to translate these operating system mandated function calls into device specific calls. In theory a new device, which is controlled in a new manner, should function correctly if a suitable driver is available. This new driver ensures that the device appears to operate as usual from the operating system's point of view.

Under versions of Windows before Vista and versions of Linux before 2.6, all driver execution was co-operative, meaning that if a driver entered an infinite loop it would freeze the system. More recent revisions of these operating systems incorporate kernel preemption, where the kernel interrupts the driver to give it tasks, and then separates itself from the process until it receives a response from the device driver, or gives it more tasks to do.

Networking

Currently most operating systems support a variety of networking protocols, hardware, and applications for using them. This means that computers running dissimilar operating systems can participate in a common network for sharing resources such as computing, files, printers, and scanners using either wired or wireless connections. Networks can essentially allow a computer's operating system to access the resources of a remote computer to support the same functions as it could if those resources were connected directly to the local computer. This includes everything from simple communication, to using networked file systems or even sharing another computer's graphics or sound hardware. Some network services allow the resources of a computer to be accessed transparently, such as SSH which allows networked users direct access to a computer's command line interface.

Client/server networking allows a program on a computer, called a client, to connect via a network to another computer, called a server. Servers offer (or host) various services to other network computers and users. These services are usually provided through ports or numbered access points beyond the server's network address. Each port number is usually associated with a maximum of one running program, which is responsible for handling requests to that port. A daemon, being a user program, can in turn access the local hardware resources of that computer by passing requests to the operating system kernel.

Many operating systems support one or more vendor-specific or open networking protocols as well, for example, SNA on IBM systems, DECnet on systems from Digital Equipment Corporation, and Microsoft-specific protocols (SMB) on Windows. Specific protocols for specific tasks may also be supported such asNFS for file access. Protocols like ESound, or esd can be easily extended over the network to provide sound from local applications, on a remote system's sound hardware.

Security

A computer being secure depends on a number of technologies working properly. A modern operating system provides access to a number of resources, which are available to software running on the system, and to external devices like networks via the kernel.

The operating system must be capable of distinguishing between requests which should be allowed to be processed, and others which should not be processed. While some systems may simply distinguish between “privileged” and “non-privileged”, systems commonly have a form of requester *identity*, such as a user name. To establish identity there may be a process of *authentication*. Often a username must be quoted, and each username may have a password. Other methods of authentication, such as magnetic cards or biometric data, might be used instead. In some cases, especially connections from the network, resources may be accessed with no authentication at all (such as reading files over a network share). Also covered by the concept of requester identity is *authorization*; the particular services and resources accessible by the requester once logged into a system are tied to either the requester’s user account or to the variously configured groups of users to which the requester belongs.

In addition to the allow or disallow model of security, a system with a high level of security also offers auditing options. These would allow tracking of requests for access to resources (such as, “who has been reading this file?”). Internal security, or security from an already running program is only possible if all possibly harmful requests must be carried out through interrupts to the operating system kernel. If programs can directly access hardware and resources, they cannot be secured.

External security involves a request from outside the computer, such as a login at a connected console or some kind of network connection. External requests are often passed through device drivers to the operating system’s kernel, where they can be passed onto applications, or carried out directly. Security of operating systems has long been a concern because of highly sensitive data held on computers, both of a commercial and military nature. The United StatesGovernment Department of Defense (DoD) created the *Trusted Computer System Evaluation Criteria* (TCSEC) which is a standard that sets basic requirements for assessing the effectiveness of security. This became of vital importance to operating system makers, because the TCSEC was used to evaluate, classify and select trusted operating systems being considered for the processing, storage and retrieval of sensitive or classified information.

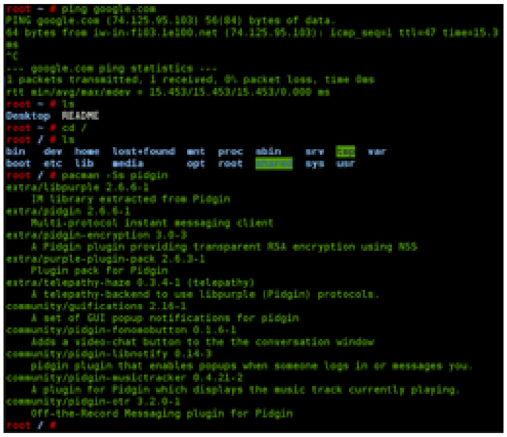
Network services include offerings such as file sharing, print services, email, web sites, and file transfer protocols (FTP), most of which can have compromised security. At the front line of security are hardware devices known as firewalls or intrusion detection/prevention systems. At the operating system level, there are a number of software firewalls available, as well as intrusion detection/prevention systems. Most modern operating systems include a software firewall, which is enabled by default. A software firewall can be configured to allow or deny network traffic to or from a service or application running on the operating system. Therefore, one can install and be running an insecure service, such as Telnet or FTP, and not have to be threatened by a security breach because the firewall would deny all traffic trying to connect to the service on that port.

An alternative strategy, and the only sandbox strategy available in systems that do not meet the Popek and Goldberg virtualization requirements, is where the operating system is not running user programs as native code, but instead either emulates a processor or provides a host for a p-code based system such as Java.

Internal security is especially relevant for multi-user systems; it allows each user of the system to have private files that the other users cannot tamper with or read. Internal security is also vital if auditing is to be of any use, since a program can potentially bypass the operating system, inclusive of bypassing auditing.

User interface

Every computer that is to be operated by an individual requires a user interface. The user interface is usually referred to as a shell and is essential if human interaction is to be supported. The user interface views the directory structure and requests services from the operating system that will acquire data from input hardware devices, such as a keyboard, mouse or credit card reader, and requests operating system services to display prompts, status messages and such on output hardware devices, such as a video monitor or printer. The two most common forms of a user interface have historically been the command-line interface, where computer commands are typed out line-by-line, and the graphical user interface, where a visual environment (most commonly a WIMP) is present.

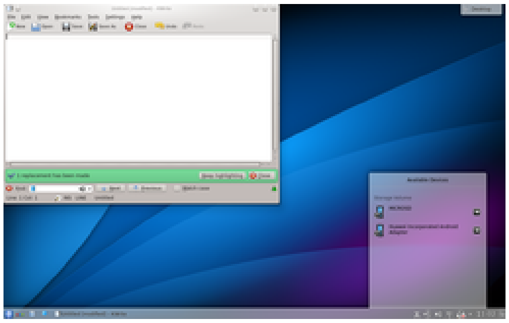


```
root ~ # ping google.com
PING google.com (74.125.95.103) 56(84) bytes of data.
64 bytes from iw-in-f103.1e100.net (74.125.95.103): icmp_seq=1 ttl=47 time=15.3
ms
^C
--- google.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 15.453/15.453/15.453/0.000 ms
root ~ # ls
Desktop  README
root ~ # cd /
root / # ls
bin   dev  home  lost+found  mnt  proc  sbin    srv  tmp  var
boot  etc  lib   media       opt  root  shared  sys  usr
root / # pacman -Ss pidgin
extra/libpurple 2.6.6-1
    IM library extracted from Pidgin
extra/pidgin 2.6.6-1
    Multi-protocol instant messaging client
extra/pidgin-encryption 3.0-3
    A Pidgin plugin providing transparent RSA encryption using NSS
extra/purple-plugin-pack 2.6.3-1
    Plugin pack for Pidgin
extra/telepathy-haze 0.3.4-1 (telepathy)
    A telepathy-backend to use libpurple (Pidgin) protocols.
community/guifications 2.16-1
    A set of GUI popup notifications for pidgin
community/pidgin-fonomobutton 0.1.6-1
    Adds a video-chat button to the the conversation window
community/pidgin-libnotify 0.14-3
    pidgin plugin that enables popups when someone logs in or messages you.
community/pidgin-musictracker 0.4.21-2
    A plugin for Pidgin which displays the music track currently playing.
community/pidgin-otr 3.2.0-1
    Off-the-Record Messaging plugin for Pidgin
root / #
```

A screenshot of the Bourne Again Shell command line. Each command is typed out after the 'prompt', and then its output appears below, working its way down the screen. The current command prompt is at the bottom.

Graphical user interfaces

A screenshot of the KDE Plasma Desktop graphical user interface. Programs take the form of images on the screen, and the files, folders (directories), and applications take the form of icons and symbols. A mouse is used to navigate the computer.



Most of the modern computer systems support graphical user interfaces (GUI), and often include them. In some computer systems, such as the original implementation of Mac OS, the GUI is integrated into the kernel.

While technically a graphical user interface is not an operating system service, incorporating support for one into the operating system kernel can allow the GUI to be more responsive by reducing the number of context switches required for the GUI to perform its output functions. Other operating systems are modular, separating the graphics subsystem from the kernel and the Operating System. In the 1980s UNIX, VMS and many others had operating systems that were built this way. Linux and Mac OS X are also built this way. Modern releases of Microsoft Windows such as Windows Vista implement a graphics subsystem that is mostly in user-space; however the graphics drawing routines of versions betweenWindows NT 4.0 and Windows Server 2003 exist mostly in kernel space. Windows 9x had very little distinction between the interface and the kernel.

Many computer operating systems allow the user to install or create any user interface they desire. The X Window System in conjunction with GNOME or KDE Plasma Desktop is a commonly found setup on most Unix and Unix-like (BSD, Linux, Solaris) systems. A number of Windows shell replacements have been released for Microsoft Windows, which offer alternatives to the included Windows shell, but the shell itself cannot be separated from Windows.

Numerous Unix-based GUIs have existed over time, most derived from X11. Competition among the various vendors of Unix (HP, IBM, Sun) led to much fragmentation, though an effort to standardize in the 1990s to COSE and CDE failed for various reasons, and were eventually eclipsed by the widespread adoption of GNOME and K Desktop Environment. Prior to free software-based toolkits and desktop environments, Motif was the prevalent toolkit/desktop combination (and was the basis upon which CDE was developed).

Graphical user interfaces evolve over time. For example, Windows has modified its user interface almost every time a new major version of Windows is released, and the Mac OS GUI changed dramatically with the introduction of Mac OS X in 1999.

Real-time operating systems

A real-time operating system (RTOS) is an operating system intended for applications with fixed deadlines (real-time computing). Such applications include some small embedded systems, automobile engine controllers, industrial robots, spacecraft, industrial control, and some large-scale computing systems.

An early example of a large-scale real-time operating system was Transaction Processing Facility developed by American Airlines and IBM for the Sabre Airline Reservations System.

Embedded systems that have fixed deadlines use a real-time operating system such as VxWorks, PikeOS, eCos, QNX, MontaVista Linux and RTLinux.Windows CE is a real-time operating system that shares similar APIs to desktop Windows but shares none of desktop Windows' codebase. Symbian OS also has an RTOS kernel (EKA2) starting with version 8.0b.

Some embedded systems use operating systems such as Palm OS, BSD, and Linux, although such operating systems do not support real-time computing.

Operating system development as a hobby

Operating system development is one of the most complicated activities in which a computing hobbyist may engage. A hobby operating system may be classified as one whose code has not been directly derived from an existing operating system, and has few users and active developers.

In some cases, hobby development is in support of a "home brew" computing device, for example, a simple single-board computer powered by a 6502 microprocessor. Or, development may be for an architecture already in widespread use. Operating system development may come from entirely new concepts, or may commence by modeling an existing operating system. In either case, the hobbyist is his/her own developer, or may interact with a small and sometimes unstructured group of individuals who have like interests.

Examples of a hobby operating system include ReactOS and Syllable.

Diversity of operating systems and portability

Application software is generally written for use on a specific operating system, and sometimes even for specific hardware. When porting the application to run on another OS, the functionality required by that application may be implemented differently by that OS (the names of functions, meaning of arguments, etc.) requiring the application to be adapted, changed, or otherwise maintained.

Unix was the first operating system not written in assembly language, making it very portable to systems different from its native PDP-11.

This cost in supporting operating systems diversity can be avoided by instead writing applications against software platforms like Java or Qt. These abstractions have already borne the cost of adaptation to specific operating systems and their system libraries.

Another approach is for operating system vendors to adopt standards. For example, POSIX and OS abstraction layers provide commonalities that reduce porting costs.

Market share

2013 Worldwide Device Shipments by Operating System

Operating System	2012 (Million of Units)	2013 (Million of Units)
Android	504	878

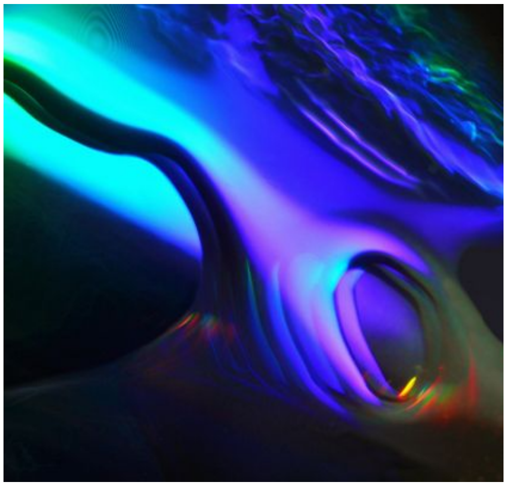
Windows	346	328
iOS/Mac OS	214	267
BlackBerry	35	24
Others	1,117	803
Total	2,216	2,300

Licensing & Attributions

CC licensed content, Shared previously

- Operating System. **Provided by:** Wikipedia. **Located at:** https://en.wikipedia.org/wiki/Operating_system. **License:** *CC BY-SA: Attribution-ShareAlike*
- Operating Systems. **Authored by:** Brennen Bearnes. **Located at:** https://www.flickr.com/photos/p1k3/4544340039/. **License:** *CC BY-SA: Attribution-ShareAlike*

READING: UTILITY SOFTWARE



Utility software is system software designed to help analyze, configure, optimize or maintain a computer.

Utility software usually focuses on *how* the computer infrastructure (including the computer hardware, operating system, software and data storage) operates. Utility software, along with operating system software, is a type of system software, distinguishing it from application software.

Utility software

- Anti-virus utilities scan for computer viruses.
- Archivers output a stream or a single file when provided with a directory or a set of files. Archive utilities, unlike archive suites, usually do not include compression or encryption capabilities. Some archive utilities may even have a separate un-archive utility for the reverse operation.
- Backup software can make copies of all information stored on a disk and restore either the entire disk (e.g. in an event of disk failure) or selected files (e.g. in an event of accidental deletion).
- Clipboard managers expand the clipboard functionality of an operating system .
- Cryptographic utilities encrypt and decrypt streams and files.
- Data compression utilities output a shorter stream or a smaller file when provided with a stream or file.
- Data synchronization utilities establish consistency among data from a source to a target data storage and vice versa. There are several branches of this type of utility:
    - File synchronization utilities maintain consistency between two sources. They may be used to create redundancy or backup copies but are also used to help users carry their digital music, photos and video in their mobile devices.
    - Revision control utilities are intended to deal with situations where more than one user attempts to simultaneously modify the same file.
- Debuggers are used to test and “debug” other programs, mainly to solve programming errors. Also utilized for reverse engineering of software or systems.
- Disk checkers can scan operating hard drive.
- Disk cleaners can find files that are unnecessary to computer operation, or take up considerable amounts of space. Disk cleaner helps the user to decide what to delete when their hard disk is full.

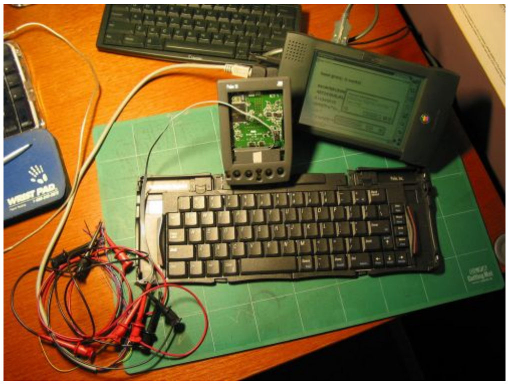
- Disk compression utilities can transparently compress/uncompress the contents of a disk, increasing the capacity of the disk.
- Disk defragmenters can detect computer files whose contents are scattered across several locations on the hard disk, and move the fragments to one location to increase efficiency.
- Disk partitions can divide an individual drive into multiple logical drives, each with its own file system which can be mounted by the operating system and treated as an individual drive.
- Disk space analyzers for the visualization of disk space usage by getting the size for each folder (including sub folders) & files in folder or drive. showing the distribution of the used space.
- Disk storage utilities
- File managers provide a convenient method of performing routine data management tasks, such as deleting, renaming, cataloging, uncataloging, moving, copying, merging, generating and modifying data sets.
- Hex editors directly modify the text or data of a file. These files could be data or an actual program.
- Memory testers check for memory failures.
- Network utilities analyze the computer's network connectivity, configure network settings, check data transfer or log events.
- Package managers are used to configure, install or keep up to date other software on a computer.
- Registry cleaners clean and optimize the Windows Registry by removing old registry keys that are no longer in use.
- Screensavers were desired to prevent phosphor burn-in on CRT and plasma computer monitors by blanking the screen or filling it with moving images or patterns when the computer is not in use. Contemporary screensavers are used primarily for entertainment or security.
- System monitors for monitoring resources and performance in a computer system.
- System profilers provide detailed information about the software installed and hardware attached to the computer.

Licensing & Attributions

CC licensed content, Shared previously

- Utility Software. **Provided by:** Wikipedia. **Located at:** https://en.wikipedia.org/wiki/Utility_software. **License:** *CC BY-SA: Attribution-ShareAlike*
- Blue Wrench Emulsion. **Authored by:** Bill Gracey. **Located at:** https://www.flickr.com/photos/9422878@N08/5396568649/. **License:** *CC BY-NC-ND: Attribution-NonCommercial-NoDerivatives*
- Screensaver. **Authored by:** Russ Morris. **Located at:** https://www.flickr.com/photos/russmorris/939663869/. **License:** *CC BY-NC-ND: Attribution-NonCommercial-NoDerivatives*

READING: DEVICE DRIVER



In computing, a **device driver** (commonly referred to as a *driver*) is a computer program that operates or controls a particular type of device that is attached to a computer. A driver provides a software interface to hardware devices, enabling operating systems and other computer programs to access hardware functions without needing to know precise details of the hardware being used.

A driver typically communicates with the device through the computer bus or communications subsystem to which the hardware connects. When a calling program invokes a routine in the driver, the driver issues commands to the device. Once the device sends data back to the driver, the driver may invoke routines in the original calling program. Drivers are hardware-dependent and operating-system-specific. They usually provide the interrupt handling required for any necessary asynchronous time-dependent hardware interface.

Purpose

Device drivers simplify programming by acting as translator between a hardware device and the applications or operating systems that use it. Programmers can write the higher-level application code independently of whatever specific hardware the end-user is using.

For example, a high-level application for interacting with a serial port may simply have two functions for “send data” and “receive data.” At a lower level, a device driver implementing these functions would communicate to the particular serial port controller installed on a user’s computer. The commands needed to control a 16550 UART are much different from the commands needed to control an FTDI serial port converter, but each hardware-specific device driver abstracts these details into the same (or similar) software interface.

Development

Writing a device driver requires an in-depth understanding of how the hardware and the software works for a given platform function. Because drivers require low-level access to hardware functions in order to operate, drivers typically operate in a highly privileged environment and can cause system operational issues if something goes wrong. In contrast, most user-level software on modern operating systems can be stopped without greatly affecting the rest of the system. Even drivers executing in user mode can crash a system if the device is erroneously programmed. These factors make it more difficult and dangerous to diagnose problems.

The task of writing drivers thus usually falls to software engineers or computer engineers who work for hardware-development companies. This is because they have better information than most outsiders about the design of their hardware. Moreover, it was traditionally considered in the hardware manufacturer’s interest to guarantee that their clients can use their hardware in an optimum way. Typically, the *logical device driver* (LDD) is written by the operating system vendor, while the *physical device driver* (PDD) is implemented by the device vendor. But in recent years non-vendors have written numerous device drivers, mainly for use with free and open source operating systems. In such cases, it is important that the hardware manufacturer provides information on how the device communicates. Although this information can instead be learned by reverse engineering, this is much more difficult with hardware than it is with software.

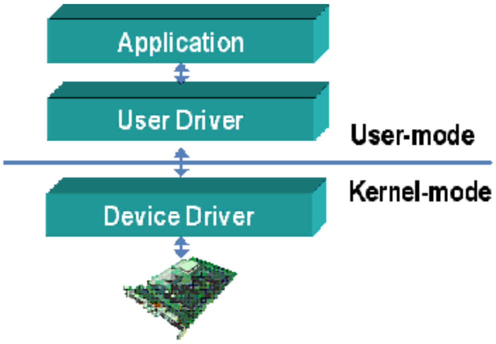
Microsoft has attempted to reduce system instability due to poorly written device drivers by creating a new framework for driver development, called Windows Driver Foundation (WDF). This includes User-Mode Driver Framework (UMDF) that encourages development of certain types of drivers—primarily those that implement a message-based protocol for communicating with their devices—as user-mode drivers. If such drivers malfunction, they do not cause system instability. The Kernel-Mode Driver Framework (KMDF) model continues to allow development of kernel-mode device drivers, but attempts to provide standard implementations of functions that are known to cause problems, including cancellation of I/O operations, power management, and plug and play device support.

Apple has an open-source framework for developing drivers on Mac OS X called the I/O Kit.

In Linux environments, programmers can build device drivers as parts of the kernel, separately as loadable modules, or as user-mode drivers (for certain types of devices where kernel interfaces exist, such as for USB devices). Makedev includes a list of the devices in Linux: ttyS (terminal), lp (parallel port), hd (disk), loop, sound (these include mixer, sequencer, dsp, and audio)…

The Microsoft Windows .sys files and Linux .ko modules contain loadable device drivers. The advantage of loadable device drivers is that they can be loaded only when necessary and then unloaded, thus saving kernel memory.

Kernel mode vs. user mode



Device drivers, particularly on modern Microsoft Windows platforms, can run in kernel-mode (Ring 0 on x86 CPUs) or in user-mode (Ring 3 on x86 CPUs). The primary benefit of running a driver in user mode is improved stability, since a poorly written user mode device driver cannot crash the system by overwriting kernel memory. On the other hand, user/kernel-mode transitions usually impose a considerable performance overhead, thereby prohibiting user-mode drivers for low latency and high throughput requirements.

Kernel space can be accessed by user module only through the use of system calls. End user programs like the UNIX shell or other GUI-based applications are part of the user space. These applications interact with hardware through kernel supported functions.

Applications

Because of the diversity of modern hardware and operating systems, drivers operate in many different environments. Drivers may interface with:

- printers
- video adapters
- Network cards
- Sound cards
- Local buses of various sorts—in particular, for bus mastering on modern systems
- Low-bandwidth I/O buses of various sorts (for pointing devices such as mice, keyboards, USB, etc.)
- Computer storage devices such as hard disk, CD-ROM, and floppy disk buses (ATA, SATA, SCSI)
- Implementing support for different file systems
- Image scanners
- Digital cameras

Common levels of abstraction for device drivers include:

- For hardware:
  - Interfacing directly
  - Writing to or reading from a device control register
  - Using some higher-level interface (e.g. Video BIOS)
  - Using another lower-level device driver (e.g. file system drivers using disk drivers)
  - Simulating work with hardware, while doing something entirely different[*citation needed*]
- For software:
  - Allowing the operating system direct access to hardware resources
  - Implementing only primitives
  - Implementing an interface for non-driver software (e.g., TWAIN)
  - Implementing a language, sometimes quite high-level (e.g., PostScript)

So choosing and installing the correct device drivers for given hardware is often a key component of computer system configuration.

Virtual device drivers

Virtual device drivers represent a particular variant of device drivers. They are used to emulate a hardware device, particularly in virtualization environments, for example when a DOS program is run on a Microsoft Windows computer or when a guest operating system is run on, for example, a Xen host. Instead of enabling the guest operating system to dialog with hardware, virtual device drivers take the opposite role and emulate a piece of hardware, so that the guest operating system and its drivers running inside a virtual machine can have the illusion of accessing real hardware. Attempts by the guest operating system to access the hardware are routed to the virtual device driver in the host operating system as e.g., function calls. The virtual device driver can also send simulated processor-level events like interrupts into the virtual machine.

Virtual devices may also operate in a non-virtualized environment. For example a virtual network adapter is used with a virtual private network, while a virtualdisk device is used with iSCSI. A good example for virtual device drivers can be Daemon Tools.

There are several variants of virtual device drivers, such as VxDs, VLMs, VDDs.

Open drivers

- Printers: CUPS
- RAIDs: CCISS (Compaq Command Interface for SCSI-3 Support)
- Scanners: SANE
- Video: Vidix, Direct Rendering Infrastructure

Solaris descriptions of commonly used device drivers

- fas: Fast/wide SCSI controller
- hme: Fast (10/100 Mbit/s) Ethernet
- isp: Differential SCSI controllers and the SunSwift card
- glm: (Gigabaud Link Module) UltraSCSI controllers
- scsi: Small Computer Serial Interface (SCSI) devices
- sf: soc+ or social Fiber Channel Arbitrated Loop (FCAL)
- soc: SPARC Storage Array (SSA) controllers and the control device
- social: Serial optical controllers for FCAL (soc+)

APIs

- Windows Display Driver Model (WDDM)—the graphic display driver architecture for Windows Vista, Windows 7 and Windows 8.
- Windows Driver Foundation (WDF)
- Windows Driver Model (WDM)
- Network Driver Interface Specification (NDIS)—a standard network card driver API
- Advanced Linux Sound Architecture (ALSA)—as of 2009 the standard Linux sound-driver interface
- Scanner Access Now Easy (SANE)—a public-domain interface to raster-image scanner-hardware
- I/O Kit—an open-source framework from Apple for developing Mac OS X device drivers
- Installable File System (IFS)—a filesystem API for IBM OS/2 and Microsoft Windows NT
- Open Data-Link Interface (ODI)—a network card API similar to NDIS
- Uniform Driver Interface (UDI)—a cross-platform driver interface project
- Dynax Driver Framework (dxd)—C++ open source cross-platform driver framework for KMDF and IOKit

Identifiers

A device on the PCI bus or USB is identified by two IDs which consist of 4 hexadecimal numbers each. The vendor ID identifies the vendor of the device. The device ID identifies a specific device from that manufacturer/vendor.

A PCI device has often an ID pair for the main chip of the device, and also a subsystem ID pair which identifies the vendor, which may be different from the chip manufacturer.

Licensing & Attributions

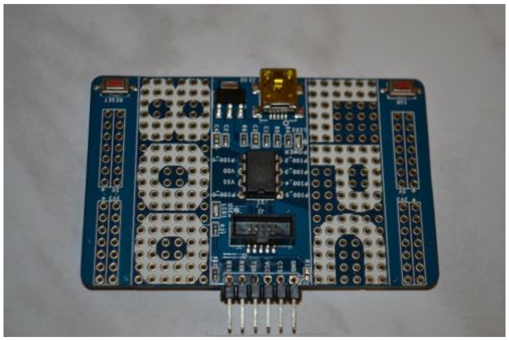
CC licensed content, Shared previously

- Device Driver. **Provided by:** Wikipedia. **Located at:** https://en.wikipedia.org/wiki/Device_driver. **License:** *CC BY-SA: Attribution-ShareAlike*
- Stowaway Keyboard Equipped Newton. **Authored by:** Grant Hutchinson. **Located at:** https://www.flickr.com/photos/splorp/5216150/. **License:** *CC BY-NC-SA: Attribution-NonCommercial-ShareAlike*

Public domain content

- Device driver architecture. **Authored by:** Tutorial. **Located at:** https://commons.wikimedia.org/wiki/File:Driverarch.png. **License:** *Public Domain: No Known Copyright*

READING: FIRMWARE



In electronic systems and computing, **firmware** is a tangible electronic component with embedded software instructions, such as a BIOS. Typically, those software instructions are used to tell an electronic device how to operate. As of 2013, most firmware can be updated. Typical examples of devices containing firmware are embedded systems (such as traffic lights, consumer appliances, and digital watches), computers, computer peripherals, mobile phones, and digital cameras. The firmware contained in these devices provides the control program for the device.

Firmware is held in non-volatile memory devices such as ROM, EPROM, or flash memory. Changing the firmware of a device may rarely or never be done during its economic lifetime; some firmware memory devices are permanently installed and cannot be changed after manufacture. Common reasons for updating firmware include fixing bugs or adding features to the device. This may require ROM integrated circuits to be physically replaced, or flash memory to be reprogrammed through a special procedure. Firmware such as the ROM BIOS of a personal computer may contain only elementary basic functions of a device and may only provide services to higher-level software. Firmware such as the program of an embedded system may be the only program that will run on the system and provide all of its functions.

Before integrated circuits, other firmware devices included a discrete semiconductor diode matrix. The Apollo guidance computer had firmware consisting of a specially manufactured core memory plane, called “core rope memory,” where data were stored by physically threading wires through (1) or around (0) the core storing each data bit.

Origin of the term

Ascher Opler coined the term “firmware” in a 1967 *Datamation* article. Originally, it meant the contents of a writable control store (a small specialized high speed memory), containing microcode that defined and implemented the computer’s instruction set, and that could be reloaded to specialize or modify the instructions that the central processing unit (CPU) could execute. As originally used, firmware contrasted with hardware (the CPU itself) and software (normal instructions executing on a CPU). It was not composed of CPU machine instructions, but of lower-level microcode involved in the implementation of machine instructions. It existed on the boundary between hardware and software; thus the name “firmware”.

Still later, popular usage extended the word “firmware” to denote anything ROM-resident, including processor machine-instructions for BIOS, bootstrap loaders, or specialized applications.

Until the mid-1990s, updating firmware typically involved replacing a storage medium containing firmware, usually a socketed ROM integrated circuit. Flash memory allows firmware to be updated without physically removing an integrated circuit from the system. An error during the update process may make the device non-functional, or “bricked.”

Personal computers

ROM BIOS firmware on a Baby AT motherboard

In some respects, the various firmware components are as important as the operating system in a working computer. However, unlike most modern operating systems, firmware rarely has a well-evolved automatic mechanism of updating itself to fix any functionality issues detected after shipping the unit.

The BIOS may be “manually” updated by a user, using a small utility program. In contrast, firmware in storage devices (hard disks, DVD drives, flash storage) rarely gets updated, even when flash (rather than ROM) storage is used for the firmware; there are no standardized mechanisms for detecting or updating firmware versions.

Most computer peripherals are themselves special-purpose computers. Devices such as printers, scanners, cameras and USB flash drives have internally stored firmware; some devices may also permit field upgrading of their firmware.

Some low-cost peripherals no longer contain non-volatile memory for firmware, and instead rely on the host system to transfer the device control program from a disk file or CD.

Consumer products

As of 2010 most portable music players support firmware upgrades. Some companies use firmware updates to add new playable file formats (codecs); iriver added Vorbis playback support this way, for instance. Other features that may change with firmware updates include the GUI or even the battery life. Most mobile phones have a Firmware Over The Air firmware upgrade capability for much the same reasons; some may even be upgraded to enhance reception or sound quality, illustrating the fact that firmware is used at more than one level in complex products (in a CPU-like microcontroller versus in a digital signal processor, in this particular case).

Automobiles

Since 1996 most automobiles have employed an on-board computer and various sensors to detect mechanical problems. As of 2010 modern vehicles also employ computer-controlled ABS systems and computer-operated Transmission Control Units (TCU). The driver can also get in-dash information while driving in this manner, such as real-time fuel-economy and tire-pressure readings. Local dealers can update most vehicle firmware.

Examples

Examples of firmware include:

- In consumer products:
  - Timing and control systems for washing machines
  - Controlling sound and video attributes, as well as the channel list, in modern TVs
  - EPROM chips used in the Eventide H-3000 series of digital music processors
- In computers:
  - The BIOS found in IBM-compatible personal computers
  - The (U)EFI-compliant firmware used on Itanium systems, Intel-based computers from Apple, and many Intel desktop computer motherboards
  - Open Firmware, used in SPARC-based computers from Sun Microsystems and Oracle Corporation, PowerPC-based computers from Apple, and computers from Genesi
  - ARCS, used in computers from Silicon Graphics

    - Kickstart, used in the Amiga line of computers (POST, hardware init + Plug and Play auto-configuration of peripherals, kernel, etc.)
    - RTAS (Run-Time Abstraction Services), used in computers from IBM
    - The Common Firmware Environment (CFE)
- In routers and firewalls:
    - LibreWRT—a 100% free software router distribution based on the Linux-libre kernel
    - IPFire—an open-source firewall/router distribution based on the Linux kernel
    - fli4l—an open-source firewall/router distribution based on the Linux kernel
    - OpenWrt—an open-source firewall/router distribution based on the Linux kernel
    - m0n0wall—an embedded firewall distribution of FreeBSD
- In NAS systems:
    - NAS4Free—an open-source NAS operating system based on FreeBSD 9.1
    - Openfiler—a open-source NAS operating system based on the Linux kernel

Flashing

Flashing involves the overwriting of existing firmware or data on EEPROM modules present in an electronic device with new data. This can be done to upgrade a device or to change the provider of a service associated with the function of the device, such as changing from one mobile phone service provider to another or installing a new operating system. If firmware is upgradable, it is often done via a program from the provider, and will often allow the old firmware to be saved before upgrading so it can be reverted to if the process fails, or if the newer version performs worse.

Firmware hacking

Sometimes, third parties create an unofficial new or modified (“aftermarket”) version of firmware to provide new features or to unlock hidden functionality; this is referred to as custom firmware (also “Custom Firmware” in the video game console community). An example is Rockbox as a firmware replacement forportable media players. There are many homebrew projects for video game consoles, which often unlock general-purpose computing functionality in previously limited devices (e.g., running Doom on iPods).

Firmware hacks usually take advantage of the firmware update facility on many devices to install or run themselves. Some, however, must resort to exploits in order to run, because the manufacturer has attempted to lock the hardware to stop it from running unlicensed code.

Most firmware hacks are free software.

HDD firmware hacks

The Moscow-based Kaspersky Lab discovered that a group of developers it refers to as the “Equation Group” has developed hard disk drive firmware modifications for various drive models, containing a trojan horse that allows data to be stored on the drive in locations that will not be erased even if the drive is formatted or wiped. Although the Kaspersky Lab report did not explicitly claim that this group is part of the United States National Security Agency (NSA), evidence obtained from the code of various Equation Group software suggests that they are part of the NSA.

Researchers from the Kaspersky Lab categorized the undertakings by Equation Group as the most advanced hacking operation ever uncovered, also documenting around 500 infections caused by the Equation Group in at least 42 countries.

Security risks

Mark Shuttleworth, founder of the Ubuntu Linux distribution, has described proprietary firmware as a security risk, saying that “firmware on your device is theNSA’s best friend” and calling firmware “a trojan horse of monumental proportions”. He has pointed out that low-quality, nonfree firmware is a major threat to system security: “Your biggest mistake is to assume that the NSA is the only institution abusing this position of trust – in fact, it's reasonable to assume that all firmware is a cesspool of insecurity, courtesy of incompetence of the highest degree from manufacturers, and competence of the highest degree from a very wide range of such agencies". As a solution to this problem, he has called for declarative firmware, which would describe "hardware linkage and dependencies" and "should not include executable code".

Custom firmware hacks have also focused on injecting malware into devices such as smartphones or USB devices. One such smartphone injection was demonstrated on the Symbian OS at MalCon, a hacker convention. A USB device firmware hack called *BadUSB* was presented at Black Hat USA 2014conference, demonstrating how a USB flash drive microcontroller can be reprogrammed to spoof various other device types in order to take control of a computer, exfiltrate data, or spy on the user. Other security researchers have worked further on how to exploit the principles behind BadUSB, releasing at the same time the source code of hacking tools that can be used to modify the behavior of different USB devices.

Licensing & Attributions

CC licensed content, Shared previously

- Firmware. **Provided by:** Wikipedia. **Located at:** https://en.wikipedia.org/wiki/Firmware. **License:** *CC BY-SA: Attribution-ShareAlike*
- firmware circuit. **Located at:** https://pixabay.com/en/lpc-ic-testing-firmware-circuit-301425/. **License:** *CC0: No Rights Reserved*
- Tesla Series S - Computer Panel. **Authored by:** Wesley Fryer. **Located at:** https://www.flickr.com/photos/wfryer/8081558847/. **License:** *CC BY: Attribution*

READING: SYSTEMS SOFTWARE DIAGRAM

View this diagram depicting the relationship of *system software* to a*pplication software* to *computer hardware*. You can think of the computer itself as the hardware, then the two subcategories of systems software, which are operating systems and utility programs, and applications software, which are designed with a specific function for the computer user in mind.

Licensing & Attributions

CC licensed content, Original

- Introduction to Computer Applications and Concepts, Link to Systems Software diagram. **Authored by:** Jim Shannon. **Provided by:** Extended Learning Institute (ELI), Northern Virginia Community College (NOVA). **Located at:** http://eli.nvcc.edu/. **License:** *CC BY: Attribution*

Public domain content

- Image of blue i. **Authored by:** OpenClipArtVectors. **Located at:** https://pixabay.com/en/information-circle-icon-blue-160885/. **License:** *CC0: No Rights Reserved*

MODULE 4: WINDOWS

READING: WINDOWS 8

Read, view, and complete the following tutorials at GCF LearnFree.org.

All about Windows 8

- #2 – WINDOWS 8.1 FEATURES
- #3 – UPGRADING TO WINDOWS 8.1

Supplemental Trainings

You can find the following supplemental trainings by clicking the MS Office Tutorials link in the course menu.

- Atomic Learning—Use these videos to supplement your GCF Learn Free Tutorials as necessary.
- Windows Help Menu—Additional learning materials are also available from the Windows HELP menu. Click on the BLUE QUESTION MARK in the upper right hand corner of your desktop page. This will take you to the Windows Help page where you will find tutorials and a searchable index of topics.

Licensing & Attributions

CC licensed content, Shared previously

- Introduction to Computer Applications and Concepts, Windows Training. **Authored by:** Jim Shannon. **Provided by:** Extended Learning Institute of Northern Virginia Community College. **Located at:** http://www.gcflearnfree.org/windows8. **License:** *CC BY: Attribution*

Public domain content

- Image of blue i. **Authored by:** OpenClipArtVectors. **Located at:** https://pixabay.com/en/information-circle-icon-blue-160885/. **License:** *CC0: No Rights Reserved*

READING: USING WINDOWS 8

Read, view, and complete the following tutorials at GCF LearnFree.org.

Using Windows 8

- #4 – GETTING STARTED WITH WINDOWS 8
- #5 – USING ONEDRIVE WITH WINDOWS 8
- #6 – USING THE SEARCH FEATURE
- #7 – PERSONALIZING YOUR START SCREEN

Supplemental Trainings
You can find the following supplemental trainings by clicking the MS Office Tutorials link in the course menu.

- Atomic Learning—Use these videos to supplement your GCF Learn Free Tutorials as necessary.

- Windows Help Menu—Additional learning materials are also available from the Windows HELP menu. Click on the BLUE QUESTION MARK in the upper right hand corner of your desktop page. This will take you to the Windows Help page where you will find tutorials and a searchable index of topics.

Licensing & Attributions

CC licensed content, Original

- Introduction to Computer Applications and Concepts, Introduction to Windows Training. **Authored by:** Jim Shannon. **Provided by:** Extended Learning Institute (ELI), Northern Virginia Community College (NOVA). **Located at:** http://eli.nvcc.edu/. **License:** *CC BY: Attribution*

Public domain content

- Image of file folder and magnifying glass. **Authored by:** ClkrFreeVectorImages. **Located at:** https://pixabay.com/en/folders-glass-zoom-lens-search-25133/. **License:** *CC0: No Rights Reserved*

READING: WORKING WITH THE DESKTOP

Read, view, and complete the following tutorials at GCF LearnFree.org.

Working with the Desktop

- #8 – GETTING STARTED WITH THE DESKTOP
- #9 – MANAGING YOUR FILES AND FOLDERS
- #10- PERSONALIZING YOUR DESKTOP

Supplemental Trainings
You can find the following supplemental trainings by clicking the MS Office Tutorials link in the course menu.

- Atomic Learning—Use these videos to supplement your GCF Learn Free Tutorials as necessary.
- Windows Help Menu—Additional learning materials are also available from the Windows HELP menu. Click on the BLUE QUESTION MARK in the upper right hand corner of your desktop page. This will take you to the Windows Help page where you will find tutorials and a searchable index of topics.

Licensing & Attributions

CC licensed content, Original

- Introduction to Computer Applications and Concepts. **Authored by:** Jim Shannon. **Provided by:** Extended Learning Institute (ELI), Northern Virginia Community College (NOVA). **Located at:** http://eli.nvcc.edu/. **License:** *CC BY: Attribution*

Public domain content

- Image of file folder and magnifying glass. **Authored by:** ClkrFreeVectorImages. **Located at:** https://pixabay.com/en/folders-glass-zoom-lens-search-25133/. **License:** *CC0: No Rights Reserved*

TUTORIALS: WINDOWS 8 APPS

Read, view, and complete the following tutorials at GCF LearnFree.org.

Windows 8 Apps

- #11- USING THE PEOPLE APP
- #12- USING THE MAIL APP
- #13- INTERNET EXPLORER
- #14- THE MUSIC AND VIDEO APP
- #15- DOWNLOADING APPS FROM THE WINDOWS STORE

Supplemental Trainings
You can find the following supplemental trainings by clicking the MS Office Tutorials link in the course menu.

- Atomic Learning—Use these videos to supplement your GCF Learn Free Tutorials as necessary.

- To access these trainings, follow the instructions from NVCC.

- **Windows Help Menu**—Additional learning materials are also available from the Windows HELP menu. Click on the BLUE QUESTION MARK in the upper right hand corner of your desktop page. This will take you to the Windows Help page where you will find tutorials and a searchable index of topics.

Licensing & Attributions

CC licensed content, Original

- Introduction to Computer Applications and Concepts. **Authored by:** Jim Shannon. **Provided by:** Extended Learning Institute (ELI), Northern Virginia Community College (NOVA). **Located at:** http://eli.nvcc.edu/. **License:** *CC BY: Attribution*

Public domain content

- Image of file folder and magnifying glass. **Authored by:** ClkrFreeVectorImages. **Located at:** https://pixabay.com/en/folders-glass-zoom-lens-search-25133/. **License:** *CC0: No Rights Reserved*

TUTORIALS: CHANGING SETTINGS

Read, view, and complete the following tutorials at GCF LearnFree.org.

Changing your Computer's Settings

- #16- MANAGING USER ACCOUNTS AND PARENTAL CONTROLS
- #17- OPENING YOUR FILES WITH DIFFERENT APPS
- #18 – SECURITY AND MAINTENANCE

Supplemental Trainings
You can find the following supplemental trainings by clicking the MS Office Tutorials link in the course menu.

- Atomic Learning—Use these videos to supplement your GCF Learn Free Tutorials as necessary.
  - To access these trainings, follow the instructions from NVCC.
- **Windows Help Menu**—Additional learning materials are also available from the Windows HELP menu. Click on the BLUE QUESTION MARK in the upper right hand corner of your desktop page. This will take you to the Windows Help page where you will find tutorials and a searchable index of topics.

Licensing & Attributions

CC licensed content, Original

- Introduction to Computer Applications and Concepts. **Authored by:** Jim Shannon. **Provided by:** Extended Learning Institute (ELI), Northern Virginia Community College (NOVA). **Located at:** http://eli.nvcc.edu/. **License:** *CC BY: Attribution*

Public domain content

- Image of file folder and magnifying glass. **Authored by:** ClkrFreeVectorImages. **Located at:** https://pixabay.com/en/folders-glass-zoom-lens-search-25133/. **License:** *CC0: No Rights Reserved*

MODULE 5: COMMUNICATIONS AND INFORMATION LITERACY

READING: THE INTERNET BASICS

Introduction

The Internet is probably the most exciting, the most popular, most visible and definitely the “coolest” information systems development of the decade.

What is the Internet?

The origins of the Internet can be found in the early sixties, when the U.S. Department of Defense sponsored a project to develop a telecommunications network that would survive a nuclear attack. It had to link together a diverse set of computers and work in a decentralized manner so that, if any part of the network were not functioning, network traffic would automatically be re-routed via other network nodes. This project quickly grew into a popular academic network linking virtually all major research institutions and U.S. universities. Soon other countries jumped onto the bandwagon, thus linking academics and researchers across the globe. True to the academic ethos, it quickly became a means for global information sharing. By now, businesses also got a piece of the action. This was spurred on by the trend to network the personal computers in home and business environments and the development of more user-friendly, graphical interfaces: the web-browser and the Windows operating system.

The Internet (or, more colloquially, the Net) consists of a huge and fast-growing number (hundreds of thousands) of interconnected networks linked together. Currently more than 100 million users are connected to the Internet. The popularity of the Internet can be explained by the amount of information it makes available: the equivalent of many libraries of information is stored on millions of computers (Internet hosts), much of it free of charge to all Internet users. This information is provided by educational institutions, governmental agencies and organizations, individuals, and increasingly by businesses. Hence, the Internet is frequently referred to as the Information Highway or the Infobahn.

But the Internet is more than just a huge information resource. Its initial purpose was to act as a communications network and it fulfills that role well. It is the transport mechanism for electronic mail, the transfer of computer files, remote computer access and even allows for voice calls. Businesses quickly realized the potential of the multimedia-enabled Internet for marketing purposes. Of late, more and more business transactions are being conducted via the Internet: electronic commerce (e-commerce) is the latest revolution to be embraced by the Internet community.

Electronic mail

Probably the most popular Internet service is electronic mail, more commonly known as email. This consists of the sending of messages composed on the computer, via a network, directly to the computer of the recipient who reads the message on his/her computer. Knowledge workers with access to e-mail write five to ten times as many e-mail messages as hand-written notes. The following are just some of the advantages of e-mail.

- Reliability: although there is no guarantee, you will normally receive quick feedback if the address does not exist or there is a similar delivery problem.
- Efficiency: many short-cut tools exist to increase your efficiency when composing messages. You can use your computer's cut-and-paste function, you can have managed address books and lists, when replying to another message you can automatically incorporate any part of the message to which you are, etc. And it is just as easy to send a message to one as to a whole list of addressees. (Admittedly, this results in a lot of abuse and information overload on the recipient's side.)
- Digital: e-mail is composed on a computer and remains in computer-readable format all the way to its destination. Thus one can also easily incorporate other computer data such as graphics or document files.
- Cheap: because the capacity of the Internet and disk storage is increasing all the time, the cost of a sending and storing a one-page e-mail message is negligible.
- Speed: messages are generally delivered across the world in a matter of seconds.

The e-mail address

Just like with ordinary postal mail (now usually referred to as snail-mail), you need to know the recipient's address before you can send your message. Internet e-mail addresses have a standard format: username@domain. The username is often the name that your addressee uses to connect to the network, e.g. "jvanbelle" or sometimes a long number. This username is allocated by the LAN administrator. The domain identifies the file server, which acts as the local post office for your recipient's e-mail. The domain consists of several parts, separated by full stops or dots. The international standard for domain identification is ….

- The country code is the international two-letter code for the country (e.g., *au* for Australia, *za* for South Africa, *sa* for Saudi-Arabia, *uk* for Great Britain, etc).
- The two most common types of organizations are co for a commercial organization and ac for an academic institution . Less frequent are org for (not-for-profit) organizations, mil for military, net for networks and gov for government agencies.
- Each country has a national Internet naming body that allows its organizations to chose their own name, as long as no one has claimed the same name before. Examples of South African domain names are anc.org.za, uct.ac.za, fnb.co.za.
- Large organizations often refine the domain further by adding the name of their LAN servers, e.g mail.uct.ac.za.

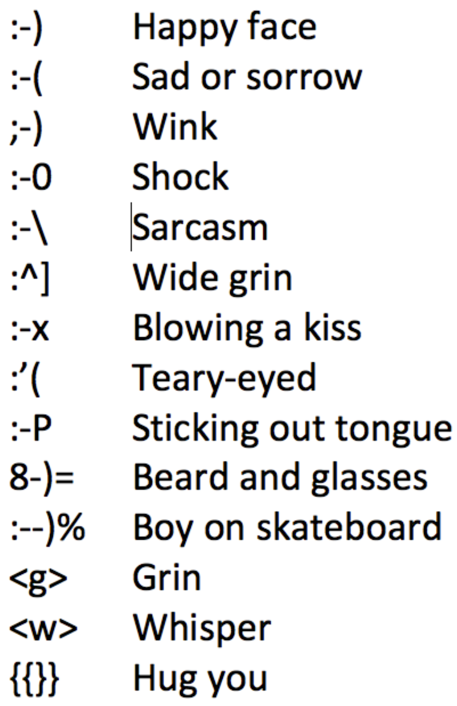
Examples of possible e-mail addresses are: JaneDoe@stats.uct.ac.za (Jane working in the statistics department at the University of Cape Town in South Africa); info@anc.org.za (information department at the ANC, a political party) or SoapJoe@marketing.bt.co.uk (Joe Soap in the marketing department of British Telekom in the U.K.).

The US Americans, having "invented" the Internet, use a slightly different way for their addresses. They leave off their country code (us) and use com for commercial organization or edu for educational institution. Since the majority of Internet users hail from the US, you will encounter many addresses such as JWood@mit.edu or Bill@microsoft.com.

Netiquette

Just as in any other social interaction environment, there are some rules and guidelines for appropriate social behavior on the Net: Netiquette (etiquette on the Internet). The following are some illustrative examples pertaining primarily to e-mail.

- Shouting, THE PRACTICE OF TYPING ENTIRE SENTENCES IN UPPER CASE, is generally seen as novice (newbie) behavior and frowned upon. Perhaps it stems from the disgust with old teletypes and mainframe terminals that did not have lower-case characters.
- The use of emoticons to indicate the emotive content of a sentence is highly recommended. Typed text does not reveal any body language and a joking remark can easily be interpreted the wrong way. Whenever one writes something in jest or with humorous intent, it is advisable to add an emoticon. An emoticon (an icon indicating emotional content) consists of a series of text characters which are meant to be rotated a quarter turn and represent a laughing ? (i.e. equivalent to J or the smiley) winking ? or sad face ?
- Flaming is the carrying on of a heated personal emotional debate between two or more individuals on a public Internet forum. A flame war is generally a sign of immature behavior by individuals who cannot take perspective and should really take the discussion off-line.
- Netizens (inhabitants of the Internet i.e. frequent net surfers) often use standard but, to the non-initiated, cryptic abbreviations. Examples are: BTW = by the way ; ROFL = rolling on the floor with laughter ; TPTB = the powers that be; BRB = Be Right Back. This vocabulary has been adopted and expanded with the growth of Short Message Service (SMS) use on cellular phones.



:-)	Happy face
:-(	Sad or sorrow
;-)	Wink
:-0	Shock
:-\	Sarcasm
:^]	Wide grin
:-x	Blowing a kiss
:'(	Teary-eyed
:-P	Sticking out tongue
8-)=	Beard and glasses
:--)%	Boy on skateboard
<g>	Grin
<w>	Whisper
{{}}	Hug you

The Web

The Internet service that has received the most attention from the public media is the WorldWide Web or the Web for short (sometimes also called WWW or W3). The Web is a vast collection of multimedia information located on Web servers attached to the Internet.

Its popularity is due to a number of reasons.

- Information links are transparent. Links to any other piece of information located anywhere on the Internet can be inserted in a web document. A simple click of the mouse takes the reader completely automatically from one Web server to another, quite possibly in another country.
- Information can be presented in a hypertext link format whereby one can jump immediately from one concept to a related concept or explanation. No need to read text in the traditional top-to-bottom sequential way.
- It allows for multimedia information. A Web document can incorporate rich and colourful graphics, animation, video clips, sound etc. Just think of the marketing opportunities!
- The Web supports interactive applications. Web applications can request information from visiting users and documents can include programming instructions. Users can even download small programs (often written in Java) that could perform some processing on the user's computer or display special visual effects.

Reading or accessing information on the Web is called surfing the Net because one jumps from one hypertext link to another following whatever takes your fancy. In order to surf the Net you need some special browser program that understands the Web protocols and formats and presents the information to suit your computer monitor. You also need an access point or connection to the Internet. Your Internet connection may be automatic if your computer is connected to a (corporate) LAN that connects directly to the Internet, or it may be by means of a special subscription to a business that specializes in providing Internet access for others: the Internet Service Provider (ISP). Access to the ISP for individual users is usually via a dial-up connection i.e. using a modem and telephone.

Once a newbie (new user) is connected to the Internet (online), she faces the daunting task of finding her way amongst the huge variety of information offered. The easiest way in is usually by means of a search engine: a Web site that tries to catalogue the information available on the Internet. By entering one or more search words, the engine will provide you with a couple of adverts and a list of documents that contain the word(s) for which you are looking.

All information on the Web is uniquely identified by the URL (Uniform Resource Location), which is really the full Internet address of a Web document. The URL consists again of the Web server's domain address, followed by the access path and file name on the server. Examples of URLs are www.hotbot.com/sports/main.html (the main page on the sports section of the HotBot search engine) or http://www.commerce.uct.ac.za/informationsystems/ (containing details about UCT's department of information systems). Note the similarities and differences between an URL and an e-mail address.

Other Internet services

A number of other services are available on the Internet. The Usenet consists of ongoing discussion fora (or newsgroups) on an extremely wide variety of topics, from forensic psychology to Douglas Adams, from Star Trek to cryptography. The discussion happens entirely by means of e-mail and, when you subscribe to a given newsgroup, you can browse through the contributions of the last few days and reply with your own contribution.

More specialized services exist, such as ftp (file transfer protocol) for the transfer of large computer files, and telnet, the remote access of computers elsewhere, but they are used less frequently. In any way, these services are now being performed transparently by most Web browsers. Similarly, older services such as Gopher and Veronica have really been replaced almost entirely by the Web.

Internet protocols and standards

Different computers and networks can communicate via the Internet because a number of basic Internet communication standards have been defined. Any network connected to the Internet will translate its own standards and protocols into those used on the Internet by means of a bridge.

The most fundamental and "lowest level" protocol is the TCP/IP (Transmission Control Protocol/Internet Protocol). This protocol is also the native protocol of computers using the Unix operating system, which explains why Unix computers are so popular as Internet servers.

On top of TCP/IP are the "mid-level" protocols defined for the various Internet services. Perhaps the best known of these is http (Hypertext Transmission Protocol), which specifies how the Web information is made available and transmitted across the Internet. Other protocols and standards are STMP and MIME (for e-mail) or ftp.

```
      class="container">
    <div class="carousel-caption">
      <h1>One more for good measure.</h1>
      <p>Cras justo odio, dapibus ac facilisis in, egestas eget quam.
      .</p>
      <p><a class="btn btn-lg btn-primary" href="#" role="button">Browse gallery</a></p>
    </div>
  </div>
 </div>
</div>
<a class="left carousel-control" href="#myCarousel" role="button" data-slide="prev">
  <span class="glyphicon glyphicon-chevron-left" aria-hidden="true"></span>
  <span class="sr-only">Previous</span>
</a>
<a class="right carousel-control" href="#myCarousel" role="button" data-slide="next">
  <span class="glyphicon glyphicon-chevron-right" aria-hidden="true"></span>
  <span class="sr-only">Next</span>
</a>
</div><!-- /.carousel -->

<!--Featured Content Section-->

<div class="container">
  <div class="row">
    <div class="col-md-4"></div>
    <div class="col-md-4"> <h2> FEATURED CONTENT </h2> <hr class="featurette-divider">
    <div class="col-md-4"></div>
    <div class
```

HTML

Information made available via the Web is usually formatted using a special standard: the Hypertext Markup Language (HTML), which actually consists of plain text files with visual formatting commands inserted between the text. Most desktop productivity software allows you to save your document directly in the HTML format. Special HTML editors allow much finer control over the final layout of your Web document. A later development is Extensible Markup Language (XML), which increases the flexibility of web documents by allowing them to be viewed not only using a web browser, but also on different platforms such as a PDA or cellular telephone.

Licensing & Attributions

CC licensed content, Shared previously

- Discovering Information Systems Section 6.4. **Authored by:** Jean-Paul Van Belle, Mike Eccles, and Jane Nash. **Located at:** https://archive.org/details/ost-computer-science-discoveringinformationsystems. **License:** *CC BY-NC-ND: Attribution-NonCommercial-NoDerivatives*
- **Authored by:** geralt. **Located at:** https://pixabay.com/en/tree-structure-networks-internet-200795/. **License:** *CC0: No Rights Reserved*
- HTML. **Authored by:** lmonk72. **Located at:** https://pixabay.com/en/website-code-html-coding-647013/. **License:** *CC0: No Rights Reserved*
- Buenas noticias - email marketing. **Authored by:** RaHuL Rodriguez. **Located at:** https://www.flickr.com/photos/rahulrodriguez/9162677329/. **License:** *CC BY-SA: Attribution-ShareAlike*

READING: E-COMMERCE

Introduction

Business was quick to grasp the marketing and business potential offered by the Internet. Initially, businesses used the Internet to facilitate communication by means of e-mail. This was quickly followed by tapping the web's potential for the dissemination of product and other marketing information. The provision of advertising space {banners) on frequently visited web sites is the main source of income for search engines (sites allowing you to search the Internet for information) and web portals (web sites that provide additional value-added personal services such as news, financial information, weather forecasts, items of interest etc.)

A number of specialized companies have realized that the Internet can be a direct and extremely cost-effective channel of distribution. Some companies already have a physical infrastructure and use the web to enhance their distribution channel e.g. you can now order your pizza, bank statements or movie tickets via the web. Other, virtual companies have almost no physical infrastructure and are mere "conductors" for the flow of products of services.

Important categories of e-commerce include:

- Business-to-consumer (B2C) in which organizations provide information online to customers, who can in turn place orders and make payments via the internet
- Business-to-business (B2B) in which business partners collaborate electronically
- Consumer-to-consumer (C2C) in which individuals sell products or services directly to other individuals.

The technologies that are needed to support electronic commerce include the network infrastructure (Internet, intranets, extranets), software tools for web site development and maintenance, secure ordering and payment methods, and resources for information sharing, communication and collaboration. When e-commerce is done in a wireless environment, such as through the use of cellphones, this is referred to as mobile commerce (m-commerce).

B2C e-Commerce

Electronic retailing is similar in principle to home shopping from catalogues, but offers a wider variety of products and services, often at lower prices. Search engines make it easy to locate and compare competitor's products from one convenient location and without being restricted to usual shopping hours. Electronic malls provide access to a number of individual shops from one website. On-line auctions have also proved a popular way of disposing of items that need a quick sale.

Business-to-consumer commerce allows customers to make enquiries about products, place orders, pay accounts, and obtain service support via the Internet. Since customers can enter transactions at any time of the day or night, and from any geographical location, this can be a powerful tool for expanding the customer base of a business. However, the existence of a website does not guarantee that customers will use it, or that they will return to it after a first visit. Firms investing in electronic commerce need to consider a number of factors in developing and maintaining their e-commerce sites.

A successful web site should be attractive to look at and easy to use. In addition, it should offer its customers good performance, efficient service, personalization, incentives to purchase and security. Inadequate server power and communications capacity may cause customers to become frustrated when browsing or selecting products.

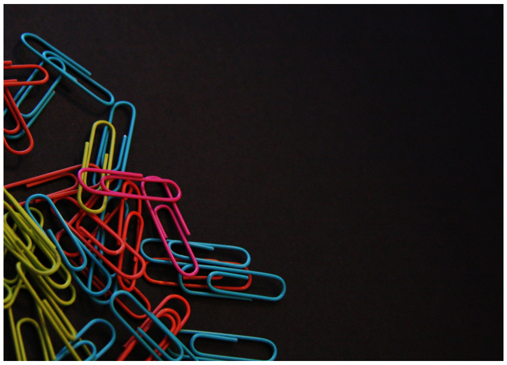
Many sites record details of their customers' interests, so that they can be guided to the appropriate parts of the site. Customer loyalty can also be developed by offering discussion forums and links to related sites, and by providing incentives such as discounts and special offers for regular customers. And if you expect customers to purchase goods, and not just browse, then it is vital that customers should have complete confidence in the security of their personal information, and in the ability of the web store to deliver the goods as requested.

Much of the business value of the Internet lies in the ability to provide increased value to customers, with the focus on quality of service rather than simply price. By opening additional channels of communication between the business and its customers, businesses can find out the preferences of their customers, and tailor products to their needs. Customers can use the Internet to ask questions, air complaints, or request product support, which increase customer involvement in business functions such as product development and service.

However, although businesses may increase their markets while gaining from reduced advertising and administration costs, problems that have emerged include alienation of regular distributors, difficulty in shipping small orders over large distances, fierce competition and inadequate profit margins. Because of the delivery problem for physical products, many successful e-commerce firms have focused on the delivery of services, such as banking, securities trading, employment agencies and travel bureaus. Of course, every problem can be regarded as an opportunity – a local software developer has created and marketed a route scheduling application which provides optimized route sheets, with maps for individual routes and step-by-step driving instructions for effective and timeous order management, based on powerful geographical information systems to provide a user-friendly interface.

B2C e-commerce has also made it easier for firms to conduct market research, not only by collecting shopping statistics, but also by using questionnaires to find out what specific groups of customers want. This in turn has enabled the personalization of products to meet customer preferences.

B2B e-Commerce



Business-to-business e-commerce comprises the majority of electronic transactions, involving the supply chain between organizations and their distributors, resellers, suppliers and other partners. Efficient management of the supply chain can cut costs, increase profits, improve relationships with customers and suppliers, and gain competitive advantage. To achieve this, firms need to

- Get the right product to the right place at the least cost;
- Keep inventory as low as possible while meeting customer requirements;
- Reduce cycle times by speeding up the acquisition and processing of raw materials.

Information technologies used to support business-to-business e-commerce include email, EDI and EFT, product catalogues, and order processing systems. These functions may be linked to traditional accounting and business information systems, to ensure that inventory and other databases are automatically updated via web transactions. Intranets provide a facility for members of an organization to chat, hold meetings and exchange information, while at the same time sensitive information is protected from unauthorized access by means of a firewall. An extranet provides a means of access to the intranet for authorized users such as business consultants.

Electronic data interchange (EDI) involves the electronic exchange of business transaction documents over computer networks, between organizations and their customers or suppliers. Value-added networks provided by third parties are frequently used for this purpose. Documents such as purchase orders, invoices and requests for quotations are electronically interchanged using standard message formats, which are specified by international protocols. EDI eliminates printing, postage and manual handling of documents, reducing time delays and errors, and thus increasing productivity. It also provides support for implementing a Justin-Time approach, which reduces lead time, lowers inventory levels, and frees capital for the business.

Marketing to other businesses is done by means of electronic catalogues and auction sites, which can increase sales while reducing advertising and administrative costs. From the buyer's perspective, reverse auctions can be used to advertise requests for quotation in a bidding marketplace in order to attract potential suppliers. Third party vendors can make use of group purchasing to aggregate a number of separate small orders in order to increase negotiating power.

Collaborative commerce involves long-term relationships between organizations in areas such as demand forecasting, inventory management, and product design and manufacture. However, this presents a number of business challenges such as software integration, compatibility of technologies, and building of trust between firms.

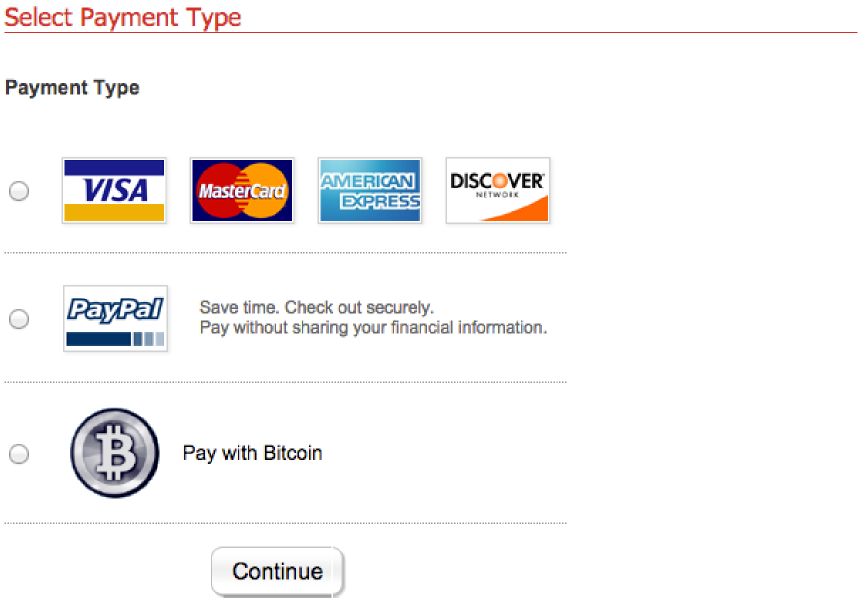
C2C e-Commerce

Auctions are the most popular method of conducting business between individuals over the Internet. (Unfortunately, auction fraud was also the most common type of crime reported to the Internet Fraud Complaint Centre in 2002.) Other C2C activities include classified advertising, selling of personal services such as astrology and medical advice, and the exchange of files especially music and computer games.

Electronic funds transfer

Electronic payment systems can be used to transfer funds between the bank accounts of a business and its suppliers, or from a customer to the business. In retail stores, wide area networks may connect POS terminals in retail stores to bank EFT systems. In most cases, an intermediary organization acts as an automated clearinghouse, which debits and credits the relevant accounts.

The most popular payment method used by individual consumers is the credit card, which requires the merchant to pay a commission to the bank on each transaction. For transactions involving small amounts that do not justify the payment of commission, merchants may accept electronic money in the form of digital cash In this case, the customer "buys" money from the bank in the form of a unique cash number, which is transmitted to the merchant at the time of purchase and "deposited" in an account at a participating bank. In South Africa several banks have developed their own forms of digital cash, such as e-bucks from First National Bank.



Select Payment Type

Payment Type

○ VISA MasterCard AMERICAN EXPRESS DISCOVER NETWORK

○ PayPal Save time. Check out securely.
Pay without sharing your financial information.

○ Pay with Bitcoin

An important issue in electronic commerce is the security of Internet transactions. Data is commonly encrypted to reduce the vulnerability of credit card transactions. Secure Sockets Layer (SSL) and Secure Electronics Transaction (SET) are two of the standards used to secure electronic payments on the Internet. Secure sites usually have URLs that begin with https instead of the usual http.

Current Issues in e-Commerce

For e-commerce to succeed, companies need to make large investments in hardware and telecommunications infrastructures that will be up and running 100% of the time, and software that is easy to use and reliable. A number of early participants in the e-commerce market suffered financial losses because their technology was not able to handle the huge numbers of transactions to be processed. Internet customers are often impatient, and will move to a competing site if the response is too slow.

Gaining the trust of customers can be difficult—the seller is often reluctant to despatch goods before payment, and the buyer may be reluctant to pay before receiving the goods. In South Africa, the speed of electronic ordering is often negated by delays in physical delivery.

Societal problems have also emerged, with children, gamblers and shopping addicts enjoying unrestricted access to electronic commerce sites. A German cannibal posted a web advertisement seeking a victim who was willing to be killed, sliced and eaten – and apparently found one! (reported in www.iol.co.za, 18 December 2002). The laws governing electronic commerce are still in their infancy, and international standards need to be developed in areas such as information privacy and taxation.

Since e-commerce supports global business transactions, it presents the challenge of customizing web sites to appeal to people of different nationalities and cultures (and even different languages). South Africa's leading role in e-commerce in Africa can probably be attributed to the fact that it has a relatively advanced telecommunication infrastructure and a large number of English-speaking users.

South African Perspective

Because a website can easily be developed as a front for a fraudulent company, businesses need a way to guarantee their authenticity to potential customers. Thawte Consulting, the company established by Mark Shuttleworth after graduating from UCT and later sold to market leader Verisign, provided (among other products) digital certificates which serve two purposes: to ensure that no sensitive information can be viewed by unauthorized users, and to provide users with assurance regarding the ownership of the site. By providing certificates at a lower cost than its competitors, but with similar technological and security standards, Thawte rapidly established itself as the second largest provider of digital certificates.

These non-forgeable Secure Sockets Layer (SSL) certificates are issued and digitally signed by a company such as Thawte, which has verified that the website really is owned by the organization requesting the certificate. Once the digital certificate has been installed on the site, the SSL uses complex encryption techniques to scramble confidential information.

Beyond the Basics

Encryption is the process of converting readable data into unreadable characters to prevent unauthorised access. Encrypted data can be safely transmitted or stored, but must be decrypted before it can be read, by using an encryption key, which is sometimes the same formula that was used to scramble it in the first place. Simple encryption methods include:

- Transposition: in which the order of characters is switched, for example each pair of adjacent characters is swapped.
- Substitution: in which each character is replaced by some other predetermined character.
- Expansion: additional letters are inserted after each of the characters in the original text.
- Compaction: characters are removed from specific positions and then stored or transmitted separately.

Most encryption programs use a combination of all four methods.

Private key encryption relies on both sender and recipient having access to the same encryption key. Public key encryption makes use of two keys: a message encrypted with your public key can only be decrypted using your private key. This means that you can safely communicate your public key to business contacts, who are then able to send you confidential data that can only be read using your private key. Security agencies in the United States have

lobbied for some time for private keys to be independently stored, so that encrypted communications could be monitored when national security is considered to be at risk.

Reflection Questions

B2C e-commerce

- What advantages does the customer stand to gain from B2C e-commerce, compared with traditional business models?
- Can you think of any potential disadvantages?

C2C e-commerce

- Give reasons why internet auctions are a common source of fraud, and suggest control structures that could be put in place to reduce this problem.

B2B e-commerce

- Explain how B2B e-commerce could contribute to each of the alternative strategies for competitive advantage (low-cost, differentiation, niche marketing) that were described in the previous chapter.

Licensing & Attributions

CC licensed content, Shared previously

- Discovering Information Systems Chapter 9. **Authored by:** Jean-Paul Van Belle, Mike Eccles, and Jane Nash. **Located at:** https://archive.org/details/ost-computer-science-discoveringinformationsystems. **License:** *CC BY-NC-ND: Attribution-NonCommercial-NoDerivatives*
- Pay with Bitcoin. **Authored by:** Lars Plougmann. **Located at:** https://www.flickr.com/photos/criminalintent/15300343255/. **License:** *CC BY-SA: Attribution-ShareAlike*
- Internet shopping cart. **Authored by:** geralt. **Located at:** https://pixabay.com/en/shopping-cart-internet-728409/. **License:** *CC0: No Rights Reserved*
- Bottom of the Desk Drawer. **Authored by:** Krissy Venosdale. **Located at:** https://www.flickr.com/photos/venosdale/4617152976/. **License:** *CC BY-NC-ND: Attribution-NonCommercial-NoDerivatives*

READING: THE INTERNET

Introduction

The **Internet** is a global system of interconnected computer networks that use the standard Internet protocol suite (TCP/IP) to link several billion devices worldwide. It is a *network of networks* that consists of millions of private, public, academic, business, and government networks of local to global scope, linked by a broad array of electronic, wireless, and optical networking technologies.

The Internet carries an extensive range of information resources and services, such as the inter-linked hypertext documents and applications of the World Wide Web (WWW), the infrastructure to support email, and peer-to-peer networks for file sharing and telephony.

The origins of the Internet date back to research commissioned by the United States government in the 1960s to build robust, fault-tolerant communication via computer networks. This work, combined with efforts in the United Kingdom and France, led to the primary precursor network, the ARPANET, in the United States. The interconnection of regional academic networks in the 1980s marks the beginning of the transition to the modern Internet. From the early 1990s, the network experienced sustained exponential growth as generations of institutional, personal, and mobile computers were connected to it.

The Internet Messenger by Buky Schwartz in Holon.

The funding of a new U.S. backbone by the National Science Foundation in the 1980s, as well as private funding for other commercial backbones, led to worldwide participation in the development of new networking technologies, and the merger of many networks. Though the Internet has been widely used by academia since the 1980s, the commercialization of what was by the 1990s an international network resulted in its popularization and incorporation into virtually every aspect of modern human life. As of 2014, 38 percent of the world’s human population has used the services of the Internet within the past year–over 100 times more people than were using it in 1995. Internet use grew rapidly in the West from the mid-1990s to early 2000s and from the late 1990s to present in the developing world.

Most traditional communications media, including telephony and television, are being reshaped or redefined by the Internet, giving birth to new services such as voice over Internet Protocol (VoIP) and Internet Protocol television (IPTV). Newspaper, book, and other print publishing are adapting to website technology, or are reshaped into blogging and web feeds. The entertainment industry, including music, film, and gaming, was initially the fastest growing online segment. The Internet has enabled and accelerated new forms of human interactions through instant messaging, Internet forums, and social networking. Online shopping has grown exponentially both for major retailers and small artisans and traders. Business-to-business and financial services on the Internet affect supply chains across entire industries.

The Internet has no centralized governance in either technological implementation or policies for access and usage; each constituent network sets its own policies. Only the overreaching definitions of the two principal name spaces in the Internet, the Internet Protocol address space and the Domain Name System (DNS), are directed by a maintainer organization, the Internet Corporation for Assigned Names and Numbers (ICANN). The technical underpinning and standardization of the core protocols is an activity of the Internet Engineering Task Force (IETF), a non-profit organization of loosely affiliated international participants that anyone may associate with by contributing technical expertise.

Terminology

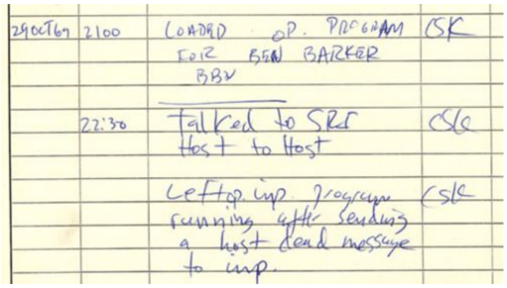
The *Internet*, referring to the specific global system of interconnected IP networks, is a proper noun and may be written with an initial capital letter. In the media and common use it is often not capitalized, viz. *the internet.* Some guides specify that the word should be capitalized when used as a noun, but not capitalized when used as an adjective. The Internet is also often referred to as *the Net*.

Historically the word *internetted* was used, uncapitalized, as early as 1849 as an adjective meaning "Interconnected; interwoven". The designers of early computer networks used *internet* both as a noun and as a verb in shorthand form of internetwork or internetworking, meaning interconnecting computer networks.

The terms *Internet* and *World Wide Web* are often used interchangeably in everyday speech; it is common to speak of "going on the Internet" when invoking a web browser to view web pages. However, the World Wide Webor *the Web* is only one of a large number of Internet services. The Web is a collection of interconnected documents (web pages) and other web resources, linked by hyperlinks and URLs. As another point of comparison, Hypertext Transfer Protocol, or HTTP, is the language used on the Web for information transfer, yet it is just one of many languages or protocols that can be used for communication on the Internet.

The term *Interweb* is a portmanteau of *Internet* and *World Wide Web* typically used sarcastically to parody a technically unsavvy user.

History



29OCT69	2100	LOADED OP. PROGRAM FOR BEN BARKER BBN	CSK
	22:30	Talked to SRI Host to Host	CSK
		Left op. imp program running after sending a host dead message to imp.	CSK

Text from the very first message ever sent via the ARPANET.

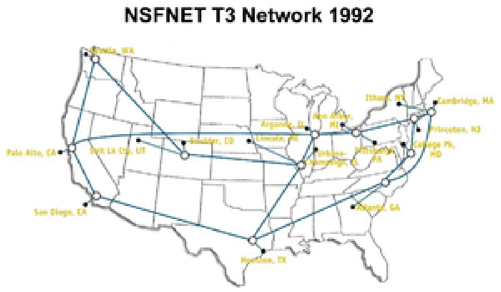
Research into packet switching started in the early 1960s and packet switched networks such as Mark I at NPL in the UK, ARPANET, CYCLADES, Merit Network, Tymnet, and Telenet, were developed in the late 1960s and early 1970s using a variety of protocols. The ARPANET in particular led to the development of protocols for internet working, where multiple separate networks could be joined together into a network of networks.

The first two nodes of what would become the ARPANET were interconnected between Leonard Kleinrock's Network Measurement Center at the UCLA's School of Engineering and Applied Science and Douglas Engelbart's NLS system at SRI International (SRI) in Menlo Park, California, on 29 October 1969. The third site on the ARPANET was the Culler-Fried Interactive Mathematics center at the University of California at Santa Barbara, and the fourth was the University of UtahGraphics Department. In an early sign of future growth, there were already fifteen sites connected to the young ARPANET by the end of 1971. These early years were documented in the 1972 film Computer Networks: The Heralds of Resource Sharing.

Early international collaborations on the ARPANET were rare. European developers were concerned with developing the X.25networks. Notable exceptions were the *Norwegian Seismic Array* (NORSAR) in June 1973, followed in 1973 by Sweden with satellite links to the Tanum Earth Station and Peter T. Kirstein's research group in the UK, initially at the Institute of Computer Science, University of London and later at University College London.

In December 1974, *RFC 675 – Specification of Internet Transmission Control Program*, by Vinton Cerf, Yogen Dalal, and Carl Sunshine, used the term *internet* as a shorthand for *internet working* and later RFCs repeat this use. Access to the ARPANET was expanded in 1981 when the National Science Foundation (NSF) developed the Computer Science Network (CSNET). In 1982, the Internet Protocol Suite (TCP/IP) was standardized and the concept of a world-wide network of fully interconnected TCP/IP networks called the Internet was introduced.

TCP/IP network access expanded again in 1986 when the National Science Foundation Network (NSFNET) provided access to supercomputer sites in the United States from research and education organizations, first at 56 kbit/s and later at 1.5 Mbit/s and 45 Mbit/s. Commercial Internet service providers (ISPs) began to emerge in the late 1980s and early 1990s. The ARPANET was decommissioned in 1990. The Internet was fully commercialized in the U.S. by 1995 when NSFNET was decommissioned, removing the last restrictions on the use of the Internet to carry commercial traffic. The Internet started a rapid expansion to Europe and Australia in the mid to late 1980s and to Asia in the late 1980s and early 1990s.



T3 NSFNET Backbone, c. 1992.

Since the mid-1990s the Internet has had a tremendous impact on culture and commerce, including the rise of near instant communication by email, instant messaging, Voice over Internet Protocol (VoIP) “phone calls”, two-way interactive video calls, and the World Wide Web with its discussion forums, blogs, social networking, and online shopping sites. Increasing amounts of data are transmitted at higher and higher speeds over fiber optic networks operating at 1-Gbit/s, 10-Gbit/s, or more.

Worldwide Internet users

	2005	2010	2014[a]
World population	6.5 billion	6.9 billion	7.2 billion
Not using the Internet	84%	70%	60%
Using the Internet	16%	30%	40%
Users in the developing world	8%	21%	32%
Users in the developed world	51%	67%	78%

[a] Estimate. Source: International Telecommunications Union.

The Internet continues to grow, driven by ever greater amounts of online information and knowledge, commerce, entertainment and social networking. During the late 1990s, it was estimated that traffic on the public Internet grew by 100 percent per year, while the mean annual growth in the number of Internet users was thought to be between 20% and 50%. This growth is often attributed to the lack of central administration, which allows organic growth of the network, as well as the non-proprietary open nature of the Internet protocols, which encourages vendor interoperability and prevents any one company from exerting too much control over the network. As of 31 March 2011, the estimated total number of Internet users was 2.095 billion (30.2% of world population). It is estimated that in 1993 the Internet carried only 1% of the information flowing through two-way telecommunication, by 2000 this figure had grown to 51%, and by 2007 more than 97% of all telecommunicated information was carried over the Internet.

Governance

ICANN headquarters in the Playa Vista neighborhood of Los Angeles, California, United States.

The Internet is a globally distributed network comprising many voluntarily interconnected autonomous networks. It operates without a central governing body.

The technical underpinning and standardization of the core protocols (IPv4 and IPv6) is an activity of the Internet Engineering Task Force (IETF), a non-profit organization of loosely affiliated international participants that anyone may associate with by contributing technical expertise.

To maintain interoperability, the principal name spaces of the Internet are administered by the Internet Corporation for Assigned Names and Numbers (ICANN), headquartered in the neighborhood of Playa Vista in the city of Los Angeles, California. ICANN is the authority that coordinates the assignment of unique identifiers for use on the Internet, including domain names, Internet Protocol (IP) addresses, application port numbers in the transport protocols, and many other parameters. Globally unified name spaces, in which names and numbers are uniquely assigned, are essential for maintaining the global reach of the Internet. ICANN is governed by an international board of directors drawn from across the Internet technical, business, academic, and other non-commercial communities. ICANN's role in coordinating the assignment of unique identifiers distinguishes it as perhaps the only central coordinating body for the global Internet.

Regional Internet Registries (RIRs) allocate IP addresses:

- African Network Information Center (AfriNIC) for Africa
- American Registry for Internet Numbers (ARIN) for North America
- Asia-Pacific Network Information Centre (APNIC) for Asia and the Pacific region
- Latin American and Caribbean Internet Addresses Registry (LACNIC) for Latin America and the Caribbean region
- Réseaux IP Européens – Network Coordination Centre (RIPE NCC) for Europe, the Middle East, and Central Asia

The National Telecommunications and Information Administration, an agency of the United States Department of Commerce, continues to have final approval over changes to the DNS root zone.

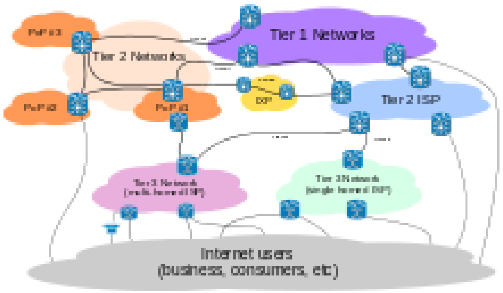
The Internet Society (ISOC) was founded in 1992 with a mission to *"assure the open development, evolution and use of the Internet for the benefit of all people throughout the world"*. Its members include individuals (anyone may join) as well as corporations, organizations, governments, and universities. Among other activities ISOC provides an administrative home for a number of less formally organized groups that are involved in developing and managing the Internet, including: the Internet Engineering Task Force (IETF), Internet Architecture Board (IAB), Internet Engineering Steering Group (IESG), Internet Research Task Force (IRTF), and Internet Research Steering Group (IRSG).

On 16 November 2005, the United Nations-sponsored World Summit on the Information Society, held in Tunis, established the Internet Governance Forum(IGF) to discuss Internet-related issues.

Infrastructure

The communications infrastructure of the Internet consists of its hardware components and a system of software layers that control various aspects of the architecture.

Routing and service tiers



Packet routing across the Internet involves several tiers of Internet service providers.

Internet service providers establish the world-wide connectivity between individual networks at various levels of scope. End-users who only access the Internet when needed to perform a function or obtain information, represent the bottom of the routing hierarchy. At the top of the routing hierarchy are the tier 1 networks, large telecommunication companies that exchange traffic directly with each other via peering agreements. Tier 2 and lower level networks buyInternet transit from other providers to reach at least some parties on the global Internet, though they may also engage in peering. An ISP may use a single upstream provider for connectivity, or implement multihoming to achieve redundancy and load balancing. Internet exchange points are major traffic exchanges with physical connections to multiple ISPs.

Large organizations, such as academic institutions, large enterprises, and governments, may perform the same function as ISPs, engaging in peering and purchasing transit on behalf of their internal networks. Research networks tend to interconnect with large subnetworks such as GEANT, GLORIAD, Internet2, and the UK's national research and education network, JANET.

It has been determined that both the Internet IP routing structure and hypertext links of the World Wide Web are examples of scale-free networks.

Computers and routers use routing tables in their operating system to direct IP packets to the next-hop router or destination. Routing tables are maintained by manual configuration or automatically by routing protocols. End-nodes typically use a default route that points toward an ISP providing transit, while ISP routers use the Border Gateway Protocol to establish the most efficient routing across the complex connections of the global Internet.

Access

Common methods of Internet access by users include dial-up with a computer modem via telephone circuits, broadband over coaxial cable, fiber optic or copper wires, Wi-Fi, satellite and cellular telephone technology (3G, 4G). The Internet may often be accessed from computers in libraries and Internet cafes.Internet access points exist in many public places such as airport halls and coffee shops. Various terms are used, such as *public Internet kiosk*, *public access terminal*, and *Web payphone*. Many hotels also have public terminals, though these are usually fee-based. These terminals are widely accessed for various usage, such as ticket booking, bank deposit, or online payment. Wi-Fi provides wireless access to the Internet via local computer networks. Hotspots providing such access include Wi-Fi cafes, where users need to bring their own wireless-enabled devices such as a laptop or PDA. These services may be free to all, free to customers only, or fee-based.

Grassroots efforts have led to wireless community networks. Commercial Wi-Fi services covering large city areas are in place in London, Vienna, Toronto, San Francisco, Philadelphia, Chicago and Pittsburgh. The Internet can then be accessed from such places as a park bench. Apart from Wi-Fi, there have been experiments with proprietary mobile wireless networks like Ricochet, various high-speed data services over cellular phone networks, and fixed wireless services. High-end mobile phones such as smartphones in general come with Internet access through the phone network. Web browsers such as Opera are available on these advanced handsets, which can also run a wide variety of other Internet software. More mobile phones have Internet access than PCs, though this is not as widely used. An Internet access provider and protocol matrix differentiates the methods used to get online.

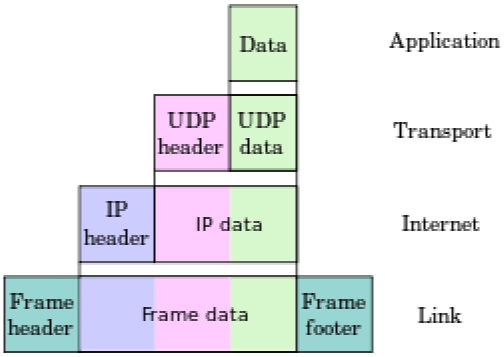
Protocols

While the hardware components in the Internet infrastructure can often be used to support other software systems, it is the design and the standardization process of the software that characterizes the Internet and provides the foundation for its scalability and success. The responsibility for the architectural design of the Internet software systems has been assumed by the Internet Engineering Task Force (IETF). The IETF conducts standard-setting work groups, open to any individual, about the various aspects of Internet architecture. Resulting contributions and standards are published as *Request for Comments* (RFC) documents on the IETF web site.

The principal methods of networking that enable the Internet are contained in specially designated RFCs that constitute the Internet Standards. Other less rigorous documents are simply informative, experimental, or historical, or document the best current practices (BCP) when implementing Internet technologies.

The Internet standards describe a framework known as the Internet protocol suite. This is a model architecture that divides methods into a layered system of protocols, originally documented in RFC 1122 and RFC 1123. The layers correspond to the environment or scope in which their services operate. At the top is the application layer, the space for the application-specific networking methods used in software applications. For example, a web browser program uses the client-server application model and a specific protocol of interaction between servers and clients, while many file-sharing systems use a peer-to-peer paradigm. Below this top layer, the transport layer connects applications on different hosts with a logical channel through the network with appropriate data exchange methods.

Underlying these layers are the networking technologies that interconnect networks at their borders and hosts via the physical connections. The internet layeren ables computers to identify and locate each other via Internet Protocol (IP) addresses, and routes their traffic via intermediate (transit) networks. Last, at the bottom of the architecture is the link layer, which provides connectivity between hosts on the same network link, such as a physical connection in form of a local area network (LAN) or a dial-up connection. The model, also known as TCP/IP, is designed to be independent of the underlying hardware, which the model therefore does not concern itself with in any detail. Other models have been developed, such as the OSI model, that attempt to be comprehensive in every aspect of communications. While many similarities exist between the models, they are not compatible in the details of description or implementation; indeed, TCP/IP protocols are usually included in the discussion of OSI networking.



As user data is processed through the protocol stack, each abstraction layer adds encapsulation information at the sending host. Data is transmitted over the wire at the link level between hosts and routers. Encapsulation is removed by the receiving host. Intermediate relays update link encapsulation at each hop, and inspect the IP layer for routing purposes.

The most prominent component of the Internet model is the Internet Protocol (IP), which provides addressing systems (IP addresses) for computers on the Internet. IP enables internet working and in essence establishes the Internet itself. Internet Protocol Version 4 (IPv4) is the initial version used on the first generation of the Internet and is still in dominant use. It was designed to address up to ~4.3 billion (10^9) Internet hosts. However, the explosive growth of the Internet has led to IPv4 address exhaustion, which entered its final stage in 2011, when the global address allocation pool was exhausted. A new protocol version, IPv6, was developed in the mid-1990s, which provides vastly larger addressing capabilities and more efficient routing of Internet traffic. IPv6 is currently in growing deployment around the world, since Internet address registries (RIRs) began to urge all resource managers to plan rapid adoption and conversion.

IPv6 is not directly interoperable by design with IPv4. In essence, it establishes a parallel version of the Internet not directly accessible with IPv4 software. This means software upgrades or translator facilities are necessary for networking devices that need to communicate on both networks. Essentially all modern computer operating systems support both versions of the Internet Protocol. Network infrastructure, however, is still lagging in this development. Aside from the complex array of physical connections that make up its infrastructure, the Internet is facilitated by bi- or multi-lateral commercial contracts, e.g., peering agreements, and by technical specifications or protocols that describe how to exchange data over the network. Indeed, the Internet is defined by its interconnections and routing policies.

Services

The Internet carries many network services, most prominently the World Wide Web, electronic mail, Internet telephony, and File sharing services.

World Wide Web

Many people use the terms *Internet* and *World Wide Web*, or just the *Web*, interchangeably, but the two terms are not synonymous. The World Wide Web is only one of hundreds of services used on the Internet. The Web is a global set of documents, images and other resources, logically interrelated by hyperlinks and referenced with Uniform Resource Identifiers (URIs). URIs symbolically identify services, servers, and other databases, and the documents and resources that they can provide. Hypertext Transfer Protocol (HTTP) is the main access protocol of the World Wide Web. Web services also use HTTP to allow software systems to communicate in order to share and exchange business logic and data.

This NeXT Computer was used by Tim Berners-Lee at CERN and became the world’s first Web server.

World Wide Web browser software, such as Microsoft’s Internet Explorer, Mozilla Firefox, Opera, Apple’s Safari, and Google Chrome, lets users navigate from one web page to another via hyperlinks embedded in the documents. These documents may also contain any combination of computer data, including graphics, sounds, text, video, multimedia and interactive content that runs while the user is interacting with the page. Client-side software can include animations, games, office applications and scientific demonstrations. Through keyword-driven Internet research using search engines like Yahoo! and Google, users worldwide have easy, instant access to a vast and diverse amount of online information. Compared to printed media, books, encyclopedias and traditional libraries, the World Wide Web has enabled the decentralization of information on a large scale.

The Web has also enabled individuals and organizations to publish ideas and information to a potentially large audience online at greatly reduced expense and time delay. Publishing a web page, a blog, or building a website involves little initial cost and many cost-free services are available. However, publishing and maintaining large, professional web sites with attractive, diverse and up-to-date information is still a difficult and expensive proposition. Many individuals and some companies and groups use *web logs* or blogs, which are largely used as easily updatable online diaries. Some commercial organizations encouragestaff to communicate advice in their areas of specialization in the hope that visitors will be impressed by the expert knowledge and free information, and be attracted to the corporation as a result.

One example of this practice is Microsoft, whose product developers publish their personal blogs in order to pique the public’s interest in their work. Collections of personal web pages published by large service providers remain popular, and have become increasingly sophisticated. Whereas operations such as Angelfire and GeoCities have existed since the early days of the Web, newer offerings from, for example, Facebook and Twitter currently have large followings. These operations often brand themselves as social network services rather than simply as web page hosts.

Advertising on popular web pages can be lucrative, and e-commerce or the sale of products and services directly via the Web continues to grow.

When the Web developed in the 1990s, a typical web page was stored in completed form on a web server, formatted in HTML, complete for transmission to a web browser in response to a request. Over time, the process of creating and serving web pages has become dynamic, creating flexible design, layout, and content. Websites are often created using content management software with, initially, very little content. Contributors to these systems, who may be paid staff, members of an organization or the public, fill underlying databases with content using editing pages designed for that purpose, while casual visitors view and read this content in HTML form. There may or may not be editorial, approval and security systems built into the process of taking newly entered content and making it available to the target visitors.

Communication

Email is an important communications service available on the Internet. The concept of sending electronic text messages between parties in a way analogous to mailing letters or memos predates the creation of the Internet. Pictures, documents and other files are sent as email attachments. Emails can be cc-ed to multiple email addresses.

Internet telephony is another common communications service made possible by the creation of the Internet. VoIP stands for Voice-over-Internet Protocol, referring to the protocol that underlies all Internet communication. The idea began in the early 1990s with walkie-talkie-like voice applications for personal computers. In recent years many VoIP systems have become as easy to use and as convenient as a normal telephone. The benefit is that, as the Internet carries the voice traffic, VoIP can be free or cost much less than a traditional telephone call, especially over long distances and especially for those with always-on Internet connections such as cable or ADSL. VoIP is maturing into a competitive alternative to traditional telephone service. Interoperability between different providers has improved and the ability to call or receive a call from a traditional telephone is available. Simple, inexpensive VoIP network adapters are available that eliminate the need for a personal computer.

Voice quality can still vary from call to call, but is often equal to and can even exceed that of traditional calls. Remaining problems for VoIP include emergency telephone number dialing and reliability. Currently, a few VoIP providers provide an emergency service, but it is not universally available. Older traditional phones with no “extra features” may be line-powered only and operate during a power failure; VoIP can never do so without a backup power source for the phone equipment and the Internet access devices. VoIP has also become increasingly popular for gaming applications, as a form of communication between players. Popular VoIP clients for gaming include Ventrilo and Teamspeak. Modern video game consoles also offer VoIP chat features.

Data transfer

File sharing is an example of transferring large amounts of data across the Internet. A computer file can be emailed to customers, colleagues and friends as an attachment. It can be uploaded to a website or FTP server for easy download by others. It can be put into a “shared location” or onto a file server for instant use by colleagues. The load of bulk downloads to many users can be eased by the use of “mirror” servers or peer-to-peer networks. In any of these cases, access to the file may be controlled by user authentication, the transit of the file over the Internet may be obscured by encryption, and money may change hands for access to the file. The price can be paid by the remote charging of funds from, for example, a credit card whose details are also passed – usually fully encrypted – across the Internet. The origin and authenticity of the file received may be checked by digital signatures or by MD5 or other message digests. These simple features of the Internet, over a worldwide basis, are changing the production, sale, and distribution of anything that can be reduced to a computer file for transmission. This includes all manner of print publications, software products, news, music, film, video, photography, graphics and the other arts. This in turn has caused seismic shifts in each of the existing industries that previously controlled the production and distribution of these products.

Streaming media is the real-time delivery of digital media for the immediate consumption or enjoyment by end users. Many radio and television broadcasters provide Internet feeds of their live audio and video productions. They may also allow time-shift viewing or listening such as Preview, Classic Clips and Listen Again features. These providers have been joined by a range of pure Internet “broadcasters” who never had on-air licenses. This means that an Internet-connected device, such as a computer or something more specific, can be used to access on-line media in much the same way as was previously possible only with a television or radio receiver. The range of available types of content is much wider, from specialized technical webcasts to on-demand popular multimedia services. Podcasting is a variation on this theme, where – usually audio – material is downloaded and played back on a computer or shifted to a portable media player to be listened to on the move. These techniques using simple equipment allow anybody, with little censorship or licensing control, to broadcast audio-visual material worldwide.

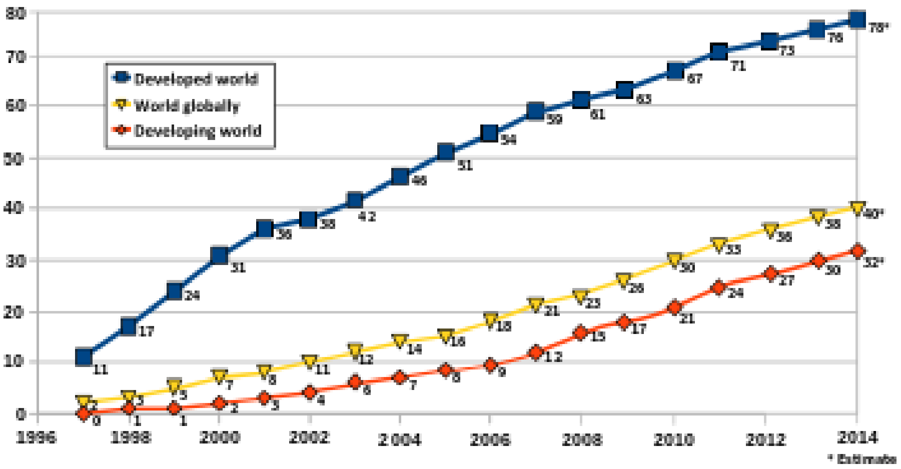
Digital media streaming increases the demand for network bandwidth. For example, standard image quality needs 1 Mbit/s link speed for SD 480p, HD 720p quality requires 2.5 Mbit/s, and the top-of-the-line HDX quality needs 4.5 Mbit/s for 1080p.

Webcams are a low-cost extension of this phenomenon. While some webcams can give full-frame-rate video, the picture either is usually small or updates slowly. Internet users can watch animals around an African waterhole, ships in the Panama Canal, traffic at a local roundabout or monitor their own premises, live and in real time. Video chat rooms and video conferencing are also popular with many uses being found for personal webcams, with and without two-way sound. YouTube was founded on 15 February 2005 and is now the leading website for free streaming video with a vast number of users. It uses a flash-based web player to stream and show video files. Registered users may upload an unlimited amount of video and build their own personal profile. YouTube claims that its users watch hundreds of millions, and upload hundreds of thousands of videos daily. Currently, YouTube also uses an HTML5 player.

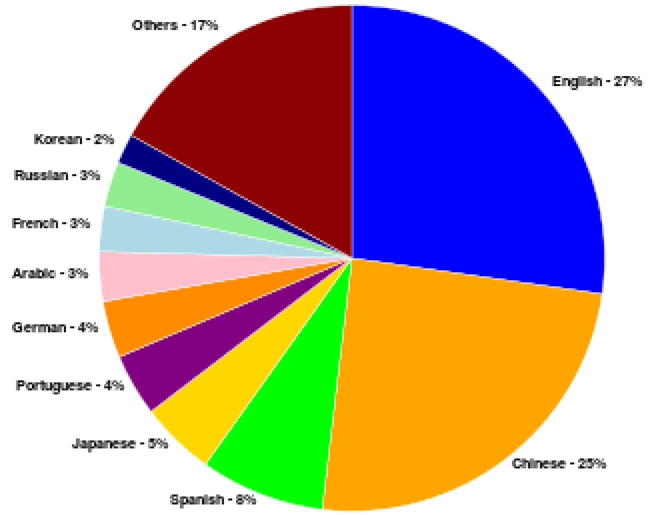
Social impact

The Internet has enabled new forms of social interaction, activities, and social associations.

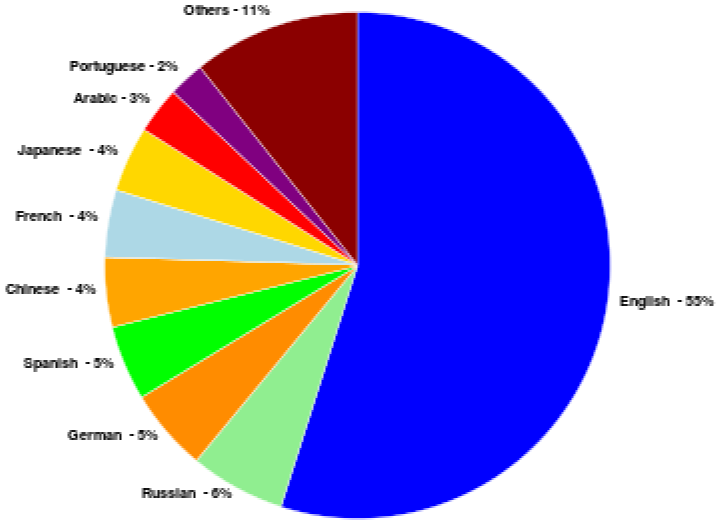
Users



Internet users per 100 inhabitants



Internet users by language.



April 2013

Website content languages.

Overall Internet usage has seen tremendous growth. From 2000 to 2009, the number of Internet users globally rose from 394 million to 1.858 billion. By 2010, 22 percent of the world's population had access to computers with 1 billion Google searches every day, 300 million Internet users reading blogs, and 2 billion videos viewed daily on YouTube. In 2014 the world's Internet users surpassed 3 billion or 43.6 percent of world population, but two-thirds of the users came from richest countries, with 78.0 percent of Europe countries population using the Internet, followed by 57.4 percent of the Americas.

The prevalent language for communication on the Internet has been English. This may be a result of the origin of the Internet, as well as the language's role as a lingua franca. Early computer systems were limited to the characters in the American Standard Code for Information Interchange (ASCII), a subset of the Latin alphabet.

After English (27%), the most requested languages on the World Wide Web are Chinese (25%), Spanish (8%), Japanese (5%), Portuguese and German (4% each), Arabic, French and Russian (3% each), and Korean (2%). By region, 42% of the world's Internet users are based in Asia, 24% in Europe, 14% in North America, 10% in Latin America and the Caribbean taken together, 6% in Africa, 3% in the Middle East and 1% in Australia/ Oceania. The Internet's technologies have developed enough in recent years, especially in the use of Unicode, that good facilities are available for development and communication in the world's widely used languages. However, some glitches such as *mojibake* (incorrect display of some languages' characters) still remain.

In an American study in 2005, the percentage of men using the Internet was very slightly ahead of the percentage of women, although this difference reversed in those under 30. Men logged on more often, spent more time online, and were more likely to be broadband users, whereas women tended to make more use of opportunities to communicate (such as email). Men were more likely to use the Internet to pay bills, participate in auctions, and for recreation such as downloading music and videos. Men and women were equally likely to use the Internet for shopping and banking. More recent studies indicate that in 2008, women significantly outnumbered men on most social networking sites, such as Facebook and Myspace, although the ratios varied with age. In addition, women watched more streaming content, whereas men downloaded more. In terms of blogs, men were more likely to blog in the first place; among those who blog, men were more likely to have a professional blog, whereas women were more likely to have a personal blog.

According to forecasts by Euromonitor International, 44% of the world's population will be users of the Internet by 2020. Splitting by country, in 2012 Iceland, Norway, Sweden, the Netherlands, and Denmark had the highest Internet penetration by the number of users, with 93% or more of the population with access.

Several neologisms exist that refer to Internet users: Netizen (as in as in "citizen of the net") refers to those actively involved in improving online communities, the Internet in general or surrounding political affairs and rights such as free speech, Internaut refers to operators or technically highly capable users of the Internet, digital citizen refers to a person using the Internet in order to engage in society, politics, and government participation.

Usage

The Internet allows greater flexibility in working hours and location, especially with the spread of unmetered high-speed connections. The Internet can be accessed almost anywhere by numerous means, including through mobile Internet devices. Mobile phones, data cards, handheld game consoles and cellular routers allow users to connect to the Internet wirelessly. Within the limitations imposed by small screens and other limited facilities of such pocket-sized devices, the services of the Internet, including email and the web, may be available. Service providers may restrict the services offered and mobile data charges may be significantly higher than other access methods.

Educational material at all levels from pre-school to post-doctoral is available from websites. Examples range from CBeebies, through school and high-school revision guides and virtual universities, to access to top-end scholarly literature through the likes of Google Scholar. For distance education, help with homework and other assignments, self-guided learning, whiling away spare time, or just looking up more detail on an interesting fact, it has never been easier for people to access educational information at any level from anywhere. The Internet in general and the World Wide Web in particular are important enablers of both formal and informal education. Further, the Internet allows universities, in particular researchers from the social and behavioral sciences, to conduct research remotely via virtual laboratories, with profound changes in reach and generalizability of findings as well as in communication between scientists and in the publication of results.

The low cost and nearly instantaneous sharing of ideas, knowledge, and skills has made collaborative work dramatically easier, with the help of collaborative software. Not only can a group cheaply communicate and share ideas but the wide reach of the Internet allows such groups more easily to form. An example of this is the free software movement, which has produced, among other things, Linux, Mozilla Firefox, and OpenOffice.org. Internet chat, whether using an IRC chat room, an instant messaging system, or a social networking website, allows colleagues to stay in touch in a very convenient way while working at their computers during the day. Messages can be exchanged even more quickly and conveniently than via email. These systems may allow files to be exchanged, drawings and images to be shared, or voice and video contact between team members.

Content management systems allow collaborating teams to work on shared sets of documents simultaneously without accidentally destroying each other's work. Business and project teams can share calendars as well as documents and other information. Such collaboration occurs in a wide variety of areas including scientific research, software development, conference planning, political activism and creative writing. Social and political collaboration is also becoming more widespread as both Internet access and computer literacy spread.

The Internet allows computer users to remotely access other computers and information stores easily, wherever they may be. They may do this with or without computer security, i.e. authentication and encryption technologies, depending on the requirements. This is encouraging new ways of working from home, collaboration and information sharing in many industries. An accountant sitting at home can audit the books of a company based in another country, on a server situated in a third country that is remotely maintained by IT specialists in a fourth. These accounts could have been created by home-working bookkeepers, in other remote locations, based on information emailed to them from offices all over the world. Some of these things were possible before the widespread use of the Internet, but the cost of private leased lines would have made many of them infeasible in practice. An office worker away from their desk, perhaps on the other side of the world on a business trip or a holiday, can access their emails, access their data using cloud computing, or open a remote desktop session into their office PC using a secure Virtual Private Network (VPN) connection on the Internet. This can give the worker complete access to all of their normal files and data, including email and other applications, while away from the office. It has been referred to among system administrators as the Virtual Private Nightmare, because it extends the secure perimeter of a corporate network into remote locations and its employees' homes.

Social networking and entertainment

Many people use the World Wide Web to access news, weather and sports reports, to plan and book vacations and to pursue their personal interests. People use chat, messaging and email to make and stay in touch with friends worldwide, sometimes in the same way as some previously had pen pals.

Social networking websites such as Facebook, Twitter, and Myspace have created new ways to socialize and interact. Users of these sites are able to add a wide variety of information to pages, to pursue common interests, and to connect with others. It is also possible to find existing acquaintances, to allow communication among existing groups of people. Sites like LinkedIn foster commercial and business connections. YouTube and Flickr specialize in users' videos and photographs.

While social networking sites were initially for individuals only, today they are widely used by businesses and other organizations to promote their brands, to market to their customers and to encourage posts to "go viral". "Black hat" social media techniques are also employed by some organizations, such as spam accounts and astroturfing.

A risk for both individuals and organizations writing posts (especially public posts) on social networking websites, is that especially foolish or controversial posts occasionally lead to an unexpected and possibly large-scale backlash on social media from other internet users. This is also a risk in relation to controversial *offline* behavior, if it is widely made known. The nature of this backlash can range widely from counter-arguments and public mockery, through insults and hate speech, to, in extreme cases, rape and death threats. The online disinhibition effect describes the tendency of many individuals to behave more stridently or offensively online than they would in person. A significant number of feminist women have been the target of various forms of harassment in response to posts they have made on social media, and Twitter in particular has been criticised in the past for not doing enough to aid victims of online abuse.

For organizations, such a backlash can cause overall brand damage, especially if reported by the media. However, this is not always the case, as any brand damage in the eyes of people with an opposing opinion to that presented by the organization could sometimes be outweighed by strengthening the brand in the eyes of others. Furthermore, if an organization or individual gives in to demands that others perceive as wrong-headed, that can then provoke a counter-backlash.

Some websites, such as Reddit, have rules forbidding the posting of personal information of individuals (also known as doxxing), due to concerns about such postings leading to mobs of large numbers of Internet users directing harassment at the specific individuals thereby identified. In particular, the Reddit rule forbidding the posting of personal information is widely understood to imply that all identifying photos and names must be censored in Facebook screenshots posted to Reddit. However, the interpretation of this rule in relation to public Twitter posts is less clear, and in any case like-minded people online have many other ways they can use to direct each other's attention to public social media posts they disagree with.

Children also face dangers online such as cyberbullying and approaches by sexual predators, who sometimes pose as children themselves. Children may also encounter material which they may find upsetting, or material which their parents consider to be not age-appropriate. Due to naivety, they may also post personal information about themselves online, which could put them or their families at risk, unless warned not to do so. Many parents choose to enable internet filtering, and/or supervise their children's online activities, in an attempt to protect their children from inappropriate material on the internet. The most popular social networking websites, such as Facebook and Twitter, commonly forbid users under the age of 13. However, these policies are typically trivial to circumvent by registering an account with a false birth date, and a significant number of children aged under 13 join such sites anyway. Social networking sites for younger children, which claim to provide better levels of protection for children, also exist.

The Internet has been a major outlet for leisure activity since its inception, with entertaining social experiments such as MUDs and MOOs being conducted on university servers, and humor-related Usenet groups receiving much traffic. Today, many Internet forums have sections devoted to games and funny videos. Over 6 million people use blogs or message boards as a means of communication and for the sharing of ideas. The Internet pornography and online gambling industries have taken advantage of the World Wide Web, and often provide a significant source of advertising revenue for other websites. Although many governments have attempted to restrict both industries' use of the Internet, in general this has failed to stop their widespread popularity.

Another area of leisure activity on the Internet is multiplayer gaming. This form of recreation creates communities, where people of all ages and origins enjoy the fast-paced world of multiplayer games. These range from MMORPG to first-person shooters, from role-playing video games to online gambling. While online gaming has been around since the 1970s, modern modes of online gaming began with subscription services such as GameSpy and MPlayer. Non-subscribers were limited to certain types of game play or certain games. Many people use the Internet to access and download music, movies and other works for their enjoyment and relaxation. Free and fee-based services exist for all of these activities, using centralized servers and distributed peer-to-peer technologies. Some of these sources exercise more care with respect to the original artists' copyrights than others.

Image created for a GameSpy contest.

Internet usage has been correlated to users' loneliness. Lonely people tend to use the Internet as an outlet for their feelings and to share their stories with others, such as in the "I am lonely will anyone speak to me" thread.

Cybersectarianism is a new organizational form which involves: "highly dispersed small groups of practitioners that may remain largely anonymous within the larger social context and operate in relative secrecy, while still linked remotely to a larger network of believers who share a set of practices and texts, and often a common devotion to a particular leader. Overseas supporters provide funding and support; domestic practitioners distribute tracts, participate in acts of resistance, and share information on the internal situation with outsiders. Collectively, members and practitioners of such sects construct viable virtual communities of faith, exchanging personal testimonies and engaging in collective study via email, on-line chat rooms and web-based message boards." In particular, the British government has raised concerns about the prospect of young British Muslims being indoctrinated into Islamic extremism by material on the Internet, being persuaded to join terrorist groups such as the so-called "Islamic State", and then potentially committing acts of terrorism on returning to Britain after fighting in Syria or Iraq.

Cyberslacking can become a drain on corporate resources; the average UK employee spent 57 minutes a day surfing the Web while at work, according to a 2003 study by Peninsula Business Services. Internet addiction disorder is excessive computer use that interferes with daily life. Psychologist Nicolas Carr believe that Internet use has other effects on individuals, for instance improving skills of scan-reading and interfering with the deep thinking that leads to true creativity.

Electronic business

Electronic business (*e-business*) encompasses business processes spanning the entire value chain: purchasing, supply chain management, marketing, sales, customer service, and business relationship. E-commerce seeks to add revenue streams using the Internet to build and enhance relationships with clients and partners.

According to International Data Corporation, the size of worldwide e-commerce, when global business-to-business and -consumer transactions are combined, equate to $16 trillion for 2013. A report by Oxford Economics adds those two together to estimate the total size of the digital economy at $20.4 trillion, equivalent to roughly 13.8% of global sales.

Drawbacks

While much has been written of the economic advantages of Internet-enabled commerce, there is also evidence that some aspects of the Internet such as maps and location-aware services may serve to reinforce economic inequality and the digital divide. Electronic commerce may be responsible for consolidation and the decline of mom-and-pop, brick and mortar businesses resulting in increases in income inequality.

Author Andrew Keen, a long-time critic of the social transformations caused by the Internet, has recently focused on the economic effects of consolidation from Internet businesses. Keen cites a 2013 Institute for Local Self-Reliance report saying brick-and-mortar retailers employ 47 people for every $10 million in sales, while Amazon employs only 14. Similarly, the 700-employee room rental start-up Airbnb was valued at $10 billion in 2014, about half as much as Hilton Hotels, which employs 152,000 people. And car-sharing Internet startup Uber employs 1,000 full-time employees and is valued at $18.2 billion, about the same valuation as Avis and Hertz combined, which together employ almost 60,000 people.

Telecommuting

Remote work is facilitated by tools such as groupware, virtual private networks, conference calling, videoconferencing, and Voice over IP (VOIP). It can be efficient and useful for companies as it allows workers to communicate over long distances, saving significant amounts of travel time and cost. As broadbandInternet connections become more commonplace, more and more workers have adequate bandwidth at home to use these tools to link their home to their corporate intranet and internal phone networks.

Crowdsourcing

Internet provides a particularly good venue for crowdsourcing (outsourcing tasks to a distributed group of people) since individuals tend to be more open in web-based projects where they are not being physically judged or scrutinized and thus can feel more comfortable sharing.

Crowdsourcing systems are used to accomplish a variety of tasks. For example, the crowd may be invited to develop a new technology, carry out a design task, refine or carry out the steps of an algorithm (see human-based computation), or help capture, systematize, or analyze large amounts of data (see also citizen science).

Wikis have also been used in the academic community for sharing and dissemination of information across institutional and international boundaries. In those settings, they have been found useful for collaboration on grant writing, strategic planning, departmental documentation, and committee work. The United States Patent and Trademark Office uses a wiki to allow the public to collaborate on finding prior art relevant to examination of pending patent applications. Queens, New York has used a wiki to allow citizens to collaborate on the design and planning of a local park.

The English Wikipedia has the largest user base among wikis on the World Wide Web and ranks in the top 10 among all Web sites in terms of traffic.

Politics and political revolutions

The Internet has achieved new relevance as a political tool. The presidential campaign of Howard Dean in 2004 in the United States was notable for its success in soliciting donation via the Internet. Many political groups use the Internet to achieve a new method of organizing for carrying out their mission, having given rise to Internet activism, most notably practiced by rebels in the Arab Spring.

The New York Times suggested that social media websites, such as Facebook and Twitter, helped people organize the political revolutions in Egypt, by helping activists organize protests, communicate grievances, and disseminate information.

The potential of the Internet as a civic tool of communicative power was explored by Simon R. B. Berdal in his 2004 thesis:

> As the globally evolving Internet provides ever new access points to virtual discourse forums, it also promotes new civic relations and associations within which communicative power may flow and accumulate. Thus, traditionally … national-embedded peripheries get entangled into greater, international peripheries, with stronger combined powers… The Internet, as a consequence, changes the topology of the “centre-periphery” model, by stimulating conventional peripheries to interlink into “super-periphery” structures, which enclose and “besiege” several centers at once.

Banner in Bangkok during the 2014 Thai coup d’état, informing the Thai public that ‘like’ or ‘share’ activities on social media could result in imprisonment (observed June 30, 2014).

Berdal, therefore, extends the Habermasian notion of the Public sphere to the Internet, and underlines the inherent global and civic nature that interwoven Internet technologies provide. To limit the growing civic potential of the Internet, Berdal also notes how “self-protective measures” are put in place by those threatened by it:

> If we consider China’s attempts to filter “unsuitable material” from the Internet, most of us would agree that this resembles a self-protective measure by the system against the growing civic potentials of the Internet. Nevertheless, both types represent limitations to “peripheral capacities”. Thus, the Chinese government tries to prevent communicative power to build up and unleash (as the 1989 Tiananmen Square uprising suggests, the government may find it wise to install “upstream measures”). Even though limited, the Internet is proving to be an empowering tool also to the Chinese periphery: Analysts believe that Internet petitions have influenced policy implementation in favor of the public’s online-articulated will …

Incidents of politically motivated Internet censorship have now been recorded in many countries, including western democracies.

Philanthropy

The spread of low-cost Internet access in developing countries has opened up new possibilities for peer-to-peer charities, which allow individuals to contribute small amounts to charitable projects for other individuals. Websites, such as DonorsChoose and GlobalGiving, allow small-scale donors to direct funds to individual projects of their choice.

A popular twist on Internet-based philanthropy is the use of peer-to-peer lending for charitable purposes. Kiva pioneered this concept in 2005, offering the first web-based service to publish individual loan profiles for funding. Kiva raises funds for local intermediary microfinance organizations which post stories and updates on behalf of the borrowers. Lenders can contribute as little as $25 to loans of their choice, and receive their money back as borrowers repay. Kiva falls short of being a pure peer-to-peer charity, in that loans are disbursed before being funded by lenders and borrowers do not communicate with lenders themselves.

However, the recent spread of low cost Internet access in developing countries has made genuine international person-to-person philanthropy increasingly feasible. In 2009 the US-based nonprofit Zidisha tapped into this trend to offer the first person-to-person microfinance platform to link lenders and borrowers across international borders without intermediaries. Members can fund loans for as little as a dollar, which the borrowers then use to develop business activities that improve their families' incomes while repaying loans to the members with interest. Borrowers access the Internet via public cybercafes, donated laptops in village schools, and even smart phones, then create their own profile pages through which they share photos and information about themselves and their businesses. As they repay their loans, borrowers continue to share updates and dialogue with lenders via their profile pages. This direct web-based connection allows members themselves to take on many of the communication and recording tasks traditionally performed by local organizations, bypassing geographic barriers and dramatically reducing the cost of microfinance services to the entrepreneurs.

Security

Many computer scientists describe the Internet as a "prime example of a large-scale, highly engineered, yet highly complex system". The structure was found to be highly robust to random failures, yet, very vulnerable to intentional attacks.

The Internet structure and its usage characteristics have been studied extensively and the possibility of developing alternative structures has been investigated.

Internet resources, hardware and software components, are the target of malicious attempts to gain unauthorized control to cause interruptions, or access private information. Such attempts include computer viruses which copy with the help of humans, computer worms which copy themselves automatically, denial of service attacks, ransomware, botnets, and spyware that reports on the activity and typing of users. Usually these activities constitute cybercrime. Defense theorists have also speculated about the possibilities of cyber warfare using similar methods on a large scale.

Surveillance

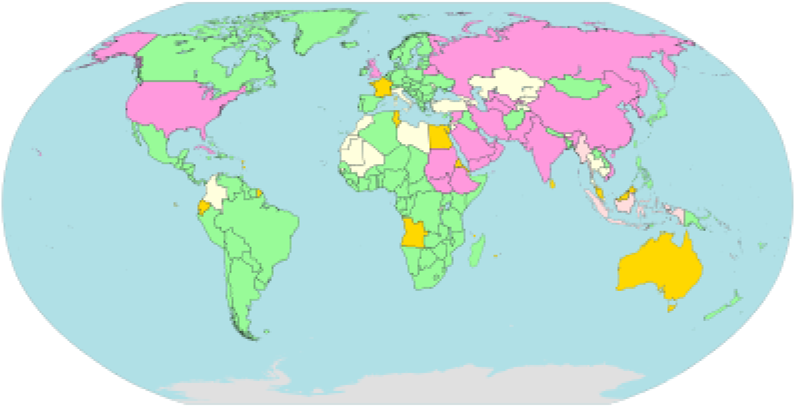
The vast majority of computer surveillance involves the monitoring of data and traffic on the Internet. In the United States for example, under the Communications Assistance For Law Enforcement Act, all phone calls and broadband Internet traffic (emails, web traffic, instant messaging, etc.) are required to be available for unimpeded real-time monitoring by Federal law enforcement agencies.

Packet capture (also sometimes referred to as "packet sniffing") is the monitoring of data traffic on a computer network. Computers communicate over the Internet by breaking up messages (emails, images, videos, web pages, files, etc.) into small chunks called "packets", which are routed through a network of computers, until they reach their destination, where they are assembled back into a complete "message" again. Packet Capture Appliance intercepts these packets as they are traveling through the network, in order to examine their contents using other programs. A packet capture is an information *gathering* tool, but not an *analysis* tool. That is it gathers "messages" but it does not analyze them and figure out what they mean. Other programs are needed to perform traffic analysis and sift through intercepted data looking for important/useful information. Under the Communications Assistance For Law Enforcement Act all U.S. telecommunications providers are required to install packet sniffing technology to allow Federal law enforcement and intelligence agencies to intercept all of their customers' broadband Internet and voice over Internet protocol (VoIP) traffic.

There is far too much data gathered by these packet sniffers for human investigators to manually search through all of it. So automated Internet surveillance computers sift through the vast amount of intercepted Internet traffic, and filter out and report to human investigators those bits of information which are "interesting"—such as the use of certain words or phrases, visiting certain types of web sites, or communicating via email or chat with a certain individual or group. Billions of dollars per year are spent, by agencies such as the Information Awareness Office, NSA, GCHQ and the FBI, to develop, purchase, implement, and operate systems which intercept and analyze all of this data, and extract only the information which is useful to law enforcement and intelligence agencies.

Similar systems are now operated by Iranian secret police to identify and suppress dissidents. All required hardware and software has been allegedly installed by German Siemens AG and Finnish Nokia.

Censorship



Internet censorship and surveillance by country

🟪	Pervasive	🟨	Changing situation
🟥	Substantial	🟩	Little or none
⬜	Selective	⬜	Not classified or no data

Some governments, such as those of Burma, Iran, North Korea, the Mainland China,Saudi Arabia and the United Arab Emirates restrict access to content on the Internet within their territories, especially to political and religious content, with domain name and keyword filters.

In Norway, Denmark, Finland, and Sweden, major Internet service providers have voluntarily agreed to restrict access to sites listed by authorities. While this list of forbidden resources is supposed to contain only known child pornography sites, the content of the list is secret. Many countries, including the United States, have enacted laws against the possession or distribution of certain material, such as child pornography, via the Internet, but do not mandate filter software. Many free or commercially available software programs, called content-control software are available to users to block offensive websites on individual computers or networks, in order to limit access by children to pornographic material or depiction of violence.

Performance

As the Internet is a heterogeneous network, the physical characteristics, including for example the data transfer rates of connections, vary widely. It exhibits emergent phenomena that depend on its large-scale organization.

Outages

An Internet blackout or outage can be caused by local signaling interruptions. Disruptions of submarine communications cables may cause blackouts or slowdowns to large areas, such as in the 2008 submarine cable disruption. Less-developed countries are more vulnerable due to a small number of high-capacity links. Land cables are also vulnerable, as in 2011 when a woman digging for scrap metal severed most connectivity for the nation of Armenia. Internet blackouts affecting almost entire countries can be achieved by governments as a form of Internet censorship, as in the blockage of the Internet in Egypt, whereby approximately 93% of networks were without access in 2011 in an attempt to stop mobilization for anti-government protests.

Energy use

In 2011 researchers estimated the energy used by the Internet to be between 170 and 307 GW, less than two percent of the energy used by humanity. This estimate included the energy needed to build, operate, and periodically replace the estimated 750 million laptops, a billion smart phones and 100 million servers worldwide as well as the energy that routers, cell towers, optical switches, Wi-Fi transmitters and cloud storage devices use when transmitting Internet traffic.

Licensing & Attributions

CC licensed content, Shared previously

- Internet. **Provided by:** Wikipedia. **Located at:** https://en.wikipedia.org/wiki/Internet#Social_impact. **License:** *CC BY-SA: Attribution-ShareAlike*
- Behold - The Beef Bush!. **Authored by:** Roxanne Ready. **Located at:** https://www.flickr.com/photos/shardsofblue/2654099615/. **License:** *CC BY-SA: Attribution-ShareAlike*

READING: THE WORLD WIDE WEB

Introduction

The World Wide Web (www, W3) is an information space where documents and other web resources are identified by URIs, interlinked by hypertext links, and can be accessed via the Internet. It has become known simply as *the Web*. Hypertext documents are commonly called *web pages*, which are primarily text documents formatted and annotated with the Hypertext Markup Language (HTML). Webpages may contain links to images, video, and software components that are rendered to users of a web browser application, running on the user's computer, as coherent pages of multimedia content. Embedded hyperlinks permit users to navigate between web pages. When multiple web pages are published with a common theme or within a common domain name, the collection is usually called a *web site*.

British computer scientist Tim Berners-Lee is the inventor of the Web. As a CERN employee, Berners-Lee distributed a proposal on 12 March 1989 for what would eventually become the World Wide Web. The initial proposal intended a more effective CERN communication system, but Berners-Lee also realized the concept could be implemented throughout the world. Berners-Lee and Belgian computer scientist Robert Cailliau proposed in 1990 to use hypertext "to link and access information of various kinds as a web of nodes in which the user can browse at will", and Berners-Lee finished the first website in December of that year. The first test was completed around 20 December 1990 and Berners-Lee reported about the project on the newsgroup *alt.hypertext* on 7 August 1991.

History

The NeXT Computer used by Tim Berners-Lee at CERN.

On March 12, 1989, Tim Berners-Lee issued a proposal to the management at CERN that referenced ENQUIRE, a database and software project he had built in 1980, and described a more elaborate information management system based on links embedded in readable text: “Imagine, then, the references in this document all being associated with the network address of the thing to which they referred, so that while reading this document you could skip to them with a click of the mouse.” Such a system, he explained, could be referred to using one of the existing meanings of the word *hypertext*, a term that he says was coined in the 1950s. There is no reason, the proposal continues, why such hypertext links could not encompass multimedia documents including graphics, speech and video, so that Berners-Lee goes on to propose the term *hypermedia*.

With help from Robert Cailliau, he published a more formal proposal (on 12 November 1990) to build a “Hypertext project” called “WorldWideWeb” (one word, also “W3”) as a “web” of “hypertext documents” to be viewed by “browsers” using a client–server architecture. This proposal estimated that a read-only web would be developed within three months and that it would take six months to achieve “the creation of new links and new material by readers, [so that] authorship becomes universal” as well as “the automatic notification of a reader when new material of interest to him/her has become available.” While the read-only goal was met, accessible authorship of web content took longer to mature, with the wiki concept, WebDAV, blogs, Web 2.0 and RSS/Atom.

The proposal was modeled after the SGML reader Dynatext by Electronic Book Technology, a spin-off from the Institute for Research in Information and Scholarship at Brown University. The Dynatext system, licensed by CERN, was a key player in the extension of SGML ISO 8879:1986 to Hypermedia within HyTime, but it was considered too expensive and had an inappropriate licensing policy for use in the general high energy physics community, namely a fee for each document and each document alteration.

The CERN data center in 2010 housing some WWW servers

A NeXT Computer was used by Berners-Lee as the world’s first web server and also to write the first web browser,WorldWideWeb, in 1990. By Christmas 1990, Berners-Lee had built all the tools necessary for a working Web: the first web browser (which was a web editor as well); the first web server; and the first web pages, which described the project itself.

The first web page may be lost, but Paul Jones of UNC-Chapel Hill in North Carolina announced in May 2013 that Berners-Lee gave him what he says is the oldest known web page during a 1991 visit to UNC. Jones stored it on amagneto-optical drive and on his NeXT computer.

On 6 August 1991, Berners-Lee published a short summary of the World Wide Web project on the newsgroup *alt.hypertext*. This date also marked the debut of the Web as a publicly available service on the Internet, although new users only accessed it after 23 August. For this reason this is considered the internaut’s day. Several news media have reported that the first photo on the Web was published by Berners-Lee in 1992, an image of the CERN house band Les Horribles Cernettes taken by Silvano de Gennaro; Gennaro has disclaimed this story, writing that media were “totally distorting our words for the sake of cheap sensationalism.”

The first server outside Europe was installed at the Stanford Linear Accelerator Center (SLAC) in Palo Alto, California, to host the SPIRES-HEP database. Accounts differ substantially as to the date of this event. The World Wide Web Consortium says December 1992, whereas SLAC itself claims 1991. This is supported by a W3C document titled *A Little History of the World Wide Web*.

The underlying concept of hypertext originated in previous projects from the 1960s, such as the Hypertext Editing System (HES) at Brown University, Ted Nelson’s Project Xanadu, and Douglas Engelbart’s oN-Line System

(NLS). Both Nelson and Engelbart were in turn inspired by Vannevar Bush's microfilm-based *memex*, which was described in the 1945 essay "As We May Think".

Berners-Lee's breakthrough was to marry hypertext to the Internet. In his book *Weaving The Web*, he explains that he had repeatedly suggested that a marriage between the two technologies was possible to members of *both* technical communities, but when no one took up his invitation, he finally assumed the project himself. In the process, he developed three essential technologies:

- a system of globally unique identifiers for resources on the Web and elsewhere, the universal document identifier (UDI), later known as uniform resource locator (URL) and uniform resource identifier (URI);
- the publishing language HyperText Markup Language (HTML);
- the Hypertext Transfer Protocol (HTTP).

The World Wide Web had a number of differences from other hypertext systems available at the time. The Web required only unidirectional links rather than bidirectional ones, making it possible for someone to link to another resource without action by the owner of that resource. It also significantly reduced the difficulty of implementing web servers and browsers (in comparison to earlier systems), but in turn presented the chronic problem of *link rot*. Unlike predecessors such as HyperCard, the World Wide Web was non-proprietary, making it possible to develop servers and clients independently and to add extensions without licensing restrictions. On 30 April 1993, CERN announced that the World Wide Web would be free to anyone, with no fees due. Coming two months after the announcement that the server implementation of the Gopher protocol was no longer free to use, this produced a rapid shift away from Gopher and towards the Web. An early popular web browser was ViolaWWW for Unix and the X Windowing System.

Scholars generally agree that a turning point for the World Wide Web began with the introduction of the Mosaic web browser in 1993, a graphical browser developed by a team at the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign (NCSA-UIUC), led by Marc Andreessen. Funding for Mosaic came from the U.S. *High-Performance Computing and Communications Initiative* and the *High Performance Computing and Communication Act of 1991*, one of several computing developments initiated by U.S. Senator Al Gore. Prior to the release of Mosaic, graphics were not commonly mixed with text in web pages and the web's popularity was less than older protocols in use over the Internet, such as Gopher and Wide Area Information Servers (WAIS). Mosaic's graphical user interface allowed the Web to become, by far, the most popular Internet protocol.

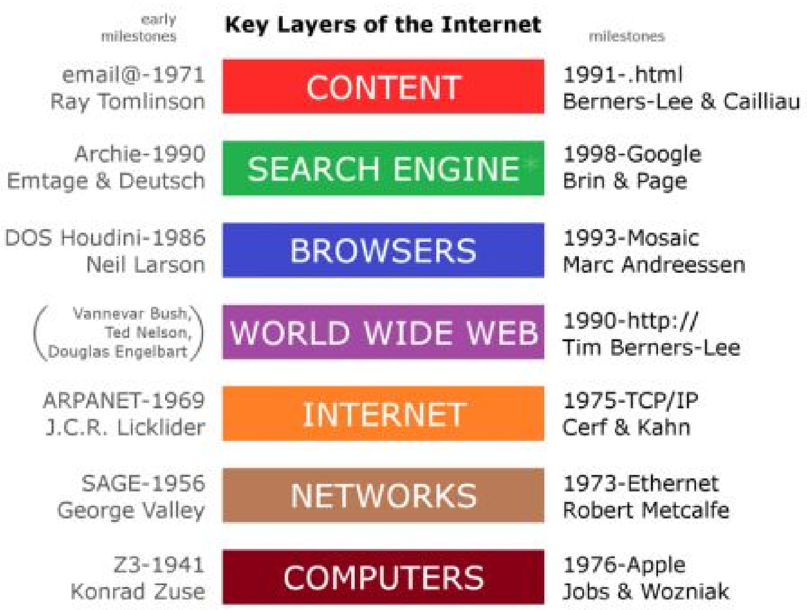
Robert Cailliau, Jean-François Abramatic of IBM, and Tim Berners-Lee at the 10th anniversary of the World Wide Web Consortium.

The World Wide Web Consortium (W3C) was founded by Tim Berners-Lee after he left the European Organization for Nuclear Research (CERN) in October 1994. It was founded at the Massachusetts Institute of Technology Laboratory for Computer Science (MIT/LCS) with support from the Defense Advanced Research Projects Agency (DARPA), which had pioneered the Internet; a year later, a second site was founded at INRIA (a French national computer research lab) with support from the European Commission DG InfSo; and in 1996, a third continental site was created in Japan at Keio University. By the end of 1994, the total number of websites was still relatively small, but many notable websites were already active that foreshadowed or inspired today's most popular services.

Connected by the existing Internet, other websites were created around the world, adding international standards for domain names and HTML. Since then, Berners-Lee has played an active role in guiding the development of web standards (such as the markup languages to compose web pages in), and has advocated his vision of a Semantic Web. The World Wide Web enabled the spread of information over the Internet through an easy-to-use and flexible format. It thus played an important role in popularizing use of the Internet. Although the two terms are sometimes conflated in popular use, *World Wide Web* is not synonymous with *Internet*. The Web is an information space containing hyperlinked documents and other resources, identified by their URIs. It is implemented as both client and server software using Internet protocols such as TCP/IP and HTTP.

Tim Berners-Lee was knighted in 2004 by Queen Elizabeth II for “services to the global development of the Internet”.

Function



The World Wide Web functions as a layer on top of the Internet, helping to make it more functional. The advent of the Mosaic web browser helped to make the web much more usable.

The terms *Internet* and *World Wide Web* are often used without much distinction. However, the two things are not the same. The Internet is a global system of interconnected computer networks. In contrast, the World Wide Web is one of the services transferred over these networks. It is a collection of text documents and other resources, linked by hyperlinks and URLs, usually accessed by web browsers, from web servers.

Viewing a web page on the World Wide Web normally begins either by typing the URL of the page into a web browser, or by following a hyperlink to that page or resource. The web browser then initiates a series of background communication messages to fetch and display the requested page. In the 1990s, using a browser to view web pages—and to move from one web page to another through hyperlinks—came to be known as ‘browsing,’ ‘web surfing,’ (after channel surfing), or ‘navigating the Web’. Early studies of this new behavior investigated user patterns in using web browsers. One study, for example, found five user patterns: exploratory surfing, window surfing, evolved surfing, bounded navigation and targeted navigation.

The following example demonstrates the functioning of a web browser when accessing a page at the URL http://example.org/wiki/World_Wide_Web. The browser resolves the server name of the URL (*example.org*) into an Internet Protocol address using the globally distributed Domain Name System (DNS). This lookup returns an IP address such as *203.0.113.4*. The browser then requests the resource by sending an HTTP request across the Internet to the computer at that address. It requests service from a specific TCP port number that is well known for the HTTP service, so that the receiving host can distinguish an HTTP request from other network protocols it may be servicing. The HTTP protocol normally uses port number 80. The content of the HTTP request can be as simple as two lines of text:

```
GET /wiki/World_Wide_Web HTTP/1.1

Host: example.org
```

The computer receiving the HTTP request delivers it to web server software listening for requests on port 80. If the web server can fulfill the request it sends an HTTP response back to the browser indicating success:

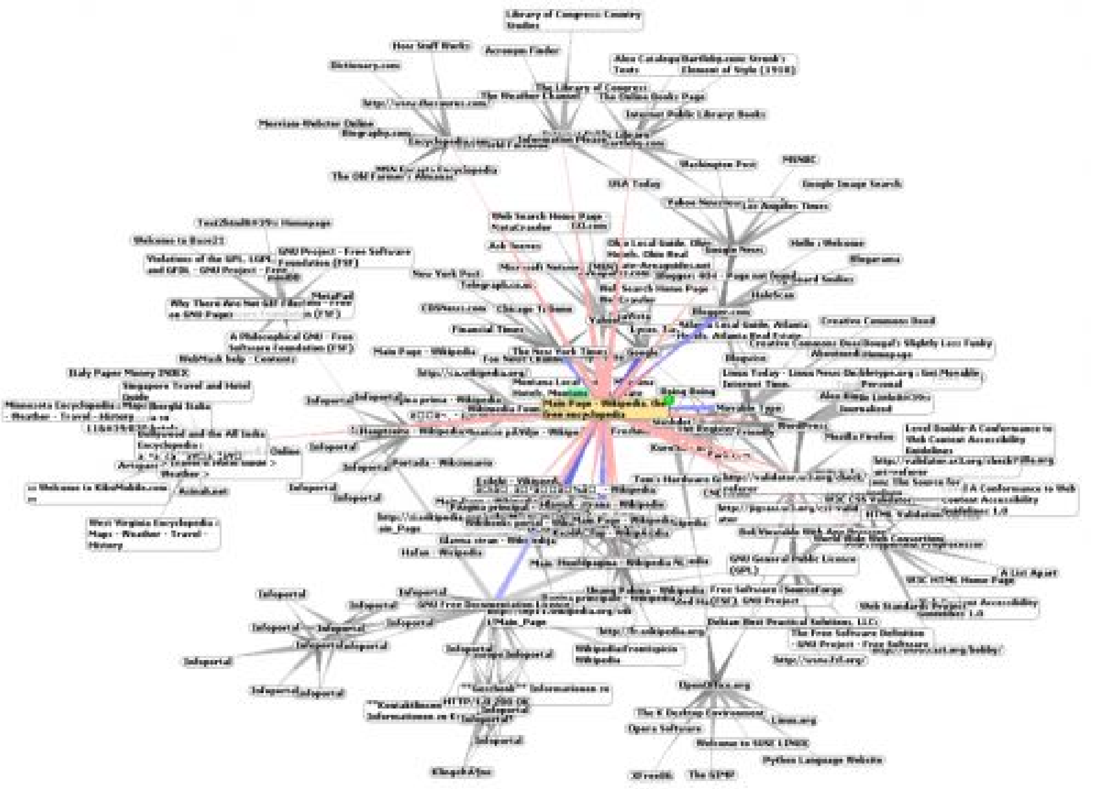
```
HTTP/1.0 200 OK

Content-Type: text/html; charset=UTF-8
```

followed by the content of the requested page. The Hypertext Markup Language for a basic web page looks like <html> <head> <title>Example.org – The World Wide Web</title> </head> <body> <p>The World Wide Web, abbreviated as WWW and commonly known …</p> </body> </html>

The web browser parses the HTML and interprets the markup (<title>, <p> for paragraph, and such) that surrounds the words to format the text on the screen. Many web pages use HTML to reference the URLs of other resources such as images, other embedded media, scripts that affect page behavior, and Cascading Style Sheets that affect page layout. The browser makes additional HTTP requests to the web server for these other Internet media types. As it receives their content from the web server, the browser progressively renders the page onto the screen as specified by its HTML and these additional resources.

Linking

Most web pages contain hyperlinks to other related pages and perhaps to downloadable files, source documents, definitions and other web resources. In the underlying HTML, a hyperlink looks like <a href="*http://example.org/wiki/Main_Page*">*Example.org, a free encyclopedia*</a>



Graphic representation of a minute fraction of the WWW, demonstrating hyperlinks.

Such a collection of useful, related resources, interconnected via hypertext links is dubbed a *web* of information. Publication on the Internet created what Tim Berners-Lee first called the *WorldWideWeb* (in its original CamelCase, which was subsequently discarded) in November 1990.

The hyperlink structure of the WWW is described by the webgraph: the nodes of the webgraph correspond to the web pages (or URLs) the directed edges between them to the hyperlinks.

Over time, many web resources pointed to by hyperlinks disappear, relocate, or are replaced with different content. This makes hyperlinks obsolete, a phenomenon referred to in some circles as link rot, and the hyperlinks affected by it are often called dead links. The ephemeral nature of the Web has prompted many efforts to archive web sites. The Internet Archive, active since 1996, is the best known of such efforts.

Dynamic updates of web pages

JavaScript is a scripting language that was initially developed in 1995 by Brendan Eich, then of Netscape, for use within web pages. The standardised version is ECMAScript. To make web pages more interactive, some web applications also use JavaScript techniques such as Ajax (asynchronous JavaScript and XML). Client-side script is delivered with the page that can make additional HTTP requests to the server, either in response to user actions such as mouse movements or clicks, or based on elapsed time. The server's responses are used to modify the current page rather than creating a new page with each response, so the server needs only to provide limited, incremental information. Multiple Ajax requests can be handled at the same time, and users can interact with the page while data is retrieved. Web pages may also regularly poll the server to check whether new information is available.

WWW prefix

Many hostnames used for the World Wide Web begin with *www* because of the long-standing practice of naming Internet hosts according to the services they provide. The hostname of a web server is often *www*, in the same way that it may be *ftp* for an FTP server, and *news* or *nntp* for a USENET news server. These host names appear as Domain Name System (DNS) or subdomain names, as in *www.example.com*. The use of *www* is not required by any technical or policy standard and many web sites do not use it; indeed, the first ever web server was called *nxoc01.cern.ch*. According to Paolo Palazzi, who worked at CERN along with Tim Berners-Lee, the popular use of *www* as subdomain was accidental; the World Wide Web project page was intended to be published at www.cern.ch while info.cern.ch was intended to be the CERN home page, however the DNS records were never switched, and the practice of prepending *www* to an institution's website domain name was subsequently copied. Many established websites still use the prefix, or they employ other subdomain names such as *www2*, *secure* or *en* for special purposes. Many such web servers are set up so that both the main domain name (e.g., example.com) and the *www* subdomain (e.g., www.example.com) refer to the same site; others require one form or the other, or they may map to different web sites.

The use of a subdomain name is useful for load balancing incoming web traffic by creating a CNAME record that points to a cluster of web servers. Since, currently, only a subdomain can be used in a CNAME, the same result cannot be achieved by using the bare domain root.

When a user submits an incomplete domain name to a web browser in its address bar input field, some web browsers automatically try adding the prefix "www" to the beginning of it and possibly ".com", ".org" and ".net" at the end, depending on what might be missing. For example, entering 'microsoft' may be transformed to *http://www.microsoft.com/* and 'openoffice' to *http://www.openoffice.org*. This feature started appearing in early versions of Mozilla Firefox, when it still had the working title 'Firebird' in early 2003, from an earlier practice in browsers such as Lynx. It is reported that Microsoft was granted a US patent for the same idea in 2008, but only for mobile devices.

In English, *www* is usually read as *double-u double-u double-u*. Some users pronounce it *dub-dub-dub*, particularly in New Zealand. Stephen Fry, in his "Podgrammes" series of podcasts, pronounces it *wuh wuh wuh.* The English writer Douglas Adams once quipped in The Independent on Sunday(1999): "The World Wide Web is the only thing I know of whose shortened form takes three times longer to say than what it's short for". In Mandarin Chinese, *World Wide Web* is commonly translated via a phono-semantic matching to *wàn wéi wǎng* (万维网), which satisfies *www* and literally means "myriad dimensional net", a translation that reflects the design concept and proliferation of the World Wide Web. Tim Berners-Lee's web-space states that *World Wide Web* is officially spelled as three separate words, each capitalised, with no intervening hyphens.

Use of the www prefix is declining as Web 2.0 web applications seek to brand their domain names and make them easily pronounceable. As the mobile web grows in popularity, services like Gmail.com, MySpace.com, Facebook.com and Twitter.com are most often mentioned without adding "www." (or, indeed, ".com") to the domain.

Scheme specifiers

The scheme specifiers *http://* and *https://* at the start of a web URI refer to Hypertext Transfer Protocol or HTTP Secure, respectively. They specify the communication protocol to use for the request and response. The HTTP protocol is fundamental to the operation of the World Wide Web, and the added encryption layer in HTTPS is essential when browsers send or retrieve confidential data, such as passwords or banking information. Web browsers usually automatically prepend http:// to user-entered URIs, if omitted.

Web security

For criminals, the web has become the preferred way to spread malware. Cybercrime on the web can include identity theft, fraud, espionage and intelligence gathering. Web-based vulnerabilities now outnumber traditional computer security concerns, and as measured by Google, about one in ten web pages may contain malicious code. Most web-based attacks take place on legitimate websites, and most, as measured by Sophos, are hosted in the United States, China and Russia. The most common of all malware threats is SQL injection attacks against websites. Through HTML and URIs, the Web was vulnerable to attacks like cross-site scripting (XSS) that came with the introduction of JavaScript and were exacerbated to some degree by Web 2.0 and Ajax web design that favors the use of scripts.Today by one estimate, 70% of all websites are open to XSS attacks on their users. Phishing is another common threat to the Web. “SA, the Security Division of EMC, today announced the findings of its January 2013 Fraud Report, estimating the global losses from phishing at $1.5 Billion in 2012.” Two of the well-known phishing methods are Covert Redirect and Open Redirect.

Proposed solutions vary to extremes. Large security vendors like McAfee already design governance and compliance suites to meet post-9/11 regulations, and some, like Finjan have recommended active real-time inspection of code and all content regardless of its source. Some have argued that for enterprise to see security as a business opportunity rather than a cost center, “ubiquitous, always-on digital rights management” enforced in the infrastructure by a handful of organizations must replace the hundreds of companies that today secure data and networks. Jonathan Zittrain has said users sharing responsibility for computing safety is far preferable to locking down the Internet.

Privacy

Every time a client requests a web page, the server can identify the request’s IP address and usually logs it. Also, unless set not to do so, most web browsers record requested web pages in a viewable *history* feature, and usually cache much of the content locally. Unless the server-browser communication uses HTTPS encryption, web requests and responses travel in plain text across the internet and can be viewed, recorded, and cached by intermediate systems.

When a web page asks for, and the user supplies, personally identifiable information—such as their real name, address, e-mail address, etc.—web-based entities can associate current web traffic with that individual. If the website uses HTTP cookies, username and password authentication, or other tracking techniques, it can relate other web visits, before and after, to the identifiable information provided. In this way it is possible for a web-based organisation to develop and build a profile of the individual people who use its site or sites. It may be able to build a record for an individual that includes information about their leisure activities, their shopping interests, their profession, and other aspects of their demographic profile. These profiles are obviously of potential interest to marketeers, advertisers and others. Depending on the website’s terms and conditions and the local laws that apply information from these profiles may be sold, shared, or passed to other organisations without the user being informed. For many ordinary people, this means little more than some unexpected e-mails in their in-box, or some uncannily relevant advertising on a future web page. For others, it can mean that time spent indulging an unusual interest can result in a deluge of further targeted marketing that may be unwelcome. Law enforcement, counter terrorism and espionage agencies can also identify, target and track individuals based on their interests or proclivities on the Web.

Social networking sites try to get users to use their real names, interests, and locations. They believe this makes the social networking experience more realistic, and therefore more engaging for all their users. On the other hand, uploaded photographs or unguarded statements can be identified to an individual, who may regret this exposure. Employers, schools, parents, and other relatives may be influenced by aspects of social networking profiles that the posting individual did not intend for these audiences. On-line bullies may make use of personal information to harass or stalk users. Modern social networking websites allow fine grained control of the privacy settings for each individual posting, but these can be complex and not easy to find or use, especially for beginners.

Photographs and videos posted onto websites have caused particular problems, as they can add a person's face to an on-line profile. With modern and potential facial recognition technology, it may then be possible to relate that face with other, previously anonymous, images, events and scenarios that have been imaged elsewhere. Because of image caching, mirroring and copying, it is difficult to remove an image from the World Wide Web.

Standards

Many formal standards and other technical specifications and software define the operation of different aspects of the World Wide Web, the Internet, and computer information exchange. Many of the documents are the work of the World Wide Web Consortium (W3C), headed by Berners-Lee, but some are produced by the Internet Engineering Task Force (IETF) and other organizations.

Usually, when web standards are discussed, the following publications are seen as foundational:

- Recommendations for markup languages, especially HTML and XHTML, from the W3C. These define the structure and interpretation of hypertext documents.
- Recommendations for stylesheets, especially CSS, from the W3C.
- Standards for ECMAScript (usually in the form of JavaScript), from Ecma International.
- Recommendations for the Document Object Model, from W3C.

Additional publications provide definitions of other essential technologies for the World Wide Web, including, but not limited to, the following:

- *Uniform Resource Identifier* (URI), which is a universal system for referencing resources on the Internet, such as hypertext documents and images. URIs, often called URLs, are defined by the IETF's RFC 3986 / STD 66: *Uniform Resource Identifier (URI): Generic Syntax*, as well as its predecessors and numerous URI scheme-defining RFCs;
- *HyperText Transfer Protocol (HTTP)*, especially as defined by RFC 2616: *HTTP/1.1* and RFC 2617: *HTTP Authentication*, which specify how the browser and server authenticate each other.

Accessibility

There are methods for accessing the Web in alternative mediums and formats to facilitate use by individuals with disabilities. These disabilities may be visual, auditory, physical, speech related, cognitive, neurological, or some combination. Accessibility features also help people with temporary disabilities, like a broken arm, or aging users as their abilities change. The Web receives information as well as providing information and interacting with society. The World Wide Web Consortium claims it essential that the Web be accessible, so it can provide equal access and equal opportunity to people with disabilities. Tim Berners-Lee once noted, "The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect." Many countries regulate web accessibility as a requirement for websites. International cooperation in the W3C Web Accessibility Initiative led to simple guidelines that web content authors as well as software developers can use to make the Web accessible to persons who may or may not be using assistive technology.

Internationalization

The W3C Internationalization Activity assures that web technology works in all languages, scripts, and cultures. Beginning in 2004 or 2005, Unicode gained ground and eventually in December 2007 surpassed both ASCII and Western European as the Web's most frequently used character encoding. OriginallyRFC 3986 allowed resources to be identified by URI in a subset of US-ASCII. RFC 3987 allows more characters—any character in the Universal Character Set—and now a resource can be identified by IRI in any language.

Statistics

Between 2005 and 2010, the number of web users doubled, and was expected to surpass two billion in 2010. Early studies in 1998 and 1999 estimating the size of the Web using capture/recapture methods showed that much of the web was not indexed by search engines and the Web was much larger than expected. According to a 2001 study, there was a massive number, over 550 billion, of documents on the Web, mostly in the invisible Web, or Deep Web. A 2002 survey of 2,024 million web pages determined that by far the most web content was in the English language: 56.4%; next were pages in German (7.7%), French (5.6%), and Japanese (4.9%). A more recent study, which used web searches in 75 different languages to sample the Web, determined that there were over 11.5 billion web pages in the publicly indexable web as of the end of January 2005. As of March 2009, the indexable web contains at least 25.21 billion pages. On 25 July 2008, Google software engineers Jesse Alpert and Nissan Hajaj announced that Google Search had discovered one trillion unique URLs. As of May 2009, over 109.5 million domains operated. Of these, 74% were commercial or other domains operating in the generic top-level domain *com*.

Statistics measuring a website's popularity are usually based either on the number of page views or on associated server 'hits' (file requests) that it receives.

Speed issues

Frustration over congestion issues in the Internet infrastructure and the high latency that results in slow browsing has led to a pejorative name for the World Wide Web: the *World Wide Wait*. Speeding up the Internet is an ongoing discussion over the use of peering and QoS technologies. Other solutions to reduce the congestion can be found at W3C. Guidelines for web response times are:

- 0.1 second (one tenth of a second). Ideal response time. The user does not sense any interruption.
- 1 second. Highest acceptable response time. Download times above 1 second interrupt the user experience.
- 10 seconds. Unacceptable response time. The user experience is interrupted and the user is likely to leave the site or system.

Web caching

A web cache is a server computer located either on the public Internet, or within an enterprise that stores recently accessed web pages to improve response time for users when the same content is requested within a certain time after the original request.

Most web browsers also implement a browser cache for recently obtained data, usually on the local disk drive. HTTP requests by a browser may ask only for data that has changed since the last access. Web pages and resources may contain expiration information to control caching to secure sensitive data, such as in online banking, or to facilitate frequently updated sites, such as news media. Even sites with highly dynamic content may permit basic resources to be refreshed only occasionally. Web site designers find it worthwhile to collate resources such as CSS data and JavaScript into a few site-wide files so that they can be cached efficiently.

Enterprise firewalls often cache Web resources requested by one user for the benefit of many. Some search engines store cached content of frequently accessed websites.

Licensing & Attributions

CC licensed content, Shared previously

- World Wide Web. **Provided by:** Wikipedia. **Located at:** https://en.wikipedia.org/wiki/World_Wide_Web#See_also. **License:** *CC BY-SA: Attribution-ShareAlike*
- Taking a selfie. **Authored by:** Susanne Nilsson. **Located at:** https://www.flickr.com/photos/infomastern/14162668916/. **License:** *CC BY-SA: Attribution-ShareAlike*
- World Wide Web. **Authored by:** sandra_schoen. **Located at:** https://pixabay.com/en/world-wide-web-www-lettering-world-341418/. **License:** *CC0: No Rights Reserved*

READING: UNIFORM RESOURCE LOCATOR

Introduction

A **uniform resource locator (URL)** is a reference to a resource that specifies the location of the resource on a computer network and a mechanism for retrieving it. A URL is a specific type of uniform resource identifier (URI), although many people use the two terms interchangeably. A URL implies the means to access an indicated resource, which is not true of every URI. URLs occur most commonly to reference web pages (http), but are also used for file transfer (ftp), email (mailto), database access (JDBC), and many other applications.

Most web browsers display the URL of a web page above the page in an address bar. A typical URL has the form *http://www.example.com/index.html*, which indicates the protocol type (*http*), the domain name, (*www.example.com*), and the specific web page (*index.html*).

History

The Uniform Resource Locator was standardized in 1994 by Tim Berners-Lee and the URI working group of the Internet Engineering Task Force (IETF) as an outcome of collaboration started at the IETF Living Documents “Birds of a Feather” session in 1992. The format combines the pre-existing system of domain names (created in 1985) with file path syntax, where slashes are used to separate directory and file names. Conventions already existed where server names could be prepended to complete file paths, preceded by a double-slash (//).

Berners-Lee later regretted the use of dots to separate the parts of the domain name within URIs, wishing he had used slashes throughout. For example,*http://www.example.com/path/to/name* would have been written *http:com/example/www/path/to/name*. Berners-Lee has also said that, given the colon following the URI scheme, the two slashes before the domain name were also unnecessary.

Syntax

Every HTTP URL consists of the following, in the given order. Several schemes other than HTTP also share this general format, with some variation.

- the scheme name (commonly called protocol, although not every URL scheme is a protocol, e.g. mailto is not a protocol)
- a colon, two slashes,
- a host, normally given as a domain name For example, *http://www.example.com/path/to/name* would have been written *http:com/example/www/path/to/name* but sometimes as a literal IP address
- optionally a colon followed by a port number
- the full path of the resource

The scheme says *how* to connect, the host specifies *where* to connect, and the remainder specifies *what* to ask for.

For programs such as Common Gateway Interface (CGI) scripts, this is followed by a query string, and an optional fragment identifier.

The syntax is:
scheme://[user:password@]domain:port/path?query_string#fragment_id

Component details:

- The scheme, which in many cases is the name of a protocol (but not always), defines how the resource will be obtained. Examples include http, https, ftp, file and many others. Although schemes are case-insensitive, the canonical form is lowercase.
- The domain name or literal numeric IP address gives the destination location for the URL. A literal numeric IPv6 address may be given, but must be enclosed in *[ ]* e.g.*[db8:0cec::99:123a]*.
  The domain *google.com*, or its numeric IP address *173.194.34.5*, is the address of Google's website.

- The domain name portion of a URL is not case sensitive since DNS ignores case:
  http://en.example.org/ and *HTTP://EN.EXAMPLE.ORG/* both open the same page.

- The port number, given in decimal, is optional; if omitted, the default for the scheme is used.
  For example, *http://vnc.example.com:5800* connects to port 5800 of vnc.example.com, which may be appropriate for a VNC remote control session. If the port number is omitted for an http: URL, the browser will connect on port 80, the default HTTP port. The default port for an https: request is 443.

- The path is used to specify and perhaps find the resource requested. This path may or may not describe folders on the file system in the web server. It may be very different from the arrangement of folders on the web server. It is case-sensitive, though it may be treated as case-insensitive by some servers, especially those based on Microsoft Windows.
  If the server is case sensitive and *http://en.example.org/wiki/URL* is correct, then *http://en.example.org/WIKI/URL* or *http://en.example.org/wiki/url* will display an HTTP 404 error page, unless these URLs point to valid resources themselves.

- The query string contains data to be passed to software running on the server. It may contain name/value pairs separated by ampersands, for example
  ?first_name=John&last_name=Doe.

- The fragment identifier, if present, specifies a part or a position within the overall resource or document.
  When used with HTML, it usually specifies a section or location within the page, and used in combination with Anchor elements or the "id" attribute of an element, the browser is scrolled to display that part of the page.

The scheme name defines the namespace, purpose, and the syntax of the remaining part of the URL. Software will try to process a URL according to its scheme and context. For example, a web browser will usually dereference the URL *http://example.org:80* by performing an HTTP request to the host at *example.org*, using port number 80.

Other examples of scheme names include https, gopher, wais, ftp. URLs with https as a scheme (such as *https://example.com/*) require that requests and responses will be made over a secure connection to the website. Some schemes that require authentication allow a username, and perhaps a password too, to be embedded in the URL, for example*ftp://asmith@ftp.example.org*. Passwords embedded in this way are not conducive to security, but the full possible syntax is
scheme://username:password@domain:port/path?query_string#fragment_id

Other schemes do not follow the HTTP pattern. For example, the *mailto* scheme only uses valid email addresses. When clicked on in an application, the URL *mailto:bob@example.com*may start an e-mail composer with the address *bob@example.com* in the To field. The *tel* scheme is even more different; it uses the public switched telephone network for addressing, instead of domain names representing Internet hosts.

List of allowed URL characters

Unreserved

The alphanumerical upper and lower case character may optionally be encoded:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

0 1 2 3 4 5 6 7 8 9 – _ . ~

Reserved

Special symbols must sometimes be percent-encoded:

! * ' ( ) ; : @ & = + $, / ? % # [ ]

Further details can for example be found in RFC 3986 and http://www.w3.org/Addressing/URL/uri-spec.html.

Relationship to URI

A URL is a URI that, in addition to identifying a web resource, provides a means of locating the resource by describing its "primary access mechanism (e.g., its network location)".

Internet hostnames

A hostname is a domain name assigned to a host computer. This is usually a combination of the host's local name with its parent domain's name. For example, en.example.org consists of a local hostname (*en*) and the domain name *example.org*. The hostname is translated into an IP address via the local hosts file, or the domain name system (DNS) resolver. It is possible for a single host computer to have several hostnames; but generally the operating system of the host prefers to have one hostname that the host uses for itself.

Any domain name can also be a hostname, as long as the restrictions mentioned below are followed. For example, both "en.example.org" and "example.org" can be hostnames if they both have IP addresses assigned to them. The domain name "xyz.example.org" may not be a hostname if it does not have an IP address, but "aa.xyz.example.org" may still be a hostname. All hostnames are domain names, but not all domain names are hostnames.

URL protocols

The protocol, or scheme, of a URL defines how the resource will be obtained. Two common protocols on the web are HTTP and HTTPS. For various reasons, many sites have been switching to permitting access through both the HTTP and HTTPS protocols. Each protocol has advantages and disadvantages, including for some of the users that one or the other protocol either does not function, or is very undesirable. When a link contains a protocol specifier it results in the browser following the link using the specified protocol regardless of the potential desires of the user.

Protocol-relative URLs

It is possible to construct valid URLs without specifying a protocol which are called protocol-relative links (PRL) or protocol-relative URLs. Using PRLs on a page permits the viewer of the page to visit new pages using whichever protocol was used to obtain the page containing the link. This supports continuing to use whichever protocol the viewer has chosen to use for obtaining the current page when accessing new pages.

An example of a PRL is //en.wikipedia.org/wiki/Main_Page which is created by removing the protocol prefix.

Internationalized URL

Internet users are distributed throughout the world using a wide variety of languages and alphabets. Users expect to be able to create URLs in their own local alphabets.

An internationalized resource identifier (IRI) is a form of URL that includes Unicode characters. All modern browsers support IRIs. The parts of the URL requiring special treatment for different alphabets are the domain name and path.

The domain name in the IRI is known as an internationalized domain name (IDN). Web and Internet software automatically convert the domain name into punycode usable by the Domain Name System.

For example, the Chinese web site http://見.香港 becomes the following for DNS lookup. *xn--* indicates the character was not originally ASCII.
    http://xn--nw2a.xn--j6w193g/

The URL path name can also be specified by the user in the local alphabet. If not already encoded, it is converted to Unicode, and any characters not part of the basic URL character set are converted to English letters using percent-encoding.

For example, the following Japanese Web page http://domainname/引き割り.html becomes http://domainname/%E5%BC%95%E3%81%8D%E5%89%B2%E3%82%8A.html. The target computer decodes the address and displays the page.

Licensing & Attributions

CC licensed content, Shared previously

- Uniform resource locator. **Provided by:** Wikipedia. **Located at:** https://en.wikipedia.org/wiki/Uniform_resource_locator. **License:** *CC BY-SA: Attribution-ShareAlike*
- URL. **Authored by:** Sam Azgor. **Located at:** https://www.flickr.com/photos/samazgor/9104870907/. **License:** *CC BY: Attribution*

READING: WEB BROWSER

Introduction

A **web browser** (commonly referred to as a **browser**) is a software application for retrieving, presenting and traversing information resources on the World Wide Web. An *information resource* is identified by a Uniform Resource Identifier (URI/URL) and may be a web page, image, video or other piece of content. Hyperlinks present in resources enable users easily to navigate their browsers to related resources.

Although browsers are primarily intended to use the World Wide Web, they can also be used to access information provided by web servers in private networks or files in file systems.

The major web browsers are Firefox, Internet Explorer, Google Chrome, Opera, and Safari.

History

The first web browser was invented in 1990 by Sir Tim Berners-Lee. Berners-Lee is the director of the World Wide Web Consortium (W3C), which oversees the Web's continued development, and is also the founder of the World Wide Web Foundation. His browser was called WorldWideWeb and later renamed Nexus.

The first commonly available web browser with a graphical user interface was Erwise. The development of Erwise was initiated by Robert Cailliau.

In 1993, browser software was further innovated by Marc Andreessen with the release of Mosaic, "the world's first popular browser", which made the World Wide Web system easy to use and more accessible to the average person. Andreesen's browser sparked the internet boom of the 1990s. The introduction of Mosaic in 1993 – one of the first graphical web browsers – led to an explosion in web use. Andreessen, the leader of the Mosaic team at National Center for Supercomputing Applications (NCSA), soon started his own company, named Netscape, and released the Mosaic-influenced Netscape Navigator in 1994, which quickly became the world's most popular browser, accounting for 90% of all web use at its peak (see usage share of web browsers).

Marc Andreessen, inventor of Netscape

Microsoft responded with its Internet Explorer in 1995, also heavily influenced by Mosaic, initiating the industry's first browser war. Bundled with Windows, Internet Explorer gained dominance in the web browser market; Internet Explorer usage share peaked at over 95% by 2002.
Opera debuted in 1996; it has never achieved widespread use, having less than 2% browser usage share as of February 2012 according to Net Applications. Its Opera-mini version has an additive share, in April 2011 amounting to 1.1% of overall browser use, but focused on the fast-growing mobile phone web browser market, being preinstalled on over 40 million phones. It is also available on several other embedded systems, including Nintendo's Wii video game console.

In 1998, Netscape launched what was to become the Mozilla Foundation in an attempt to produce a competitive browser using the open source software model. That browser would eventually evolve into Firefox, which developed a respectable following while still in the beta stage of development; shortly after the release of Firefox 1.0 in late 2004, Firefox (all versions) accounted for 7% of browser use. As of August 2011, Firefox has a 28% usage share.

Apple's Safari had its first beta release in January 2003; as of April 2011, it had a dominant share of Apple-based web browsing, accounting for just over 7% of the entire browser market.

The most recent major entrant to the browser market is Chrome, first released in September 2008. Chrome's take-up has increased significantly year by year, by doubling its usage share from 8% to 16% by August 2011. This increase seems largely to be at the expense of Internet Explorer, whose share has tended to decrease from month to month. In December 2011, Chrome overtook Internet Explorer 8 as the most widely used web browser but still had lower usage than all versions of Internet Explorer combined. Chrome's user-base continued to grow and in May 2012, Chrome's usage passed the usage of all versions of Internet Explorer combined. By April 2014, Chrome's usage had hit 45%.

Internet Explorer will be deprecated in Windows 10, with Microsoft Edge replacing it as the default web browser.

Business models

The ways that web browser makers fund their development costs has changed over time. The first web browser, WorldWideWeb, was a research project.

Netscape Navigator was sold commercially, as was Opera.

Internet Explorer, on the other hand, was bundled free with the Windows operating system (and was also downloadable free), and therefore it was funded partly by the sales of Windows to computer manufacturers and direct to users. Internet Explorer also used to be available for the Mac. It is likely that releasing IE for the Mac was part of Microsoft's overall strategy to fight threats to its quasi-monopoly platform dominance – threats such as web standards and Java – by making some web developers, or at least their managers, assume that there was "no need" to develop for anything other than Internet Explorer. In this respect, IE may have contributed to Windows and Microsoft applications sales in another way, through "lock-in" to Microsoft's browser.

In January 2009, the European Commission announced it would investigate the bundling of Internet Explorer with Windows operating systems from Microsoft, saying "Microsoft's tying of Internet Explorer to the Windows operating system harms competition between web browsers, undermines product innovation and ultimately reduces consumer choice."

Safari and Mobile Safari were likewise always included with OS X and iOS respectively, so, similarly, they were originally funded by sales of Apple computers and mobile devices, and formed part of the overall Apple experience to customers.

Today, most commercial web browsers are paid by search engine companies to make their engine default, or to include them as another option. For example, Google pays Mozilla, the maker of Firefox, to make Google Search the default search engine in Firefox. Mozilla makes enough money from this deal that it does not need to charge users for Firefox. In addition, Google Search is also (as one would expect) the default search engine in Google Chrome. Users searching for websites or items on the Internet would be led to Google's search results page, increasing ad revenue and which funds development at Google and of Google Chrome.

Many less-well-known free software browsers, such as Konqueror, were hardly funded at all and were developed mostly by volunteers free of charge.

Function

The primary purpose of a web browser is to bring information resources to the user ("retrieval" or "fetching"), allowing them to view the information ("display", "rendering"), and then access other information ("navigation", "following links").

This process begins when the user inputs a Uniform Resource Locator (URL), for example *http://en.wikipedia.org/*, into the browser. The prefix of the URL, the Uniform Resource Identifier or URI, determines how the URL will be interpreted. The most commonly used kind of URI starts with *http:* and identifies a resource to be retrieved over the Hypertext Transfer Protocol(HTTP). Many browsers also support a variety of other prefixes, such as *https:* for HTTPS, *ftp:* for the File Transfer Protocol, and *file:* for local files. Prefixes that the web browser cannot directly handle are often handed off to another application entirely. For example, *mailto:* URIs are usually passed to the user's default e-mail application, and *news:* URIs are passed to the user's default newsgroup reader.

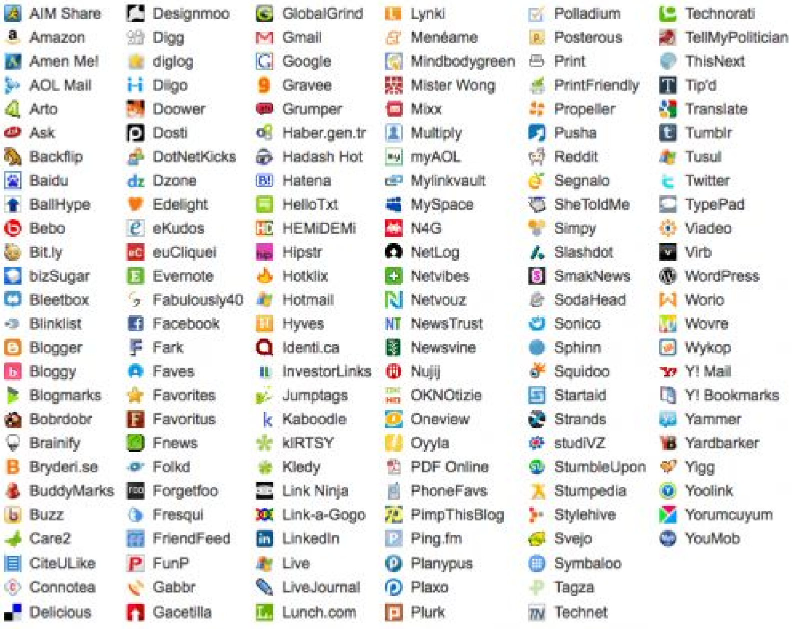
In the case of *http*, *https*, *file*, and others, once the resource has been retrieved the web browser will display it. HTML and associated content (image files, formatting information such as CSS, etc.) is passed to the browser's layout engine to be transformed from markup to an interactive document, a process known as "rendering". Aside from HTML, web browsers can generally display any kind of content that can be part of a web page. Most browsers can display images, audio, video, and XML files, and often have plug-ins to support Flash applications and Java applets. Upon encountering a file of an unsupported type or a file that is set up to be downloaded rather than displayed, the browser prompts the user to save the file to disk.

Information resources may contain hyperlinks to other information resources. Each link contains the URI of a resource to go to. When a link is clicked, the browser navigates to the resource indicated by the link's target URI, and the process of bringing content to the user begins again.

Features

Available web browsers range in features from minimal, text-based user interfaces with bare-bones support for HTML to rich user interfaces supporting a wide variety of file formats and protocols. Browsers which include additional components to support e-mail, Usenet news, and Internet Relay Chat (IRC), are sometimes referred to as “Internet suites” rather than merely “web browsers”.

All major web browsers allow the user to open multiple information resources at the same time, either in different browser windows or in different tabs of the same window. Major browsers also include pop-up blockers to prevent unwanted windows from “popping up” without the user’s consent.



AIM Share
Amazon
Amen Me!
AOL Mail
Arto
Ask
Backflip
Baidu
BallHype
Bebo
Bit.ly
bizSugar
Bleetbox
Blinklist
Blogger
Bloggy
Blogmarks
Bobrdobr
Brainify
Bryderi.se
BuddyMarks
Buzz
Care2
CiteULike
Connotea
Delicious

Designmoo
Digg
diglog
Diigo
Doower
Dosti
DotNetKicks
Dzone
Edelight
eKudos
euCliquei
Evernote
Fabulously40
Facebook
Fark
Faves
Favorites
Favoritus
Fnews
Folkd
Forgetfoo
Fresqui
FriendFeed
FunP
Gabbr
Gacetilla

GlobalGrind
Gmail
Google
Gravee
Grumper
Haber.gen.tr
Hadash Hot
Hatena
HelloTxt
HEMiDEMi
Hipstr
Hotklix
Hotmail
Hyves
Identi.ca
InvestorLinks
Jumptags
Kaboodle
kIRTSY
Kledy
Link Ninja
Link-a-Gogo
LinkedIn
Live
LiveJournal
Lunch.com

Lynki
Menéame
Mindbodygreen
Mister Wong
Mixx
Multiply
myAOL
Mylinkvault
MySpace
N4G
NetLog
Netvibes
Netvouz
NewsTrust
Newsvine
Nujij
OKNOtizie
Oneview
Oyyla
PDF Online
PhoneFavs
PimpThisBlog
Ping.fm
Planypus
Plaxo
Plurk

Polladium
Posterous
Print
PrintFriendly
Propeller
Pusha
Reddit
Segnalo
SheToldMe
Simpy
Slashdot
SmakNews
SodaHead
Sonico
Sphinn
Squidoo
Startaid
Strands
studiVZ
StumbleUpon
Stumpedia
Stylehive
Svejo
Symbaloo
Tagza
Technet

Technorati
TellMyPolitician
ThisNext
Tip'd
Translate
Tumblr
Tusul
Twitter
TypePad
Viadeo
Virb
WordPress
Worio
Wovre
Wykop
Y! Mail
Y! Bookmarks
Yammer
Yardbarker
Yigg
Yoolink
Yorumcuyum
YouMob

Browser bookmarks

Most web browsers can display a list of web pages that the user has *bookmarked* so that the user can quickly return to them. Bookmarks are also called “Favorites” in Internet Explorer. In addition, all major web browsers have some form of built-in web feed aggregator. In Firefox, web feeds are formatted as “live bookmarks” and behave like a folder of bookmarks corresponding to recent entries in the feed. In Opera, a more traditional feed reader is included which stores and displays the contents of the feed.

Furthermore, most browsers can be extended via plug-ins, downloadable components that provide additional features.

User interface

Most major web browsers have these user interface elements in common:

- *Back* and *forward* buttons to go back to the previous resource and forward respectively.
- A *refresh* or *reload* button to reload the current resource.
- A *stop* button to cancel loading the resource. In some browsers, the stop button is merged with the reload button.
- A *home* button to return to the user's home page.
- An address bar to input the Uniform Resource Identifier (URI) of the desired resource and display it.
- A search bar to input terms into a search engine. In some browsers, the search bar is merged with the address bar.
- A status bar to display progress in loading the resource and also the URI of links when the cursor hovers over them, and page zooming capability.
- The *viewport*, the visible area of the webpage within the browser window.
- The ability to view the HTML source for a page.

Some home media devices now include web browsers, like this LG Smart TV. The browser is controlled using an on-screen keyboard and LG's "Magic Motion" remote.

Major browsers also possess incremental find features to search within a web page.

Privacy and security

Most browsers support HTTP Secure and offer quick and easy ways to delete the web cache, cookies, and browsing history. For a comparison of the current security vulnerabilities of browsers, see comparison of web browsers.

Standards support

Early web browsers supported only a very simple version of HTML. The rapid development of proprietary web browsers led to the development of non-standard dialects of HTML, leading to problems with interoperability. Modern web browsers support a combination of standards-based and *de facto* HTML and XHTML, which should be rendered in the same way by all browsers.

Extensibility

A browser extension is a computer program that extends the functionality of a web browser. Every major web browser supports the development of browser extensions.

Components

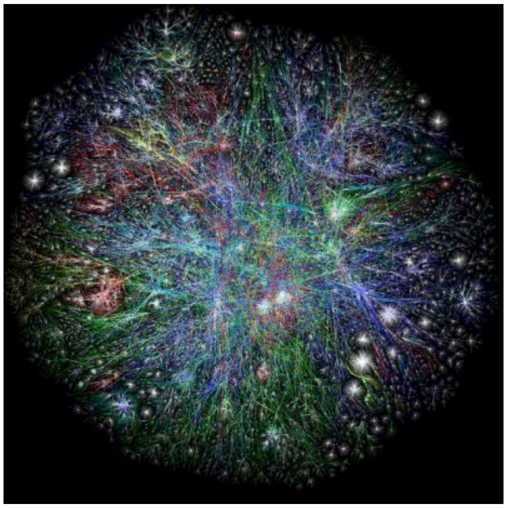
Web browsers consist of a user interface, layout engine, rendering engine, JavaScript interpreter, UI backend, networking component and data persistence component. These components achieve different functionalities of a web browser and together provide all capabilities of a web browser.

Licensing & Attributions

CC licensed content, Shared previously

- Web Browser. **Provided by:** Wikipedia. **Located at:** https://en.wikipedia.org/wiki/Web_browser. **License:** *CC BY-SA: Attribution-ShareAlike*
- Web browswers. **Authored by:** Geralt. **Located at:** https://pixabay.com/en/browser-web-www-computer-773215/. **License:** *CC0: No Rights Reserved*
- Clockwork Oyyla. **Authored by:** Anna Jumped. **Located at:** https://www.flickr.com/photos/17665603@N02/4023601737/. **License:** *CC BY-SA: Attribution-ShareAlike*

READING: INTERNET SECURITY



Internet security is a tree branch of computer security specifically related to the Internet, often involving browser security but also network security on a more general level as it applies to other applications or operating systems on a whole. Its objective is to establish rules and measures to use against attacks over the Internet. The Internet represents an insecure channel for exchanging information leading to a high risk of intrusion or fraud, such as phishing. Different methods have been used to protect the transfer of data, including encryption.

Types of security

Network layer security

TCP/IP which stands for Transmission Control Protocol (TCP) and Internet Protocol (IP) aka Internet protocol suite can be made secure with the help of cryptographic methods and protocols. These protocols include Secure Sockets Layer (SSL), succeeded by Transport Layer Security (TLS) for web traffic, Pretty Good Privacy (PGP) for email, and IPsec for the network layer security.

Internet Protocol Security (IPsec)

This protocol is designed to protect communication in a secure manner using TCP/IP aka Internet protocol suite. It is a set of security extensions developed by the Internet Task force IETF, and it provides security and authentication at the IP layer by transforming data using encryption. Two main types of transformation that form the basis of IPsec: the Authentication Header (AH) and ESP. These two protocols provide data integrity, data origin authentication, and anti-replay service. These protocols can be used alone or in combination to provide the desired set of security services for the Internet Protocol (IP) layer.

The basic components of the IPsec security architecture are described in terms of the following functionalities:

- Security protocols for AH and ESP
- Security association for policy management and traffic processing
- Manual and automatic key management for the Internet key exchange (IKE)
- Algorithms for authentication and encryption

The set of security services provided at the IP layer includes access control, data origin integrity, protection against replays, and confidentiality. The algorithm allows these sets to work independently without affecting other parts of the implementation. The IPsec implementation is operated in a host or security gateway environment giving protection to IP traffic.

Security token

Some online sites offer customers the ability to use a six-digit code which randomly changes every 30–60 seconds on a security token. The keys on the security token have built in mathematical computations and manipulate numbers based on the current time built into the device. This means that every thirty seconds there is only a certain array of numbers possible which would be correct to validate access to the online account. The website that the user is logging into would be made aware of that devices' serial number and would know the computation and correct time built into the device to verify that the number given is indeed one of the handful of six-digit numbers that works in that given 30-60 second cycle. After 30–60 seconds the device will present a new random six-digit number which can log into the website.

Electronic mail security (E-mail)

Background

Email messages are composed, delivered, and stored in a multiple step process, which starts with the message's composition. When the user finishes composing the message and sends it, the message is transformed into a standard format: an RFC 2822 formatted message. Afterwards, the message can be transmitted. Using a network connection, the mail client, referred to as a mail user agent (MUA), connects to a mail transfer agent (MTA) operating on the mail server. The mail client then provides the sender's identity to the server. Next, using the mail server commands, the client sends the recipient list to the mail server. The client then supplies the message. Once the mail server receives and processes the message, several events occur: recipient server identification, connection establishment, and message transmission. Using Domain Name System (DNS) services, the sender's mail server determines the mail server(s) for the recipient(s). Then, the server opens up a connection(s) to the recipient mail server(s) and sends the message employing a process similar to that used by the originating client, delivering the message to the recipient(s).

Pretty Good Privacy (PGP)

Pretty Good Privacy provides confidentiality by encrypting messages to be transmitted or data files to be stored using an encryption algorithm such Triple DES or CAST-128. Email messages can be protected by using cryptography in various ways, such as the following:

- Signing an email message to ensure its integrity and confirm the identity of its sender.
- Encrypting the body of an email message to ensure its confidentiality.
- Encrypting the communications between mail servers to protect the confidentiality of both message body and message header.

The first two methods, message signing and message body encryption, are often used together; however, encrypting the transmissions between mail servers is typically used only when two organizations want to protect emails regularly sent between each other. For example, the organizations could establish a virtual private network (VPN) to encrypt the communications between their mail servers over the Internet. Unlike methods that can only encrypt a message body, a VPN can encrypt entire messages, including email header information such as senders, recipients, and subjects. In some cases, organizations may need to protect header information. However, a VPN solution alone cannot provide a message signing mechanism, nor can it provide protection for email messages along the entire route from sender to recipient.

Multipurpose Internet Mail Extensions (MIME)

MIME transforms non-ASCII data at the sender's site to Network Virtual Terminal (NVT) ASCII data and delivers it to client's Simple Mail Transfer Protocol (SMTP) to be sent through the Internet. The server SMTP at the receiver's side receives the NVT ASCII data and delivers it to MIME to be transformed back to the original non-ASCII data.

Message Authentication Code

A Message authentication code (MAC) is a cryptography method that uses a secret key to encrypt a message. This method outputs a MAC value that can be decrypted by the receiver, using the same secret key used by the sender. The Message Authentication Code protects both a message's data integrity as well as its authenticity.

Firewalls

A computer firewall controls access between networks. It generally consists of gateways and filters which vary from one firewall to another. Firewalls also screen network traffic and are able to block traffic that is dangerous. Firewalls act as the intermediate server between SMTP and Hypertext Transfer Protocol (HTTP) connections.

Role of firewalls in web security

Firewalls impose restrictions on incoming and outgoing Network packets to and from private networks. Incoming or outgoing traffic must pass through the firewall; only authorized traffic is allowed to pass through it. Firewalls create checkpoints between an internal private network and the public Internet, also known as *choke points*(borrowed from the identical military term of a combat limiting geographical feature). Firewalls can create choke points based on IP source and TCP port number. They can also serve as the platform for IPsec. Using tunnel mode capability, firewall can be used to implement VPNs. Firewalls can also limit network exposure by hiding the internal network system and information from the public Internet.

Types of firewall

Packet filter

A packet filter is a first generation firewall that processes network traffic on a packet-by-packet basis. Its main job is to filter traffic from a remote IP host, so a router is needed to connect the internal network to the Internet. The router is known as a screening router, which screens packets leaving and entering the network.

Stateful packet inspection

In a stateful firewall the circuit-level gateway is a proxy server that operates at the network level of an Open Systems Interconnection (OSI) model and statically defines what traffic will be allowed. Circuit proxies will forward Network packets (formatted unit of data) containing a given port number, if the port is permitted by the algorithm. The main advantage of a proxy server is its ability to provide Network Address Translation (NAT), which can hide the user’s IP address from the Internet, effectively protecting all internal information from the Internet.

Application-level gateway

An application-level firewall is a third generation firewall where a proxy server operates at the very top of the OSI model, the IP suite application level. A network packet is forwarded only if a connection is established using a known protocol. Application-level gateways are notable for analyzing entire messages rather than individual packets of data when the data are being sent or received.

Malicious software

A computer user can be tricked or forced into downloading software onto a computer that is of malicious intent. Such software comes in many forms, such as viruses, Trojan horses, spyware, and worms.

- Malware, short for malicious software, is any software used to disrupt computer operation, gather sensitive information, or gain access to private computer systems. Malware is defined by its malicious intent, acting against the requirements of the computer user, and does not include software that causes unintentional harm due to some deficiency. The term badware is sometimes used, and applied to both true (malicious) malware and unintentionally harmful software.

- A botnet is a network of zombie computers that have been taken over by a robot or bot that performs large-scale malicious acts for the creator of the botnet.
- Computer Viruses are programs that can replicate their structures or effects by infecting other files or structures on a computer. The common use of a virus is to take over a computer to steal data.
- Computer worms are programs that can replicate themselves throughout a computer network, performing malicious tasks throughout.
- Ransomware is a type of malware which restricts access to the computer system that it infects, and demands a ransom paid to the creator(s) of the malware in order for the restriction to be removed.
- Scareware is scam software with malicious payloads, usually of limited or no benefit, that are sold to consumers via certain unethical marketing practices. The selling approach uses social engineering to cause shock, anxiety, or the perception of a threat, generally directed at an unsuspecting user.
- Spyware refers to programs that surreptitiously monitor activity on a computer system and report that information to others without the user's consent.
- A Trojan horse, commonly known as a *Trojan*, is a general term for malicious software that pretends to be harmless, so that a user willingly allows it to be downloaded onto the computer.

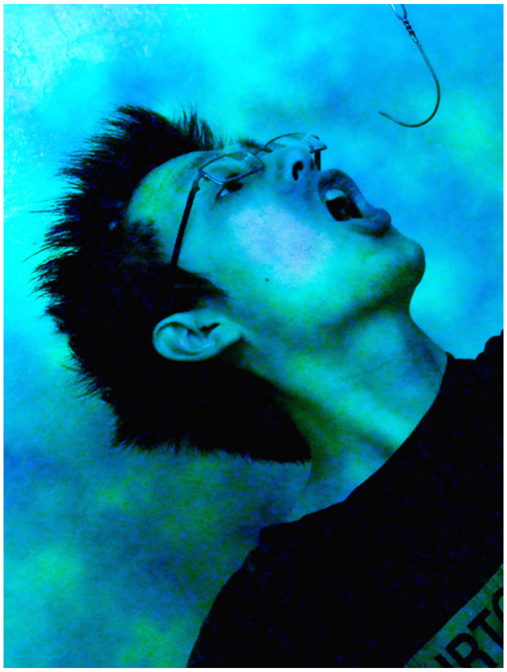
Denial-of-service attack

A denial-of-service attack (DoS attack) or distributed denial-of-service attack (DDoS attack) is an attempt to make a computer resource unavailable to its intended users. Although the means to carry out, motives for, and targets of a DoS attack may vary, it generally consists of the concerted efforts to prevent an Internet site or service from functioning efficiently or at all, temporarily or indefinitely. According to businesses who participated in an international business security survey, 25% of respondents experienced a DoS attack in 2007 and 16.8% experienced one in 2010.

Phishing

Phishing is another common threat to the Internet. "SA, the Security Division of EMC, today announced the findings of its January 2013 Fraud Report, estimating the global losses from Phishing at $1.5 Billion in 2012.". Filter evasion, website forgery, phone phishing, Covert Redirect are some well known phishing techniques.

Hackers use a variety of tools to conduct phishing attacks. They create forged websites that pretend to be other websites in order for users to leave their personal information. These hackers usually host these sites on legitimate hosting services using stolen credit cards while the last trend is to use a mailing system and finding a mailing list of people which they can try and fraud.



Browser choice

Web browser statistics tend to affect the amount a Web browser is exploited. For example, Internet Explorer 6, which used to own a majority of the Web browser market share, is considered extremely insecure because vulnerabilities were exploited due to its former popularity. Since browser choice is more evenly distributed (Internet Explorer at 28.5%, Firefox at 18.4%, Google Chrome at 40.8%, and so on) and vulnerabilities are exploited in many different browsers.

Application vulnerabilities

Applications used to access Internet resources may contain security vulnerabilities such as memory safety bugs or flawed authentication checks. The most severe of these bugs can give network attackers full control over the computer. Most security applications and suites are incapable of adequate defense against these kinds of attacks.

Internet security products

Antivirus

Antivirus and Internet security programs can protect a programmable device from malware by detecting and eliminating viruses; Antivirus software was mainly shareware in the early years of the Internet, but there are now several free security applications on the Internet to choose from for all platforms.

Security suites

So called "security suites" were first offered for sale in 2003 (McAfee) and contain a suite of firewalls, anti-virus, anti-spyware and more. They may now offer theft protection, portable storage device safety check, private Internet browsing, cloud anti-spam, a file shredder or make security-related decisions (answering popup windows) and several were free of charge as of at least 2012.

Licensing & Attributions

CC licensed content, Shared previously

- Internet Security. **Provided by:** Wikipedia . **Located at:** https://en.wikipedia.org/wiki/Internet_security. **License:** *CC BY-SA: Attribution-ShareAlike*
- Network maps. **Authored by:** NetMapper. **Located at:** https://www.flickr.com/photos/netmapper/6817128312/. **License:** *CC BY-NC: Attribution-NonCommercial*
- Firewall. **Authored by:** cosmo_71. **Located at:** https://www.flickr.com/photos/cosmok/10766534554/. **License:** *CC BY-ND: Attribution-NoDerivatives*
- Hook, Line, Sinker (How I fell for a phishing scam. **Authored by:** Kenneth Lu. **Located at:** https://www.flickr.com/photos/toasty/1276202472/. **License:** *CC BY: Attribution*

READING: NAME SERVER

A **name server** is a computer hardware or software server that implements a network service for providing responses to queries against a directory service. It translates an often humanly meaningful, text-based identifier to a system-internal, often numeric identification or addressing component. This service is performed by the server in response to a service protocol request.

An example of a name server is the server component of the Domain Name System (DNS), one of the two principal name spaces of the Internet. The most important function of DNS servers is the translation (resolution) of human-memorable domain names and hostnames into the corresponding numeric Internet Protocol (IP) addresses, the second principal name space of the Internet which is used to identify and locate computer systems and resources on the Internet.

Domain Name System

The Internet maintains two principal namespaces: the domain name hierarchy and the IP address system. The Domain Name System maintains the domain namespace and provides translation services between these two namespaces. Internet name servers implement the Domain Name System. The top hierarchy of the Domain Name System is served by the root name servers maintained by delegation by the Internet Corporation for Assigned Names and Numbers (ICANN). Below the root, Internet resources are organized into a hierarchy of domains, administered by the respective registrars and domain name holders. A DNS name server is a server that stores the DNS records, such as address (A, AAAA) records, name server (NS) records, and mail exchanger (MX) records for a domain name (see also List of DNS record types) and responds with answers to queries against its database.

Authoritative name server

Authoritative name server is a name server that gives answers in response to questions asked about names in a zone. An authoritative-only name server returns answers only to queries about domain names that have been specifically configured by the administrator. Name servers can also be configured to give authoritative answers to queries in some zones, while acting as a caching name server for all other zones.

An authoritative name server can either be a *primary* server (master) or a *secondary* server (slave). A primary server for a zone is the server that stores the definitive versions of all records in that zone. A secondary server for a zone uses an automatic updating mechanism to maintain an identical copy of the primary server's database for a zone. Examples of such mechanisms include DNS zone transfers and file transfer protocols. DNS provides a mechanism whereby the primary for a zone can notify all the known secondaries for that zone when the contents of the zone have changed. The contents of a zone are either manually configured by an administrator, or managed using Dynamic DNS.

Every domain name appears in a zone served by one or more authoritative name servers. The fully qualified domain names of the authoritative name servers of a zone are listed in the NS records of that zone. If the server for a zone is not also authoritative for its parent zone, the server for the parent zone must be configured with a delegation for the zone.

When a domain is registered with a domain name registrar, the zone administrator provides the list of name servers (typically at least two, for redundancy) that are authoritative for the zone that contains the domain. The registrar provides the names of these servers to the domain registry for the top level domain containing the zone. The domain registry in turn configures the authoritative name servers for that top level domain with delegations for each server for the zone. If the fully qualified domain name of any name server for a zone appears within that zone, the zone administrator provides IP addresses for that name server, which are installed in the parent zone as glue records; otherwise, the delegation consists of the list of NS records for that zone.

Authoritative answer

A name server indicates that its response is authoritative by setting the *Authoritative Answer* (*AA*) bit in the response to a query on a name for which it is authoritative. Name servers providing answers for which they are not authoritative (for example, name servers for parent zones), do not set the *AA* bit.

Recursive query

If a name server cannot answer a query because it does not contain an entry for the host in its database, it may recursively query name servers higher up in the hierarchy. This is known as a *recursive query* or *recursive lookup*. In principle, authoritative name servers suffice for the operation of the Internet. However, with only authoritative name-servers operating, every DNS query must start with recursive queries at the root zone of the Domain Name System and each user system must implement resolver software capable of recursive operation.

Caching name server

Caching name servers (*DNS caches*) store DNS query results for a period of time determined in the configuration (time-to-live) of each domain-name record. DNS caches improve the efficiency of the DNS by reducing DNS traffic across the Internet, and by reducing load on authoritative name-servers, particularly root name-servers. Because they can answer questions more quickly, they also increase the performance of end-user applications that use the DNS. *Recursive name servers* resolve any query they receive, even if they are not authoritative for the question being asked, by consulting the server or servers that are authoritative for the question. Caching name servers are often also recursive name servers—they perform every step necessary to answer any DNS query they receive. To do this the name server queries each authoritative name-server in turn, starting from the DNS root zone. It continues until it reaches the authoritative server for the zone that contains the queried domain name. That server provides the answer to the question, or definitively says it can't be answered, and the *caching resolver* then returns this response to the client that asked the question. The authority, resolving and caching functions can all be present in a DNS server implementation, but this is not required: a DNS server can implement any one of these functions alone, without implementing the others. Internet service providers typically provide caching resolvers for their customers. In addition, many home-networking routers implement caching resolvers to improve efficiency in the local network. Some systems utilize `nscd`—the name service caching daemon.

Microsoft networking

Name servers also exist on some Microsoft Windows networks where one host assumes the role of NetBIOS browse master and performs as a NBNS server. Small local area networks of Windows systems require no central name server, and generally perform name-resolution using a broadcast algorithm.

The Windows Internet Name Service (WINS) is a name service that translates NetBIOS names to numerical addresses.

Licensing & Attributions

CC licensed content, Shared previously

- Name Server. **Provided by:** Wikipedia. **Located at:** https://en.wikipedia.org/w/index.php?title=Name_server&redirect=no. **License:** *CC BY-SA: Attribution-ShareAlike*
- Domain name hosting. **Authored by:** seolinkbuilding.org. **Located at:** https://www.flickr.com/photos/india_7/. **License:** *CC BY: Attribution*
- Recursive purple flower. **Authored by:** Alexandre Duret-Lutz. **Located at:** https://www.flickr.com/photos/gadl/268053770/. **License:** *CC BY-NC: Attribution-NonCommercial*

READING: IP ADDRESS

Introduction

An **Internet Protocol address** (**IP address**) is a numerical label assigned to each device (e.g., computer, printer) participating in a computer network that uses the Internet Protocol for communication. An IP address serves two principal functions: host or network interface identification and location addressing. Its role has been characterized as follows: "A name indicates what we seek. An address indicates where it is. A route indicates how to get there."

The designers of the Internet Protocol defined an IP address as a 32-bit number and this system, known as Internet Protocol Version 4 (IPv4), is still in use today. However, because of the growth of the Internet and the predicted depletion of available addresses, a new version of IP (IPv6), using 128 bits for the address, was developed in 1995. IPv6 was standardized as RFC 2460 in 1998, and its deployment has been ongoing since the mid-2000s.

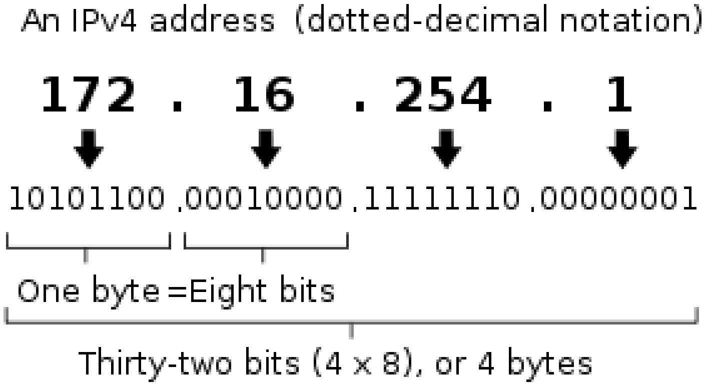
IP addresses are usually written and displayed in human-readable notations, such as 172.16.254.1 (IPv4), and 2001:db8:0:1234:0:567:8:1 (IPv6).

The Internet Assigned Numbers Authority (IANA) manages the IP address space allocations globally and delegates five regional Internet registries (RIRs) to allocate IP address blocks to local Internet registries (Internet service providers) and other entities.

IP versions

Two versions of the Internet Protocol (IP) are in use: IP Version 4 and IP Version 6. Each version defines an IP address differently. Because of its prevalence, the generic term *IP address* typically still refers to the addresses defined by IPv4. The gap in version sequence between IPv4 and IPv6 resulted from the assignment of number 5 to the experimental Internet Stream Protocol in 1979, which however was never referred to as IPv5.

IPv4 addresses


An IPv4 address (dotted-decimal notation)
172 . 16 . 254 . 1
10101100 .00010000 .11111110 .00000001
One byte =Eight bits
Thirty-two bits (4 x 8), or 4 bytes

Decomposition of an IPv4 address from dot-decimal notation to its binary value.

In IPv4 an address consists of 32 bits which limits the address space to 4294967296 (2^{32}) possible unique addresses. IPv4 reserves some addresses for special purposes such as private networks (~18 million addresses) or multicast addresses (~270 million addresses).

IPv4 addresses are canonically represented in dot-decimal notation, which consists of four decimal numbers, each ranging from 0 to 255, separated by dots, e.g., 172.16.254.1. Each part represents a group of 8 bits (octet) of the address. In some cases of technical writing, IPv4 addresses may be presented in various hexadecimal, octal, or binary representations.

Subnetting

In the early stages of development of the Internet Protocol, network administrators interpreted an IP address in two parts: network number portion and host number portion. The highest order octet (most significant eight bits) in an address was designated as the *network number* and the remaining bits were called the *rest field* or *host identifier* and were used for host numbering within a network.

This early method soon proved inadequate as additional networks developed that were independent of the existing networks already designated by a network number. In 1981, the Internet addressing specification was revised with the introduction of classful network architecture.

Classful network design allowed for a larger number of individual network assignments and fine-grained subnetwork design. The first three bits of the most significant octet of an IP address were defined as the *class* of the address. Three classes (*A*, *B*, and *C*) were defined for universal unicast addressing. Depending on the class derived, the network identification was based on octet boundary segments of the entire address. Each class used successively additional octets in the network identifier, thus reducing the possible number of hosts in the higher order classes (*B* and *C*). The following table gives an overview of this now obsolete system.

Historical classful network architecture

Class	Leading bits	Size of ***network number*** bit field	Size of ***rest*** bit field	Number of networks	Addresses per network	Start address	End address
A	0	8	24	128 (2^7)	16,777,216 (2^{24})	0.0.0.0	127.255.255.255
B	10	16	16	16,384 (2^{14})	65,536 (2^{16})	128.0.0.0	191.255.255.255
C	110	24	8	2,097,152 (2^{21})	256 (2^8)	192.0.0.0	223.255.255.255

Classful network design served its purpose in the startup stage of the Internet, but it lacked scalability in the face of the rapid expansion of the network in the 1990s. The class system of the address space was replaced with Classless Inter-Domain Routing (CIDR) in 1993. CIDR is based on variable-length subnet masking (VLSM) to allow allocation and routing based on arbitrary-length prefixes.

Today, remnants of classful network concepts function only in a limited scope as the default configuration parameters of some network software and hardware components (e.g. netmask), and in the technical jargon used in network administrators' discussions.

Private addresses

Early network design, when global end-to-end connectivity was envisioned for communications with all Internet hosts, intended that IP addresses be uniquely assigned to a particular computer or device. However, it was found that this was not always necessary as private networks developed and public address space needed to be conserved.

Computers not connected to the Internet, such as factory machines that communicate only with each other via TCP/IP, need not have globally unique IP addresses. Three ranges of IPv4 addresses for private networks were reserved in RFC 1918. These addresses are not routed on the Internet and thus their use need not be coordinated with an IP address registry.

Today, when needed, such private networks typically connect to the Internet through network address translation (NAT).

IANA-reserved private IPv4 network ranges

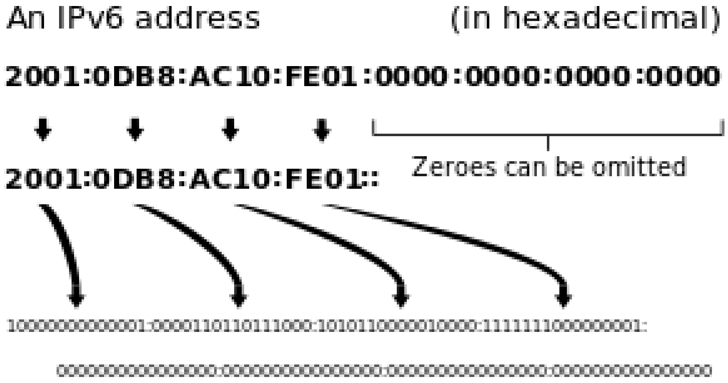
	Start	End	No. of addresses
24-bit block (/8 prefix, 1 × A)	10.0.0.0	10.255.255.255	16777216
20-bit block (/12 prefix, 16 × B)	172.16.0.0	172.31.255.255	1048576
16-bit block (/16 prefix, 256 × C)	192.168.0.0	192.168.255.255	65536

Any user may use any of the reserved blocks. Typically, a network administrator will divide a block into subnets; for example, many home routers automatically use a default address range of 192.168.0.0 through 192.168.0.255 (192.168.0.0/24).

IPv4 address exhaustion

High levels of demand have decreased the supply of unallocated Internet Protocol Version 4 (IPv4) addresses available for assignment to Internet service providers and end user organizations since the 1980s. This development is referred to as IPv4 address exhaustion. IANA's primary address pool was exhausted on 3 February 2011, when the last five blocks were allocated to the five RIRs.[5][6] APNIC was the first RIR to exhaust its regional pool on 15 April 2011, except for a small amount of address space reserved for the transition to IPv6, intended to be allocated in a restricted process.[7]

IPv6 addresses



Decomposition of an IPv6 address from hexadecimal representation to its binary value.

The rapid exhaustion of IPv4 address space prompted the Internet Engineering Task Force (IETF) to explore new technologies to expand the addressing capability in the Internet. The permanent solution was deemed to be a redesign of the Internet Protocol itself. This new generation of the Internet Protocol was eventually named *Internet Protocol Version 6* (IPv6) in 1995. The address size was increased from 32 to 128 bits (16 octets), thus providing up to 2^{128} (approximately $3.403×10^{38}$) addresses. This is deemed sufficient for the foreseeable future.

The intent of the new design was not to provide just a sufficient quantity of addresses, but also redesign routing in the Internet by more efficient aggregation of subnetwork routing prefixes. The resulted in slower growth of routing tables in routers. The smallest possible individual allocation is a subnet for 2^{64} hosts, which is the square of the size of the entire IPv4 Internet. At these levels, actual address utilization rates will be small on any IPv6 network segment. The new design also provides the opportunity to separate the addressing infrastructure of a network segment, i.e. the local administration of the segment's available space, from the addressing prefix used to route traffic to and from external networks. IPv6 has facilities that automatically change the routing prefix of entire networks, should the global connectivity or the routing policy change, without requiring internal redesign or manual renumbering.

The large number of IPv6 addresses allows large blocks to be assigned for specific purposes and, where appropriate, to be aggregated for efficient routing. With a large address space, there is no need to have complex address conservation methods as used in CIDR.

All modern desktop and enterprise server operating systems include native support for the IPv6 protocol, but it is not yet widely deployed in other devices, such as residential networking routers, voice over IP (VoIP) and multimedia equipment, and network peripherals.

Private addresses

Just as IPv4 reserves addresses for private networks, blocks of addresses are set aside in IPv6. In IPv6, these are referred to as unique local addresses (ULA). RFC 4193 reserves the routing prefix fc00::/7 for this block which is divided into two /8 blocks with different implied policies. The addresses include a 40-bit pseudorandom number that minimizes the risk of address collisions if sites merge or packets are misrouted.[8]

Early practices used a different block for this purpose (fec0::), dubbed site-local addresses. However, the definition of what constituted *sites* remained unclear and the poorly defined addressing policy created ambiguities for routing. This address type was abandoned and must not be used in new systems.

Addresses starting with fe80:, called link-local addresses, are assigned to interfaces for communication on the attached link. The addresses are automatically generated by the operating system for each network interface. This provides instant and automatic communication between all IPv6 host on a link. This feature is required in the lower layers of IPv6 network administration, such as for the Neighbor Discovery Protocol.

Private address prefixes may not be routed on the public Internet.

IP subnetworks

IP networks may be divided into subnetworks in both IPv4 and IPv6. For this purpose, an IP address is logically recognized as consisting of two parts: the *network prefix* and the *host identifier*, or *interface identifier* (IPv6). The subnet mask or the CIDR prefix determines how the IP address is divided into network and host parts.

The term *subnet mask* is only used within IPv4. Both IP versions however use the CIDR concept and notation. In this, the IP address is followed by a slash and the number (in decimal) of bits used for the network part, also called the *routing prefix*. For example, an IPv4 address and its subnet mask may be 192.0.2.1 and 255.255.255.0, respectively. The CIDR notation for the same IP address and subnet is 192.0.2.1/24, because the first 24 bits of the IP address indicate the network and subnet.

IP address assignment

Internet Protocol addresses are assigned to a host either anew at the time of booting, or permanently by fixed configuration of its hardware or software. Persistent configuration is also known as using a *static IP address.* In contrast, in situations when the computer's IP address is assigned newly each time, this is known as using a *dynamic IP address*.

Methods

Static IP addresses are manually assigned to a computer by an administrator. The exact procedure varies according to platform. This contrasts with dynamic IP addresses, which are assigned either by the computer interface or host software itself, as in Zeroconf, or assigned by a server using Dynamic Host Configuration Protocol (DHCP). Even though IP addresses assigned using DHCP may stay the same for long periods of time, they can generally change. In some cases, a network administrator may implement dynamically assigned static IP addresses. In this case, a DHCP server is used, but it is specifically configured to always assign the same IP address to a particular computer. This allows static IP addresses to be configured centrally, without having to specifically configure each computer on the network in a manual procedure.

In the absence or failure of static or stateful (DHCP) address configurations, an operating system may assign an IP address to a network interface using state-less auto-configuration methods, such as Zeroconf.

Uses of dynamic address assignment

IP addresses are most frequently assigned dynamically on LANs and broadband networks by the Dynamic Host Configuration Protocol (DHCP). They are used because it avoids the administrative burden of assigning specific static addresses to each device on a network. It also allows many devices to share limited address space on a network if only some of them will be online at a particular time. In most current desktop operating systems, dynamic IP configuration is enabled by default so that a user does not need to manually enter any settings to connect to a network with a DHCP server. DHCP is not the only technology used to assign IP addresses dynamically. Dialup and some broadband networks use dynamic address features of the Point-to-Point Protocol.

Sticky dynamic IP address

A *sticky dynamic IP address* is an informal term used by cable and DSL Internet access subscribers to describe a dynamically assigned IP address which seldom changes. The addresses are usually assigned with DHCP. Since the modems are usually powered on for extended periods of time, the address leases are usually set to long periods and simply renewed. If a modem is turned off and powered up again before the next expiration of the address lease, it will most likely receive the same IP address.

Address autoconfiguration

RFC 3330 defines an address block, 169.254.0.0/16, for the special use in link-local addressing for IPv4 networks. In IPv6, every interface, whether using static or dynamic address assignments, also receives a local-link address automatically in the block fe80::/10.

These addresses are only valid on the link, such as a local network segment or point-to-point connection, that a host is connected to. These addresses are not routable and like private addresses cannot be the source or destination of packets traversing the Internet.

When the link-local IPv4 address block was reserved, no standards existed for mechanisms of address autoconfiguration. Filling the void, Microsoft created an implementation that is called Automatic Private IP Addressing (APIPA). APIPA has been deployed on millions of machines and has, thus, become a de facto standard in the industry. In RFC 3927, the IETF defined a formal standard for this functionality, entitled *Dynamic Configuration of IPv4 Link-Local Addresses*.

Uses of static addressing

Some infrastructure situations have to use static addressing, such as when finding the Domain Name System (DNS) host that will translate domain names to IP addresses. Static addresses are also convenient, but not absolutely necessary, to locate servers inside an enterprise. An address obtained from a DNS server comes with a time to live, or caching time, after which it should be looked up to confirm that it has not changed. Even static IP addresses do change as a result of network administration (RFC 2072).

Routing

IP addresses are classified into several classes of operational characteristics: unicast, multicast, anycast and broadcast addressing.

Unicast addressing

The most common concept of an IP address is in unicast addressing, available in both IPv4 and IPv6. It normally refers to a single sender or a single receiver, and can be used for both sending and receiving. Usually, a unicast address is associated with a single device or host, but a device or host may have more than one unicast address. Some individual PCs have several distinct unicast addresses, each for its own distinct purpose. Sending the same data to multiple unicast addresses requires the sender to send all the data many times over, once for each recipient.

Broadcast addressing

In IPv4 it is possible to send data to all possible destinations (“all-hosts broadcast”), which permits the sender to send the data only once, and all receivers receive a copy of it. In the IPv4 protocol, the address 255.255.255.255 is used for local broadcast. In addition, a directed (limited) broadcast can be made by combining the network prefix with a host suffix composed entirely of binary 1s. For example, the destination address used for a directed broadcast to devices on the 192.0.2.0/24 network is 192.0.2.255. IPv6 does not implement broadcast addressing and replaces it with multicast to the specially-defined all-nodes multicast address.

Multicast addressing

A multicast address is associated with a group of interested receivers. In IPv4, addresses 224.0.0.0 through 239.255.255.255 (the former Class D addresses) are designated as multicast addresses. IPv6 uses the address block with the prefix ff00::/8 for multicast applications. In either case, the sender sends a single datagram from its unicast address to the multicast group address and the intermediary routers take care of making copies and sending them to all receivers that have joined the corresponding multicast group.

Anycast addressing

Like broadcast and multicast, anycast is a one-to-many routing topology. However, the data stream is not transmitted to all receivers, just the one which the router decides is logically closest in the network. Anycast address is an inherent feature of only IPv6. In IPv4, anycast addressing implementations typically operate using the shortest-path metric of BGP routingand do not take into account congestion or other attributes of the path. Anycast methods are useful for global load balancing and are commonly used in distributed DNS systems.

Public addresses

A *public IP address*, in common parlance, is synonymous with a *globally routable unicast IP address*.

Both IPv4 and IPv6 define address ranges that are reserved for private networks and link-local addressing. The term public IP address often used excludes these types of addresses.

Modifications to IP addressing

IP blocking and firewalls

Firewalls perform Internet Protocol blocking to protect networks from unauthorized access. They are common on today's Internet. They control access to networks based on the IP address of a client computer. Whether using a blacklist or a whitelist, the IP address that is blocked is the perceived IP address of the client, meaning that if the client is using a proxy server or network address translation, blocking one IP address may block many individual computers.

IP address translation

Multiple client devices can appear to share IP addresses: either because they are part of a shared hosting web server environment or because an IPv4 network address translator (NAT) or proxy server acts as an intermediary agent on behalf of its customers, in which case the real originating IP addresses might be hidden from the server receiving a request. A common practice is to have a NAT hide a large number of IP addresses in a private network. Only the "outside" interface(s) of the NAT need to have Internet-routable addresses.

Most commonly, the NAT device maps TCP or UDP port numbers on the side of the larger, public network to individual private addresses on the masqueraded network.

In small home networks, NAT functions are usually implemented in a residential gateway device, typically one marketed as a "router". In this scenario, the computers connected to the router would have private IP addresses and the router would have a public address to communicate on the Internet. This type of router allows several computers to share one public IP address.

Diagnostic tools

Computer operating systems provide various diagnostic tools to examine their network interface and address configuration. Windows provides the command-line interface tools ipconfig and netsh and users of Unix-like systems can use ifconfig, netstat, route, lanstat, fstat, or iproute2 utilities to accomplish the task.

Licensing & Attributions

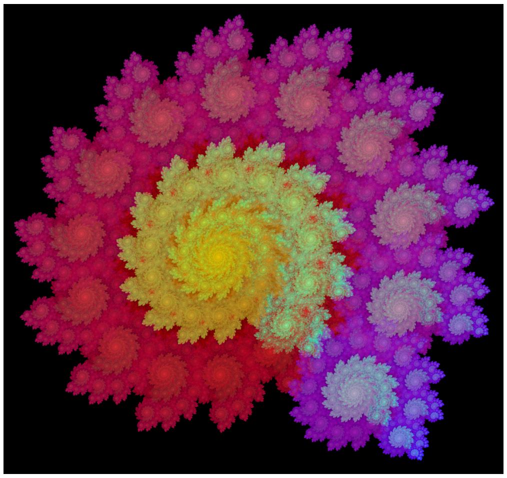
CC licensed content, Shared previously

- IP Address. **Provided by:** Wikipedia. **Located at:** https://en.wikipedia.org/wiki/IP_address. **License:** *CC BY-SA: Attribution-ShareAlike*
- Tiny house move. **Authored by:** Tammy Strobel. **Located at:** https://www.flickr.com/photos/rowdykittens/7944094174/. **License:** *CC BY: Attribution*
- route-moutonniere Alger. **Authored by:** abdelkrim Sa. **Located at:** https://www.flickr.com/photos/algeria_imaging_pro/16554842499/. **License:** *CC BY-NC-ND: Attribution-NonCommercial-NoDerivatives*

MODULE 6: ETHICS AND SOFTWARE DEVELOPMENT

READING: SYSTEM DEVELOPMENT

Introduction



This chapter will provide you with an overview of the systems development process. First we describe in detail the traditional Systems Development Life Cycle (SDLC), encompassing the stages through which each system should pass, from the initial survey to hand-over of the completed system.

Pressure for rapid development and future maintainability of systems has resulted in a number of alternative approaches to systems development, ranging from development by end-users, to the incorporation of formal methods to improve the quality and efficiency of the development process.

Systems Development Life Cycle (SDLC)

Systems development could be seen as the simple process of writing programs to solve the needs of the user. Unfortunately the user knows what he wants but has no technical expertise while the programmer understands the computer but not the user environment. This communication gap between the customer and the service provider must be handled by an intermediary, the systems analyst. Broadly speaking therefore the systems analyst translates user's needs into detailed specifications for implementation by the programmer.

Over the years the software manufacturing process has become more formalized:

> The basic idea of the systems development life cycle is that there is a well defined process by which an application is conceived, developed and implemented. The life cycle gives structure to a creative process. In order to manage and control the development effort, it is necessary to know what should have been done, what has been done, and what has yet to be accomplished. The phases in the systems development life cycle provide a basis for management and control because they define segments of the workflow which can be identified for managerial purposes and specify the documents or other deliverables to be produced in each phase.
>
> —Davis and Olson, 1985.

The number of stages and names to describe those stages differ slightly between organizations; but the SDLC normally covers the activities shown in figure 1, each with a primary purpose.

Preliminary Investigation

The preliminary investigation is carried out to determine the scope and objectives of the new system and to investigate whether there is a feasible solution. New applications normally originate from end-user requests and are weighed against the other requests for IS resources before approval to develop the system is granted. At this stage an analyst or small project team is authorized to investigate the real potential of the new application. During this brief study the analyst must investigate the problem and the existing system sufficiently to be able to identify the true extent and purpose of the new application.

In order to ensure that the new system would be of greater benefit to the organization than other competing requests for proposals, a feasibility study must be performed covering the following three major areas:

- Economic feasibility to measure the costs and benefits of the new system.
- Technical feasibility to ensure that the organization has sufficient hardware, software and personnel resources to develop and support the proposed system.
- Operational feasibility, the willingness and ability of management, users and Information Systems staff in the organization to build and use the proposed system

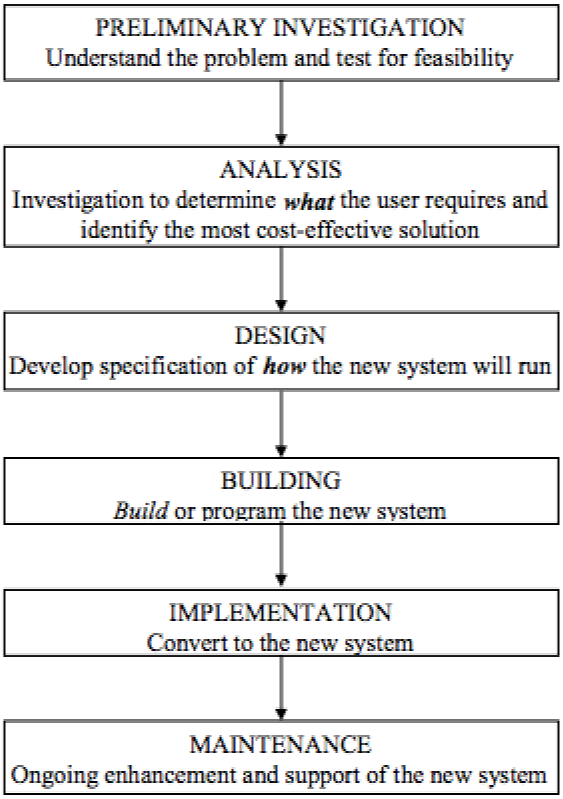

PRELIMINARY INVESTIGATION
Understand the problem and test for feasibility
ANALYSIS
Investigation to determine *what* the user requires and identify the most cost-effective solution
DESIGN
Develop specification of *how* the new system will run
BUILDING
Build or program the new system
IMPLEMENTATION
Convert to the new system
MAINTENANCE
Ongoing enhancement and support of the new system

Figure 1: Systems Development Process

Issues such as the size and complexity of the new system and the skills and availability of user and IS staff, will determine the level of potential risk to the organization in developing the system.

The output from this preliminary investigation is a statement of scope and objectives (often termed the project charter) together with a feasibility report. This document is submitted to management where a decision is made as to whether or not the development project should continue.

Systems Analysis

In this stage the analyst investigates the needs of the user and develops a conceptual solution to the problem. One human failing we all tend to exhibit is to rush into proposing solutions before we fully understand the problem we are trying to solve. It is therefore important for the analyst to develop a broad, conceptual solution to the problem (what needs to be done) prior to launching into the detailed physical design where we specify how the system will work.

In the past analysis tended to be very much a pragmatic affair with success more dependent on the experience and capabilities of the analyst than on any formalized approach. The analysis phase should include the following discrete steps:

- Understand how the existing system operates. This information can be obtained by observing people at work, interviewing users, and studying procedure manuals and other supporting documentation, questionnaires and visits to other organizations.
- Document the current physical system. A major problem in the past was how to record all the detail about the system. Most of it could be found only in the analyst's head or draft notes. Here the basic tools of structured systems analysis such as the data flow diagram (DFD), the entity relationship diagram (ERD) and data dictionary (DD) can be used to represent graphically and record the data and procedures. We will discuss these later in the chapter.
- Define the problem areas. These may include such issues as poor response times in the existing system, poor presentation of current information, high costs or weak controls in the current system, waste and sometimes duplication.
- Identify new requirements. The analyst must attempt to identify new user requirements and look for new and improved procedures that can be incorporated into the system.
- Identify possible solutions. Having derived objectives for the new system from the previous stage, the analyst now develops a conceptual model of the new system in conjunction with the user. This may involve the investigation of alternative physical designs, such as whether to remain with the existing manual system, or to choose a centralized or decentralized approach to the application.
- The culmination of the analysis stage is the preparation of the formal user requirement specification (URS) that incorporates a logical model of a system that will meet the user's requirements. A large proportion of the functional description and data specifications is best communicated from the analysis stage to the design stage through the graphic and electronic output from the structured tools used in the analysis process (data flow diagrams, entity relationship models, decision trees and the data dictionary).
- Again management is required to review the status of the project and to make its go/no-go decision.

Systems Design

The analysis stage of the SDLC has clearly identified what must be done in order to meet the user's requirements. One important decision that must be taken at this point is whether to "make or buy" the new software application. In the past, most large organizations developed their own applications as no two organizations were exactly alike, and they could afford the investment in systems developed around their user's needs.

Today the picture is changing as custom-built software is becoming very expensive to develop and even more so to maintain. Computer applications are large, complex and integrated and many businesses have become non-competitive because of their inability to develop systems that adequately support their business activities.

On the reverse side, packages (pre-written software applications) are becoming more common and can be customized to meet the needs of each organization. With major benefits in terms of speed of installation, cost, low maintenance and low risk, more and more companies are switching to packaged applications software. It is at this stage in the SDLC that the "make or buy" decision must be taken. We have analyzed our user's requirements and can use these as selection criteria in searching for an appropriate package to purchase and install. Where there is no suitable package available we can still look to other innovative ways of obtaining the software, such as hiring contract staff or appointing a software house to build the system for us. Purchasing pre-written software will obviously mean the detailed systems design, coding and testing phases of the project are bypassed, depending on the need for customization of the final system. In later courses we will look at the package selection process in more detail.

The objective of the design stage is to determine exactly how the new system will work, and to communicate this information in a document referred to as the detailed systems specification. If we take the analogy of an architect building a house, in the analysis stage he has determined the feasibility of the project and identified the owner's requirements in terms of the positioning of the house on the plot, size and architectural style, number of rooms and so on. The architect may even have built a small model to demonstrate the look and feel of the new dwelling. Once the owner is happy that the proposed house meets his requirements, the architect must communicate the detailed design to the builders. This would entail the drawing of a detailed plan of the house, specifying exactly how every part of the building is constructed and defining dimensions, materials, construction techniques etc.

We need to go through the same process when designing computer systems. This includes the design of:

- the technical platform on which the software will run. The new application may need new hardware, operating systems software and network connections
- output reports and enquiry screens
- input forms and data capture procedures
- physical file and database layouts
- description of the workings of each program module
- new clerical processes and procedures that will be required to interface with the new system.

Whereas in the analysis stage the emphasis was on identifying the user's needs with little concern for the way the computer would be used, the design stage also requires user involvement for the approval of detailed design, but the physical constraints imposed by the computer are also of major importance. Gane and Sarson [1979] define the objectives of structured design as follows:

> The most important objective of design, of course, is to deliver the functions required by the user. If the logical model calls for the production of pay cheques and the design does not produce pay cheques, or does not produce them correctly, then the design is a failure. But given that many correct designs are possible, there are three main objectives which the designer has to bear in mind while evolving and evaluating a design:
>
> - Performance. How fast the design will be able to do the user's work given a particular hardware resource.
> - Control. The extent to which the design is secure against human errors, machine malfunction, or deliberate mischief.
> - Changeability. The ease with which the design allows the system to be changed to, for example, meet the user's needs to have different transaction types processed.

The output from systems design is a detailed design specification incorporating technical, input, output, data and process specifications. In the past, much of the information was communicated in written form which was difficult to understand and often ambiguous. Imagine the builder having to construct a house from a written description. Like the output from analysis we have a number of innovative tools to help users and developers understand and communicate the workings of the system. These include data models and data dictionaries, screen and report layouts, structure charts and pseudo-code. We will look at most of these later in this chapter.

Systems Build

In this stage we program the new system. If the system has been purchased "off-the shelf, this phase would consist of the customization of the system. The success of the implementation stage is heavily reliant on the previous stages. If the analysis stage was poorly enacted, the system may not be what the user requires. Poor design will make it difficult for the programmer to construct the system, and it may be inefficient and difficult to maintain.

However, if the required effort and expertise is invested in analysis and design, there will be a precise specification of what to build available to the IS programmers and technical staff.

Unlike the previous stages, the programming stage can be undertaken as a number of separate tasks, performed in parallel. Programmers can code, data base administrators set up the database, hardware suppliers install and test networks and equipment, and we can begin to train the end-users to prepare them for the implementation phase. With so much happening, and with the need for some tasks to be completed before others begin, the analyst must develop a detailed project implementation plan to ensure tasks are scheduled and delays are quickly identified and addressed. Programming includes the following steps:

- database construction
- program coding and testing
- systems testing to check the system can handle expected loads and meets physical performance criteria
- finalise manual procedures

Systems Implementation

This entails the transition of the completed system into the operational environment, and includes the following tasks (some of which will already have been started in earlier phases):

- installation and testing of any new hardware, systems software and network infrastructure
- train users and operations staff
- transfer data (data conversion)from the manual or old system to the new database where necessary
- perform acceptance testing. This requires careful planning to ensure all data flows, error procedures, interfaces, controls and manual procedures are tested
- complete project documentation

The change over carries some risk, as failure of the new system may result in the organization being unable to do business, There are a number of approaches to converting from the old system to the new. The least risky is to run the new system in parallel with the old until the new system is stable. There is obviously a cost to running both systems. Another approach is to convert one part of the organization at a time (for example one branch office at a time). This method (known as the pilot method) reduces risk and allows the development team to focus on one area. This approach can cause some integration problems as part of the organization is running on the old system and part on the new. A similar approach is the phased implementation method where organizations convert to a large system one step at a time. For example they may start with the stock system, then implement debtors and finally the order entry system. Some organizations use the big bang approach and just switch over from the old to the new system. This option is obviously high risk as there is no system to fall back on in an emergency.

When the new system has been in operation for a few months, a post-implementation audit should be carried out. This audit must ascertain whether the project has achieved the initial objectives specified in terms of:

- meeting initial scope and objectives
- anticipated costs and benefits to the organization
- user satisfaction with the new system
- performance (response/turnaround time, reliability)
- adherence to standards
- quality of final product
- project performance in terms of cost and time.

This exercise will help to highlight problems that require maintenance of the new system and provide valuable feedback to IS management and project teams to assist in future development exercises.

Maintenance

Finally resources will be required to maintain and enhance the system over its operational life which can vary between 4 and 10 years. There is normally a formal hand-over of the system from the project team to the maintenance team. This will ensure that there is a defined time when the project is completed and that all the required documentation is produced. There are many systems in existence that are still supported by the original developer; and all knowledge of the system exists only in that individual’s head. The problem is that when this person leaves (or worse gets run over by a bus), there is no one with any knowledge of the system and the organization is at risk.

Research has shown that this is the most expensive stage of the life cycle as program bugs (as a result of poor design or bad coding and testing) or enhancements (poor analysis of user’s requirements or changes to the business) require continual analysis and programming effort.

The following table summarizes the important tasks in the six stages of the SDLC and highlights the main deliverables from each task.

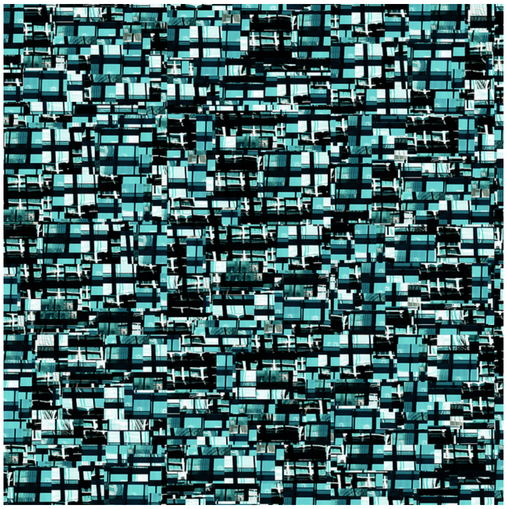
Figure 2: SDLC Tasks

Stage	Tasks	Deliverables

Preliminary Investigation	Problem Definition Scope and Objectives Data Gathering Risk Assessment Feasibility Analysis	Project Charter Feasibility Study
Systems Analysis	Data Gathering Systems Modeling User Requirements Definition	User Requirements Specification
Systems Design	Make or Buy Decision Physical Systems Design Technical Design	Detailed Systems Specification
Systems Build	Programming and testing Platform Implementation	Production System
Systems Implementation	User Training Data Conversion Systems Conversion Post-Implementation Review	Live System
Systems Maintenance	Fix system “bugs” System enhancement	Working System

Development of Structured Methodologies

New and innovative systems development techniques are frequently proposed by researchers and practitioners and, over time, these formal methods have replaced the traditional pragmatic approach to developing computer systems.



Structured Programming

In the 1960’s, the major concern in IS development environments was the efficient utilization of expensive computer hardware. Programs were written in low level languages with little or no support documentation and, over time, the code became almost impossible for maintenance programmers to understand and fix. So arose the need to introduce a set of standard rules and procedures for writing programs, often referred to as structured programming. Some of the key techniques in the structured programming approach include:

- a limited set of simple control structures (to control branching and looping)
- standard naming conventions for data and procedures
- self documenting programs.

Structured programming techniques ensured that programs were easier to write and test and much easier to maintain.

Structured Design

In the mid 1970s the focus in systems development moved from program coding to systems design. Computer applications were becoming more complex with the introduction of large on-line, integrated systems.

IS researchers, and in particular Larry Constantine, studied the problems of program size and complexity, and determined that, as a problem grew in size, so there was a more than proportional growth in the complexity of the problem and therefore in programming time. He advocated that all systems should be made up of small modules each no longer than fifty lines of program code.

Fragmenting a problem into a number of modules can be done in many ways; and he urged that the best technique would be to segment the program by function (task) with each module being as functionally independent of other modules as possible. This would ensure that changes to one program module were unlikely to affect other modules.

This technique was known as structured design; and a graphical representation of the modules and their relationships known as a structure chart, was developed to assist in the process.

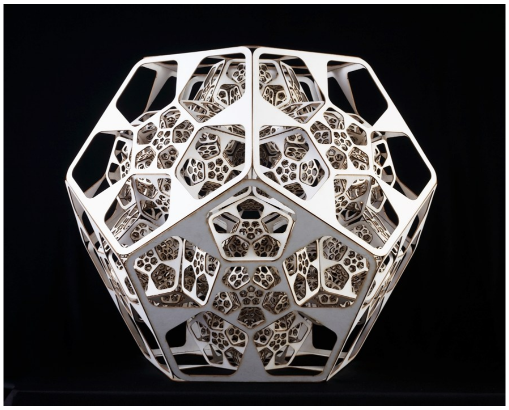
Structured Analysis

In the late 1970s the focus in systems development moved again, this time to the analysis stage. IS professionals had formalized the design and coding of computer software but had neglected the most important development issue – what are the user's real requirements. Written specifications were the main source of communication throughout the project. In the same way that architects would never attempt to describe a building in a letter, so analysts needed tools and techniques which could be used to define and communicate the user's requirements to the systems design stage. These structured tools and techniques included:

- Data Flow Diagrams (DFD). These diagrams are used to show how data flows between the various processes in the system. DFD's are an excellent communication tool as they are simple enough for users to understand and yet detailed enough to form the basis for the systems design process. A number of DFD techniques has been developed since the original work was published by Tom De Marco in 1978. However, they all basically perform the same task. Data flow diagrams are one of the most used and popular IS charting techniques.
- Entity Relationship Diagrams (ERD). Entity relationship diagrams identify the major objects about which data is stored and chart their interrelationship. Like most formal techniques, its major value is that it forces the analyst into a structured and detailed investigation of all the data used in the system.
- Decision Trees and Pseudo-code. These tools enable the analyst to express process logic clearly and unambiguously. In the detailed analysis of an information system, the analyst often has to describe a logical process that the future system will have to perform. Examples of these could be the way that a personnel system calculates pension benefits for employees or a sales system calculates sales commissions. Decision trees are diagrammatic representations of the process logic, showing the conditions that need to be tested and the resulting activities in a simple tree-like structure. Pseudo-code can be described as "structured English". It permits the analyst to define process logic using English language couched in the logical structures inherent in program code. In reality it eliminates the verbosity and ambiguity from the English narrative.
- Project Dictionary. This tool enables the analyst to capture and catalogue the entire system specification on computer with the obvious advantages in reporting, cross-referencing and updating. In a database environment, data is no longer the property of each individual application but managed centrally as a corporate resource. Vast amounts of information about this data needs to be maintained, for example field names, types and lengths, validation rules, data structures and relationships. As systems move from the development to production environment, this data about data is transferred from the project dictionary into a production data dictionary to enable the database administrator (DBA) to build and maintain the corporate database.

As we will discuss later, most of the new computer assisted software engineering (CASE) tools are now built up round a project dictionary.

Alternative approaches to developing systems



Over the past 40 years, efforts have been made to improve the quality of new systems and to reduce the time and effort expended in their development. The following section provides an overview of some of the significant tools and techniques developed for this purpose.

Prototyping

One technique that has been incorporated successfully into the SDLC is prototyping. As the name suggests a prototype is a mock-up or model of a system for review purposes.

Looking at the traditional SDLC, one of the major problems is that the user is asked to provide detailed requirements prior to the system being built. Once a system is implemented he may find flaws in his original requirements or may see the possibilities of a new and improved approach.

For applications such as general ledger and payroll, the requirements are normally well understood (and usually standardized enough to suggest the use of packages) but many other areas such as personnel are unstructured and would benefit from the prototyping stage.

The two main approaches to the use of prototypes in the SDLC are:

- discovery prototyping where the analyst builds a skeleton of the final product in the analysis stage of the project in order that the user may better understand the workings of the final system. This prototype is normally built with a fourth generation language and while it is likely to include mock-ups of screens and reports, it is seldom a fully working model. The building and refining of the prototype is an iterative process between analyst and user and stimulates discussion on the functionality of the final product. Once the analyst and user are happy that the system's requirements have been identified, detailed requirement specifications are developed and the prototype is no longer required. In some instances the prototype may serve as the specification.
- evolutionary prototyping where a working model is built with the intention of later refining it into the operational system. While this technique would appear to have great advantages in terms of productivity, the original prototype is often thrown together and not properly designed.

Prototyping can offer IS developers many advantages in that it assists in clarifying user's requirements, improves user communication and commitment, should improve the functionality and quality of the user interface and will assist in identifying problems earlier in the development life cycle.

Where prototyping can be problematic is that it raises the user's expectations that systems are quick to build and changes are easy. In addition there is a lack of experienced prototypers and quality prototyping tools.

Joint Application Development (JAD)

One major problem in systems development projects is the lack of real communication, understanding and consensus between users, management and the development team. Instead of the traditional one on one interviews spread over weeks and often months, the JAD approach involves a series of highly structured workshops where stakeholders focus on any one of the planning, analysis, design and implementation stages of the life cycle. One obvious advantage of this approach is a reduction in the time it takes to develop systems. However the real benefits of JAD come from better user requirements through improved communication and conflict resolution. Successful JAD sessions often depend on the competence of the session leader (termed the facilitator), the scribe (who is responsible for documenting the output from the sessions), strong top management support and a mix of participants with expertise and responsibility for the area under discussion.

Computer Assisted Software Engineering (CASE)

There is a famous saying, “The cobblers children have no shoes.” and this is very relevant to IS. Here we have a classic example of a group of IS professionals, dedicated to computerizing the organization in which they work, while developing these computer systems via manual means.

CASE environments attempt to address this problem by offering a set of integrated electronic tools for use in the SDLC. During the first phase of system development, CASE products provide the analyst with computerized tools to complete and document the analysis and detailed design stages of the development project by graphically modeling the data requirements and the business process flows that the intended application has to address. These models attempt to give a visual representation of a part of the business operations, following one of many modeling standards. The resultant application model is then used as a blueprint for the actual implementation in computer code. The tools that are geared specifically for this modeling phase are referred to as upper-CASE or JJ-CASE tools.

Lower-CASE tools specialize in the second phase of system development: the actual generation of executable applications or advanced prototypes. This is typically achieved through the use of a generic application generator, although CASE tools tend to be independent of any specific database management system.

Integrated or I-CASE aims to automate both phases i.e. a combination of upper and lower-CASE tools in one single package.

Most CASE environments use an electronic project dictionary as a repository and can include:

- graphic tools for charting diagrams such as DFD’s, ERD’s and Structure charts
- 4th generation languages or application generators to assist with prototyping
- data dictionary facilities to record and maintain data about data, processes and other system details
- quality control facilities to check specifications and code for correctness
- code generators to reduce programming effort
- spreadsheet models to assist with cost/benefit analysis
- project management tools to plan, monitor and control the development cycle.
- When CASE environments were originally developed in the mid 1980s, IS managers viewed them as a possible “silver bullet” to resolve the growing demand for computer systems. The promise of CASE was better quality systems, reduced development time, enforced standards and improved documentation.

As yet CASE tools have failed to make a major impact. CASE environments are complex, the cost of implementing CASE is high (both in terms of the CASE software and analyst training) and many organizations are looking to other solutions (packages and object orientation) to resolve their applications backlog.

Object Oriented Development (OOD)

Using the traditional development approach where the analyst designs procedures focused on the user’s requirements can result in systems that are costly to develop and inflexible in nature. The object oriented approach attempts to build components (objects) that model the behavior of persons, places, things or concepts that exist in the real world and then to build systems from these components. We do not design and build unique systems for motor cars or televisions; they are mostly built up from a set of common, interchangeable components. Even when components are unique they are very similar to other components. In the same way we can construct computer systems from building blocks reusing objects from other systems, making modifications to similar objects or obtaining objects from commercial component libraries.

The OOD paradigm is only now gaining momentum in the market place and most programming languages and methodologies do not support 00 development. One exception is Small Talk, the language credited with pioneering the OOP (object oriented programming) concept. Today many of the popular programming languages are appearing with 00 versions (for example Pascal, C++, Visual Basic and even COBOL).

Other development tools

The above-mentioned new programming approaches were not the only attempts to improve developer productivity. The following presents some other development tools and approaches. Note that, since distinctions between the categories cannot always be perfect, some tools could be classified in more than one category.

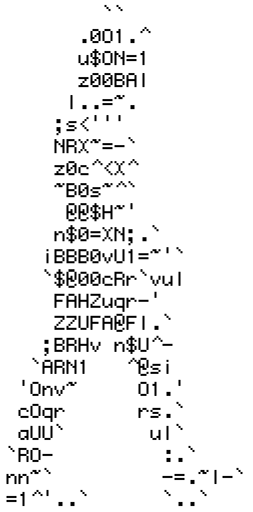
- Visual programming tools. The power of graphical user interfaces and object-orientation has spawned a number of high-level front-ends or shells to enable non-programmers to generate their own straightforward applications. These visual programming tools allow for the construction of applications by selecting, connecting, copying and arranging programming objects. For simple applications, there is no need for any code to be written at all since all required objects can be copied from a large library with all commonly used, pre-configured objects and their associated standard methods.
- Report generators are generally associated with database management systems and allow users to create ad-hoc, customized reports using the data in the database by specifying the various selection criteria and the desired report layout.
- Application generators consist of standard building blocks that can be combined or customized to create the required systems. The user specifies the inputs, the output requirements and the various data validations and transformations. Screen and report painters allow on-line, visual layout of input and output modules. Generally, these generators are supported by a comprehensive database management system that integrates the data dictionary, graphics and reporting modules as well as other utilities such as data and process modeling, security facilities, decision support modules and query-by-example (QBE) languages.
- Logic programming for knowledge based systems. Programmers quickly discovered that conventional programming languages were inadequate to develop advanced knowledge-based applications, such as expert systems and other artificial intelligence systems. These systems require reasoning capabilities and have knowledge representation requirements that are difficult to implement using procedural languages. This led to the development of logic programming languages such as LISP and Prolog. The use of these languages is generally confined to researchers and scientists. Today, a number of shells has been developed that allow the automatic generation of straight-forward knowledge-based systems.
- End-user applications. Today's end-user productivity applications have extensive programming capabilities and allow for customization by means of macros (pre-recorded sequences of commands, keystrokes) and formulae. A spreadsheet is essentially a model developed by an end-user whereby the equations (or data transformations) have been programmed by means of formulae. Many of these formulae look similar to the statements in programming languages. The following statement would calculate someone's weekly wages taking into account an overtime rate of 50%, using Microsoft Excel or Access.

  – IF (hours > 42 , hours * wage , 42 * wage + (1.5 * wage) * (hours – 42))

Critical success factors

Regardless of the development approach that may be used, a number of factors have been identified that are critical to ensuring the success of a systems development project:

- well-defined system objectives
- careful test of feasibility
- top management support
- user involvement to ensure strong commitment
- rigorous analysis to ensure detailed, unambiguous user requirements
- sound detailed design to ensure an efficient, quality, maintainable system
- project management to ensure the development team is managed and controlled.



South African Perspective

Research by a group of Scandinavian computer scientists has suggested that prototyping is better suited than the traditional SDLC to systems development projects undertaken in developing countries. Although the SDLC provides a rigorous development methodology intended to generate clearly defined system requirements, it does not take into account problems resulting from social and cultural factors. These include the uncertain availability of technical skills, user anxiety about technology, and the need for adaptability of the final product to differing local conditions. A further dimension to this approach is the need to train users in basic computer literacy skills and to inform them about the business role of IS before any attempt is made to elicit system requirements. Once workers have been empowered in this way, they are able to provide more valuable participation in the development of a system. Developers must also see the project as a mutual learning experience, since they need to understand how future changes in the business environment may affect system requirements.

Beyond the Basics

Web pages can contain multimedia effects and interactive capabilities, and the development of web pages involves using a variety of special components. Among the jargon and acronyms that you may encounter are the following:

- Hypertext Markup Language (HTML) is not actually a programming language, but has specific rules for defining the formatting of text, graphics, video and audio on a web page. Tags are used to indicate how a page should be displayed on your screen, and the details underlying links to other web pages.
- Interactive elements such as scrolling messages, pop-up windows and animated graphics are controlled by small programs, generally scripts, applets or ActiveX controls. Basically, a script is an interpreted program that runs on the client computer, as opposed to an applet, which is a compiled program running on the client computer. ActiveX controls are object-oriented technologies that allow components on a network to communicate with one another.
- Information is sent and received between your computer and a web server via the common gateway interface (CGI), which is the communications standard that defines how a web server communicates with outside sources such as a database. CGI programs are frequently written using scripting languages such as JavaScript, which is simpler to use than the full Java language.
- Web pages created using dynamic HTML can automatically update their content on the client’s machine, without having to access the web server, making them more responsive to user interaction. Extensible HTML (XHTML) uses XML technology to display web pages on different types of display devices, while wireless markup language (WML) supports browsing on PDAs and cellular telephones. Finally wireless application protocol (WAP) specifies how wireless devices such as cellular telephones communicate with the Web.

Exercises

Stages of the SDLC

Read the following article, and then briefly explain, for each stage of the SDLC, which of the standard activities appear to have been omitted or not completed when developing the system, and what effect this had on the quality of the final product.

From: Machlis, S. "U.S. Agency Puts $71m System on Ice", Computerworld, 12 May 1997.

The U.S Agency for International Development (AID) last week confirmed that it suspended overseas use of a new computer system plagued by integration snafus, data transmission bottlenecks, and response times so slow that employee efficiency suffered. For now, 39 field sites will go back to using the agency's old system for core accounting services and procurement contracts while problems with the Washington-based computers are ironed out. "We need to get the core functionality established", said Richard McCall, AID's chief of staff.

The New Management System (NMS), budgeted at $71 million, has been under fire since it was deployed in October of 1997. The AID inspector general's office criticized NMS for data errors and slow performance. In some cases, users had to spend days trying to process a single transaction.

The new system can't handle the large amount of data that passes among AID offices, McCall said. The agency must decide whether to boost expensive satellite network bandwidth to handle real time transactions or move to some batch processing. "I don't think people understood the amount of data that would be transmitted over the system", he said.

Designers also initially failed to gasp the difficulty of integrating legacy accounting systems. "We thought we had three primary accounting systems," McCall said. But numerous infield alterations to basic systems over the years meant the agency had closer to 80 different accounting systems. Some of the resulting data didn't import correctly into the new system.

In addition, McCall said, system designers should have stayed focused on core requirements instead of trying to immediately add features that users requested after early tests. For example, some overseas employees wanted to be able to call up data from any foreign site. Although that is an attractive feature, he said, "that taxes the system. You don't really need that now."

Systems conversion

Four different methods have been described for the conversion to a new system: the parallel, pilot, phased or "big bang" approach. Compare these four alternatives in terms of the likely time frame involved, level of risk incurred, and user "buy-in" to the new system.

Licensing & Attributions

CC licensed content, Shared previously

- Discovering Information Systems Chapter 12. **Authored by:** Jean-Paul Van Belle, Mike Eccles, and Jane Nash. **Located at:** https://archive.org/details/ost-computer-science-discoveringinformationsystems. **License:** *CC BY-NC-ND: Attribution-NonCommercial-NoDerivatives*
- Layers of Spirals: Version 2. **Authored by:** Ken. **Located at:** https://www.flickr.com/photos/fractal_ken/3407769130/. **License:** *CC BY: Attribution*
- STRUCTURES // 1. **Authored by:** Nick Hughs. **Located at:** https://www.flickr.com/photos/enhues/7735176726/. **License:** *CC BY: Attribution*
- Foldable Fractal 2.0. **Authored by:** Sanchtv. **Located at:** https://www.flickr.com/photos/sanchtv/4192677571/. **License:** *CC BY-NC-ND: Attribution-NonCommercial-NoDerivatives*
- Walking Ascii-man gif. **Authored by:** OC Roy. **Located at:** https://www.flickr.com/photos/offshore/14993958/. **License:** *CC BY-NC-ND: Attribution-NonCommercial-NoDerivatives*

READING: INTELLECTUAL PROPERTY

Introduction

Intellectual property (**IP**) is a term referring to creations of the intellect for which a monopoly is assigned to designated owners by law. Some common types of intellectual property rights (IPR) are copyright, patents, and industrial design rights; and the rights that protect trademarks, trade dress, and in some jurisdictions trade secrets: all these cover music, literature, and other artistic works; discoveries and inventions; and words, phrases, symbols, and designs. Intellectual property rights are themselves a form of property, called intangible property.

World Intellectual Property Organization (WIPO) in Geneva, Switzerland.

Although many of the legal principles governing IP and IPR have evolved over centuries, it was not until the 19th century that the term *intellectual property* began to be used, and not until the late 20th century that it became commonplace in the majority of the world. The Statute of Monopolies (1624) and the British Statute of Anne (1710) are now seen as the origins of patent law and copyright respectively, firmly establishing the concept of intellectual property.

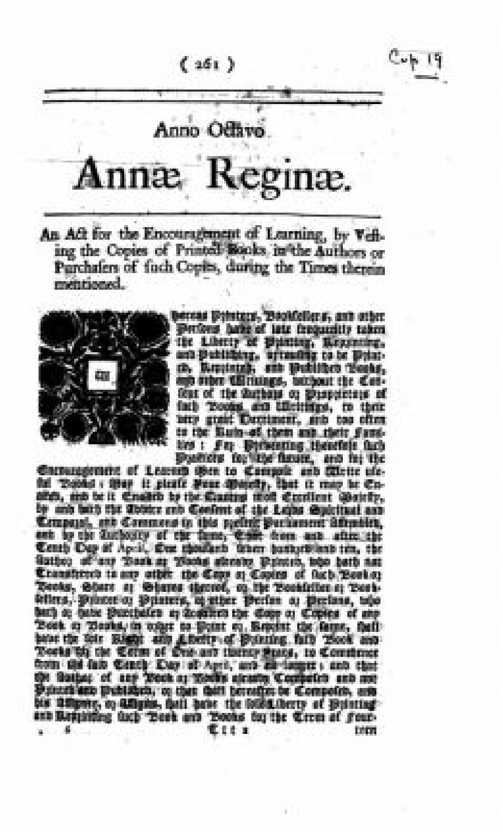
History

The first known use of the term *intellectual property* dates to 1769, when a piece published in the *Monthly Review* used the phrase. The first clear example of modern usage goes back as early as 1808, when it was used as a heading title in a collection of essays.

The German equivalent was used with the founding of the North German Confederation whose constitution granted legislative power over the protection of intellectual property (*Schutz des geistigen Eigentums*) to the confederation. When the administrative secretariats established by the Paris Convention (1883) and the Berne Convention (1886) merged in 1893, they located in Berne, and also adopted the term intellectual property in their new combined title, the United International Bureaux for the Protection of Intellectual Property.

The organization subsequently relocated to Geneva in 1960, and was succeeded in 1967 with the establishment of the World Intellectual Property Organization (WIPO) by treaty as an agency of theUnited Nations. According to Lemley, it was only at this point that the term really began to be used in the United States (which had not been a party to the Berne Convention), and it did not enter popular usage until passage of the Bayh-Dole Act in 1980.

> The history of patents does not begin with inventions, but rather with royal grants by Queen Elizabeth I (1558–1603) for monopoly privileges… Approximately 200 years after the end of Elizabeth's reign, however, a patent represents a legal right obtained by an inventor providing for exclusive control over the production and sale of his mechanical or scientific invention… [demonstrating] the evolution of patents from royal prerogative to common-law doctrine.



(261)   Cap 19

Anno Octavo

Annæ Reginæ.

An Act for the Encouragement of Learning, by Veſting the Copies of Printed Books in the Authors or Purchaſers of ſuch Copies, during the Times therein mentioned.

Whereas Printers, Bookſellers, and other Perſons have of late frequently taken the Liberty of Printing, Reprinting, and Publiſhing, or cauſing to be Printed, Reprinted, and Publiſhed Books, and other Writings, without the Conſent of the Authors or Proprietors of ſuch Books and Writings, to their very great Detriment, and too often to the Ruin of them and their Families: For Preventing therefore ſuch Practices for the future, and for the Encouragement of Learned Men to Compoſe and Write uſeful Books; May it pleaſe Your Majeſty, that it may be Enacted, and be it Enacted by the Queens moſt Excellent Majeſty, by and with the Advice and Conſent of the Lords Spiritual and Temporal, and Commons in this preſent Parliament Aſſembled, and by the Authority of the ſame, That from and after the Tenth Day of April, One thouſand ſeven hundred and ten, the Author of any Book or Books already Printed, who hath not Transferred to any other the Copy or Copies of ſuch Book or Books, Share or Shares thereof, or the Bookſeller or Bookſellers, Printer or Printers, or other Perſon or Perſons, who hath or have Purchaſed or Acquired the Copy or Copies of any Book or Books, in order to Print or Reprint the ſame, ſhall have the ſole Right and Liberty of Printing ſuch Book and Books for the Term of One and twenty Years, to Commence from the ſaid Tenth Day of April, and no longer; and that the Author of any Book or Books already Compoſed and not Printed and Publiſhed, or that ſhall hereafter be Compoſed, and his Aſſignee, or Aſſigns, ſhall have the ſole Liberty of Printing and Reprinting ſuch Book and Books for the Term of Four-

Ttt 2   teen

The Statute of Anne came into force in 1710

The term can be found used in an October 1845 Massachusetts Circuit Court ruling in the patent case *Davoll et al. v. Brown.*, in which Justice Charles L. Woodbury wrote that "only in this way can we protect intellectual property, the labors of the mind, productions and interests are as much a man's own…as the wheat he cultivates, or the flocks he rears." The statement that "discoveries are…property" goes back earlier. Section 1 of the French law of 1791 stated, "All new discoveries are the property of the author; to assure the inventor the property and temporary enjoyment of his discovery, there shall be delivered to him a patent for five, ten or fifteen years." In Europe, French author A. Nion mentioned *propriété intellectuelle* in his *Droits civils des auteurs, artistes et inventeurs*, published in 1846.

Until recently, the purpose of intellectual property law was to give as little protection possible in order to encourage innovation. Historically, therefore, they were granted only when they were necessary to encourage invention, limited in time and scope.

The concept's origins can potentially be traced back further. Jewish law includes several considerations whose effects are similar to those of modern intellectual property laws, though the notion of intellectual creations as property does not seem to exist—notably the principle of Hasagat Ge'vul (unfair encroachment) was used to justify limited-term publisher (but not author) copyright in the 16th century. In 500 BCE, the government of the Greek state of Sybaris offered one year's patent "to all who should discover any new refinement in luxury."

Intellectual property rights

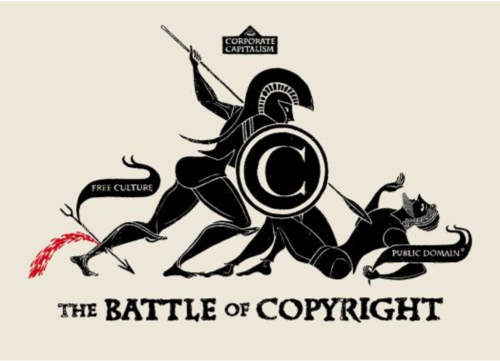
Intellectual property rights include patents, copyright, industrial design rights, trademarks, plant variety rights, trade dress, and in some jurisdictions trade secrets. There are also more specialized or derived varieties of *sui generis* exclusive rights, such as circuit design rights (called mask work rights in the US) and supplementary protection certificates for pharmaceutical products (after expiry of a patent protecting them) and database rights (in European law).

Patents

A patent is a form of right granted by the government to an inventor, giving the owner the right to exclude others from making, using, selling, offering to sell, and importing an invention for a limited period of time, in exchange for the public disclosure of the invention. An invention is a solution to a specific technological problem, which may be a product or a process and generally has to fulfill three main requirements: it has to be new, not obvious and there needs to be an industrial applicability.

Copyright

A copyright gives the creator of an original work exclusive rights to it, usually for a limited time. Copyright may apply to a wide range of creative, intellectual, or artistic forms, or “works.” Copyright does not cover ideas and information themselves, only the form or manner in which they are expressed.



Industrial design rights

An industrial design right (sometimes called “design right”) protects the visual design of objects that are not purely utilitarian. An industrial design consists of the creation of a shape, configuration or composition of pattern or color, or combination of pattern and color in three-dimensional form containing aesthetic value. An industrial design can be a two- or three-dimensional pattern used to produce a product, industrial commodity or handicraft.

Plant varieties

Plant breeders’ rights or plant variety rights are the rights to commercially use a new variety of a plant. The variety must amongst others be novel and distinct and for registration the evaluation of propagating material of the variety is examined.

Trademarks

A trademark is a recognizable sign, design or expression which distinguishes products or services of a particular trader from the similar products or services of other traders.

Trade dress

Trade dress is a legal term of art that generally refers to characteristics of the visual appearance of a product or its packaging (or even the design of a building) that signify the source of the product to consumers.

Trade secrets

A trade secret is a formula, practice, process, design, instrument, pattern, or compilation of information which is not generally known or reasonably ascertainable, by which a business can obtain an economic advantage over competitors or customers.

Objectives of intellectual property law

The stated objective of most intellectual property law (with the exception of trademarks) is to “Promote progress.” By exchanging limited exclusive rights for disclosure of inventions and creative works, society and the patentee/copyright owner mutually benefit, and an incentive is created for inventors and authors to create and disclose their work. Some commentators have noted that the objective of intellectual property legislators and those who support its implementation appears to be “absolute protection”. “If some intellectual property is desirable because it encourages innovation, they reason, more is better. The thinking is that creators will not have sufficient incentive to invent unless they are legally entitled to capture the full social value of their inventions.” This absolute protection or full value view treats intellectual property as another type of “real” property, typically adopting its law and rhetoric. Other recent developments in intellectual property law, such as the America Invents Act, stress international harmonization.

Financial incentive

These exclusive rights allow owners of intellectual property to benefit from the property they have created, providing a financial incentive for the creation of an investment in intellectual property, and, in case of patents, pay associated research and development costs. Some commentators, such as David Levine and Michele Boldrin, dispute this justification.

In 2013 the United States Patent & Trademark Office approximated that the worth of intellectual property to the U.S. economy is more than US$5 trillion and creates employment for an estimated 18 million American people. The value of intellectual property is considered similarly high in other developed nations, such as those in the European Union. In the UK, IP has become a recognized asset class for use in pension-led funding and other types of business finance. However, in 2013, the UK Intellectual Property Office stated: “There are millions of intangible business assets whose value is either not being leveraged at all, or only being leveraged inadvertently”.

Economic growth

The WIPO treaty and several related international agreements underline that the protection of intellectual property rights is essential to maintaining economic growth. The *WIPO Intellectual Property Handbook* gives two reasons for intellectual property laws:

> One is to give statutory expression to the moral and economic rights of creators in their creations and the rights of the public in access to those creations. The second is to promote, as a deliberate act of Government policy, creativity and the dissemination and application of its results and to encourage fair trading which would contribute to economic and social development.

The Anti-Counterfeiting Trade Agreement (ACTA) states that “effective enforcement of intellectual property rights is critical to sustaining economic growth across all industries and globally.”

Economists estimate that two-thirds of the value of large businesses in the United States can be traced to intangible assets. “IP-intensive industries” are estimated to generate 72 percent more value added (price minus material cost) per employee than “non-IP-intensive industries.”

A joint research project of the WIPO and the United Nations University measuring the impact of IP systems on six Asian countries found “a positive correlation between the strengthening of the IP system and subsequent economic growth.”

Economists have also shown that IP can be a disincentive to innovation when that innovation is drastic. IP makes excludable non-rival intellectual products that were previously non-excludable. This creates economic inefficiency as long as the monopoly is held. A disincentive to direct resources toward innovation can occur when monopoly profits are less than the overall welfare improvement to society. This situation can be seen as a market failure, and an issue of appropribility.

Morality

According to Article 27 of the Universal Declaration of Human Rights, "everyone has the right to the protection of the moral and material interests resulting from any scientific, literary or artistic production of which he is the author." Although the relationship between intellectual property and human rights is a complex one, there are moral arguments for intellectual property.

The arguments that justify intellectual property fall into three major categories. Personality theorists believe intellectual property is an extension of an individual. Utilitarians believe that intellectual property stimulates social progress and pushes people to further innovation. Lockeans argue that intellectual property is justified based on deservedness and hard work.

Various moral justifications for private property can be used to argue in favor of the morality of intellectual property, such as:

1. *Natural Rights/Justice Argument*: this argument is based on Locke's idea that a person has a natural right over the labour and/or products which is produced by his/her body. Appropriating these products is viewed as unjust. Although Locke had never explicitly stated that natural right applied to products of the mind, it is possible to apply his argument to intellectual property rights, in which it would be unjust for people to misuse another's ideas. Locke's argument for intellectual property is based upon the idea that laborers have the right to control that which they create. They argue that we own our bodies which are the laborers, this right of ownership extends to what we create. Thus, intellectual property ensures this right when it comes to production.
2. *Utilitarian-Pragmatic Argument*: according to this rationale, a society that protects private property is more effective and prosperous than societies that do not. Innovation and invention in 19th century America has been attributed to the development of the patent system. By providing innovators with "durable and tangible return on their investment of time, labor, and other resources", intellectual property rights seek to maximize social utility. The presumption is that they promote public welfare by encouraging the "creation, production, and distribution of intellectual works". Utilitarians argue that without intellectual property there would be a lack of incentive to produce new ideas. Systems of protection such as Intellectual property optimize social utility.
3. *"Personality" Argument*: this argument is based on a quote from Hegel: "Every man has the right to turn his will upon a thing or make the thing an object of his will, that is to say, to set aside the mere thing and recreate it as his own." European intellectual property law is shaped by this notion that ideas are an "extension of oneself and of one's personality." Personality theorists argue that by being a creator of something one is inherently at risk and vulnerable for having their ideas and designs stolen and/or altered. Intellectual property protects these moral claims that have to do with personality.

Lysander Spooner (1855) argues "that a man has a natural and absolute right—and if a natural and absolute, then necessarily a perpetual, right—of property, in the ideas, of which he is the discoverer or creator; that his right of property, in ideas, is intrinsically the same as, and stands on identically the same grounds with, his right of property in material things; that no distinction, of principle, exists between the two cases."

Writer Ayn Rand argued in her book *Capitalism: The Unknown Ideal* that the protection of intellectual property is essentially a moral issue. The belief is that the human mind itself is the source of wealth and survival and that all property at its base is intellectual property. To violate intellectual property is therefore no different morally than violating other property rights which compromises the very processes of survival and therefore constitutes an immoral act.

Infringement, misappropriation, and enforcement

Violation of intellectual property rights, called “infringement” with respect to patents, copyright, and trademarks, and “misappropriation” with respect to trade secrets, may be a breach of civil law or criminal law, depending on the type of intellectual property involved, jurisdiction, and the nature of the action.

As of 2011 trade in counterfeit copyrighted and trademarked works was a $600 billion industry worldwide and accounted for 5–7% of global trade.

Patent infringement

Patent infringement typically is caused by using or selling a patented invention without permission from the patent holder. The scope of the patented invention or the extent of protection is defined in the claims of the granted patent. There is safe harbor in many jurisdictions to use a patented invention for research. This safe harbor does not exist in the US unless the research is done for purely philosophical purposes, or in order to gather data in order to prepare an application for regulatory approval of a drug. In general, patent infringement cases are handled under civil law (e.g., in the United States) but several jurisdictions incorporate infringement in criminal law also (for example, Argentina, China, France, Japan, Russia, South Korea).

Copyright infringement

Copyright infringement is reproducing, distributing, displaying or performing a work, or to make derivative works, without permission from the copyright holder, which is typically a publisher or other business representing or assigned by the work’s creator. It is often called “piracy.” While copyright is created the instance a work is fixed, generally the copyright holder can only get money damages if the owner registers the copyright. Enforcement of copyright is generally the responsibility of the copyright holder. The ACTA trade agreement, signed in May 2011 by the United States, Japan, Switzerland, and the EU, and which has not entered into force, requires that its parties add criminal penalties, including incarceration and fines, for copyright and trademark infringement, and obligated the parties to active police for infringement. There are limitations and exceptions to copyright, allowing limited use of copyrighted works, which does not constitute infringement. Examples of such doctrines are the fair use and fair dealing doctrine.

Trademark infringement

Trademark infringement occurs when one party uses a trademark that is identical or confusingly similar to a trademark owned by another party, in relation to products or services which are identical or similar to the products or services of the other party. In many countries, a trademark receives protection without registration, but registering a trademark provides legal advantages for enforcement. Infringement can be addressed by civil litigation and, in several jurisdictions, under criminal law.

Trade secret misappropriation

Trade secret misappropriation is different from violations of other intellectual property laws, since by definition trade secrets are secret, while patents and registered copyrights and trademarks are publicly available. In the United States, trade secrets are protected under state law, and states have nearly universally adopted the Uniform Trade Secrets Act. The United States also has federal law in the form of theEconomic Espionage Act of 1996 (18 U.S.C. §§ 1831–1839), which makes the theft or misappropriation of a trade secret a federal crime. This law contains two provisions criminalizing two sorts of activity. The first, 18 U.S.C. § 1831(a), criminalizes the theft of trade secrets to benefit foreign powers. The second, 18 U.S.C. § 1832, criminalizes their theft for commercial or economic purposes. (The statutory penalties are different for the two offenses.) In Commonwealth common law jurisdictions, confidentiality and trade secrets are regarded as an equitable right rather than a property right but penalties for theft are roughly the same as the United States.

Criticisms

The term “intellectual property”

Criticism of the term *intellectual property* ranges from discussing its vagueness and abstract overreach to direct contention to the semantic validity of using words like *property* in fashions that contradict practice and law. Many detractors think this term specially serves the doctrinal agenda of parties opposing reform or otherwise abusing related legislations; for instance, by associating one view with certain attitude, or disallowing intelligent discussion about specific and often unrelated aspects of copyright, patents, trademarks, etc.

Demonstration in Sweden in support of file sharing, 2006.

Free Software Foundation founder Richard Stallman argues that, although the term *intellectual property* is in wide use, it should be rejected altogether, because it “systematically distorts and confuses these issues, and its use was and is promoted by those who gain from this confusion”. He claims that the term “operates as a catch-all to lump together disparate laws [which] originated separately, evolved differently, cover different activities, have different rules, and raise different public policy issues” and that it creates a “bias” by confusing these monopolies with ownership of limited physical things, likening them to “property rights.” Stallman advocates referring to copyrights, patents and trademarks in the singular and warns against abstracting disparate laws into a collective term.

“Copying is not theft!” badge with a character resembling Mickey Mouse in reference to the in popular culture rationale behind the Sonny Bono Copyright Term Extension Act of 1998

Similarly, economists Boldrin and Levine prefer to use the term “intellectual monopoly” as a more appropriate and clear definition of the concept, which they argue, is very dissimilar from property rights.

Law professor, writer and political activist Lawrence Lessig, along with many other copyleft and free software activists, has criticized the implied analogy with physical property (like land or an automobile). They argue such an analogy fails because physical property is generally rivalrous while intellectual works are non-rivalrous (that is, if one makes a copy of a work, the enjoyment of the copy does not prevent enjoyment of the original). Other arguments along these lines claim that unlike the situation with tangible property, there is no natural scarcity of a particular idea or information: once it exists at all, it can be re-used and duplicated indefinitely without such re-use diminishing the original. Stephan Kinsella has objected to *intellectual property* on the grounds that the word “property” implies scarcity, which may not be applicable to ideas.

Entrepreneur and politician Rickard Falkvinge and hacker Alexandre Oliva have independently compared George Orwell’s fictional dialect Newspeak to the terminology used by intellectual property supporters as a linguistic weapon to shape public opinion regarding copyright debate and DRM.

Alternative terms

In civil law jurisdictions, intellectual property has often been referred to as intellectual rights, traditionally a somewhat broader concept that has included moral rights and other personal protections that cannot be bought or sold. Use of the term *intellectual rights* has declined since the early 1980s, as use of the term *intellectual property* has increased.

Alternative terms *monopolies on information* and *intellectual monopoly* have emerged among those who argue against the “property” or “intellect” or “rights” assumptions, notably Richard Stallman. The backronyms *intellectual* *protectionism* and *intellectual poverty*, whose initials are also *IP*, have found supporters as well, especially among those who have used the backronym *digital restrictions management*.

The argument that an intellectual property right should (in the interests of better balancing of relevant private and public interests) be termed an *intellectual monopoly privilege* (IMP) has been advanced by several academics including Birgitte Andersen and Thomas Alured Faunce.

Objections to overboard intellectual property laws

Some critics of intellectual property, such as those in the free culture movement, point at intellectual monopolies as harming health (in the case of pharmaceutical patents), preventing progress, and benefiting concentrated interests to the detriment of the masses, and argue that the public interest is harmed by ever-expansive monopolies in the form of copyright extensions,software patents, and business method patents. More recently scientists and engineers are expressing concern that patent thickets are undermining technological development even in high-tech fields like nanotechnology.

The free culture movement champions the production of content that bears little or no restrictions, like Wikipedia.

Petra Moser has asserted that historical analysis suggests that intellectual property laws may harm innovation:

> Overall, the weight of the existing historical evidence suggests that patent policies, which grant strong intellectual property rights to early generations of inventors, may discourage innovation. On the contrary, policies that encourage the diffusion of ideas and modify patent laws to facilitate entry and encourage competition may be an effective mechanism to encourage innovation.

Peter Drahos notes, “Property rights confer authority over resources. When authority is granted to the few over resources on which many depend, the few gain power over the goals of the many. This has consequences for both political and economic freedoms with in a society.”

The World Intellectual Property Organization (WIPO) recognizes that conflicts may exist between the respect for and implementation of current intellectual property systems and other human rights. In 2001 the UN Committee on Economic, Social and Cultural Rights issued a document called “Human rights and intellectual property” that argued that intellectual property tends to be governed by economic goals when it should be viewed primarily as a social product; in order to serve human well-being, intellectual property systems must respect and conform to human rights laws. According to the Committee, when systems fail to do so they risk infringing upon the human right to food and health, and to cultural participation and scientific benefits. In 2004 the General Assembly of WIPO adopted *The Geneva Declaration on the Future of the World Intellectual Property Organization* which argues that WIPO should “focus more on the needs of developing countries, and to view IP as one of many tools for development—not as an end in itself.”

Further along these lines, The ethical problems brought up by IP rights are most pertinent when it is socially valuable goods like life-saving medicines are given IP protection. While the application of IP rights can allow companies to charge higher than the marginal cost of production in order to recoup the costs of research and development, the price may exclude from the market anyone who cannot afford the cost of the product, in this case a life-saving drug.”An IPR driven regime is therefore not a regime that is conductive to the investment of R&D of products that are socially valuable to predominately poor populations.”

Some libertarian critics of intellectual property have argued that allowing property rights in ideas and information creates artificial scarcity and infringes on the right to own tangible property. Stephan Kinsella uses the following scenario to argue this point:

> [I]magine the time when men lived in caves. One bright guy—let's call him Galt-Magnon—decides to build a log cabin on an open field, near his crops. To be sure, this is a good idea, and others notice it. They naturally imitate Galt-Magnon, and they start building their own cabins. But the first man to invent a house, according to IP advocates, would have a right to prevent others from building houses on their own land, with their own logs, or to charge them a fee if they do build houses. It is plain that the innovator in these examples becomes a partial owner of the tangible property (e.g., land and logs) of others, due not to first occupation and use of that property (for it is already owned), but due to his coming up with an idea. Clearly, this rule flies in the face of the first-user homesteading rule, arbitrarily and groundlessly overriding the very homesteading rule that is at the foundation of all property rights.

Thomas Jefferson once said in a letter to Isaac McPherson on August 13, 1813:

> If nature has made any one thing less susceptible than all others of exclusive property, it is the action of the thinking power called an idea, which an individual may exclusively possess as long as he keeps it to himself; but the moment it is divulged, it forces itself into the possession of every one, and the receiver cannot dispossess himself of it. Its peculiar character, too, is that no one possesses the less, because every other possesses the whole of it. He who receives an idea from me, receives instruction himself without lessening mine; as he who lights his taper at mine, receives light without darkening me.

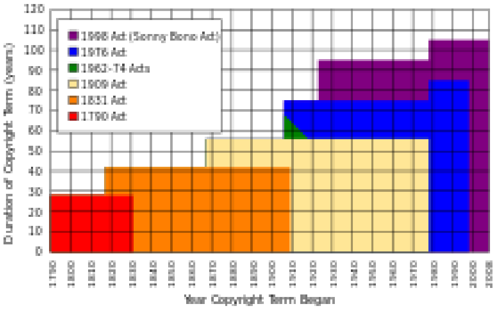
In 2005 the RSA launched the Adelphi Charter, aimed at creating an international policy statement to frame how governments should make balanced intellectual property law.

Another limitation of current U.S. Intellectual Property legislation is its focus on individual and joint works; thus, copyright protection can only be obtained in 'original' works of authorship. This definition excludes any works that are the result of community creativity, for example Native American songs and stories; current legislation does not recognize the uniqueness of indigenous cultural "property" and its ever-changing nature. Simply asking native cultures to 'write down' their cultural artifacts on tangible mediums ignores their necessary orality and enforces a Western bias of the written form as more authoritative.

Expansion in nature and scope of intellectual property laws

Other criticism of intellectual property law concerns the expansion of intellectual property, both in duration and in scope.

In addition, as scientific knowledge has expanded and allowed new industries to arise in fields such as biotechnology and nanotechnology, originators of technology have sought IP protection for the new technologies. Patents have been granted for living organisms, (and in the United States, certain living organisms have been patentable for over a century).



Expansion of U.S. copyright law (Assuming authors create their works by age 35 and live for seventy years)

The increase in terms of protection is particularly seen in relation to copyright, which has recently been the subject of serial extensions in the United States and in Europe. With no need for registration or copyright notices, this is thought to have led to an increase in orphan works(copyrighted works for which the copyright owner cannot be contacted), a problem that has been noticed and addressed by governmental bodies around the world.

Also with respect to copyright, the American film industry helped to change the social construct of intellectual property via its trade organization, the Motion Picture Association of America. In amicus briefs in important cases, in lobbying before Congress, and in its statements to the public, the MPAA has advocated strong protection of intellectual-property rights. In framing its presentations, the association has claimed that people are entitled to the property that is produced by their labor. Additionally Congress's awareness of the position of the United States as the world's largest producer of films has made it convenient to expand the conception of intellectual property. These doctrinal reforms have further strengthened the industry, lending the MPAA even more power and authority.

RIAA representative Hilary Rosen testifies before the Senate Judiciary Committee on the future of digital music (July 11, 2000)

The growth of the Internet, and particularly distributed search engines like Kazaa and Gnutella, have represented a challenge for copyright policy. The Recording Industry Association of America, in particular, has been on the front lines of the fight against copyright infringement, which the industry calls “piracy”. The industry has had victories against some services, including a highly publicized case against the file-sharing company Napster, and some people have been prosecuted for sharing files in violation of copyright. The electronic age has seen an increase in the attempt to use software-based digital rights management tools to restrict the copying and use of digitally based works. Laws such as the Digital Millennium Copyright Act have been enacted, that use criminal law to prevent any circumvention of software used to enforce digital rights management systems. Equivalent provisions, to prevent circumvention of copyright protection have existed in EU for some time, and are being expanded in, for example, Article 6 and 7 the Copyright Directive. Other examples are Article 7 of the Software Directive of 1991 (91/250/EEC), and the Conditional Access Directive of 1998 (98/84/EEC). This can hinder legal uses, affecting public domain works, limitations and exceptions to copyright, or uses allowed by the copyright holder. Some copyleft licenses, like GNU GPL 3, are designed to counter that. Laws may permit circumvention under specific conditions like when it is necessary to achieve interoperability with the circumventor’s program, or for accessibility reasons; however, distribution of circumvention tools or instructions may be illegal.

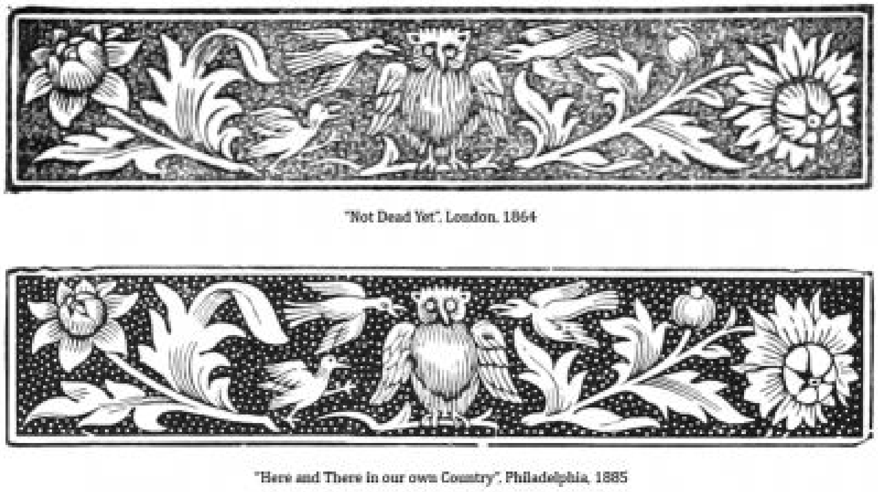
In the context of trademarks, this expansion has been driven by international efforts to harmonize the definition of “trademark”, as exemplified by the Agreement on Trade-Related Aspects of Intellectual Property Rights ratified in 1994, which formalized regulations for IP rights that had been handled by common law, or not at all, in member states. Pursuant to TRIPs, any sign which is “capable of distinguishing” the products or services of one business from the products or services of another business is capable of constituting a trademark.

Licensing & Attributions

CC licensed content, Shared previously

- Intellectual Property. **Provided by:** Wikipedia. **Located at:** https://en.wikipedia.org/wiki/Intellectual_property. **License:** *CC BY-SA: Attribution-ShareAlike*
- World Intellectual Property Organziation. **Authored by:** U.S. Mission Geneva/ Eric Bridiers. **Located at:** https://www.flickr.com/photos/us-mission/15339685942/. **License:** *CC BY-ND: Attribution-NoDerivatives*
- The Battle of Copyright 2011. **Authored by:** CHRISTOPHER DOMBRES. **Located at:** https://www.flickr.com/photos/christopherdombres/5814893360/. **License:** *CC BY: Attribution*

READING: PLAGIARISM



"Not Dead Yet". London. 1864

"Here and There in our own Country". Philadelphia, 1885

An interesting case of early plagiarism: two decorative elements that come from two entirely different books. The one at the bottom was printed twenty years after the one on top.

Introduction

Plagiarism is the “wrongful appropriation” and “stealing and publication” of another author’s “language, thoughts, ideas, or expressions” and the representation of them as one’s own original work. The idea remains problematic with unclear definitions and unclear rules. The modern concept of plagiarism as immoral and originality as an ideal emerged in Europe only in the 18th century, particularly with the Romantic movement.

Plagiarism is considered academic dishonesty and a breach of journalistic ethics. It is subject to sanctions like penalties, suspension, and even expulsion. Recently, cases of ‘extreme plagiarism’ have been identified in academia.

Plagiarism is not a crime *per se* but in academia and industry, it is a serious ethical offense, and cases of plagiarism can constitute copyright infringement.

Etymology

In the 1st century, the use of the Latin word *plagiarius* (literally *kidnapper*) to denote stealing someone else’s work was pioneered by Roman poet Martial, who complained that another poet had “kidnapped his verses.” *“Plagiary”*, a derivative of *“plagiarus”* was introduced into English in 1601 by dramatist Ben Jonson to describe someone guilty of literary theft.

The derived form *plagiarism* was introduced into English around 1620. The Latin *plagiārius*, “kidnapper”, and *plagium*, “kidnapping”, has the root *plaga* (“snare,” “net”), based on the Indo-European root **-plak*, “to weave” (seen for instance in Greek *plekein*, Bulgarian “плета”*pleta*, Latin *plectere*, all meaning “to weave”).

Legal aspects

Although plagiarism in some contexts is considered theft or stealing, the concept does not exist in a legal sense. “Plagiarism” is not mentioned in any current statute, either criminal or civil. Some cases may be treated as unfair competition or a violation of the doctrine of moral rights. The increased availability of intellectual property due to a rise in technology has furthered the debate as to whether copyright offenses are criminal. In short, people are asked to use the guideline, “…if you did not write it yourself, you must give credit.”

Plagiarism is not the same as copyright infringement. While both terms may apply to a particular act, they are different concepts, and false claims of authorship may constitute plagiarism regardless of whether the material is protected by copyright.

Copyright infringement is a violation of the rights of a copyright holder, when material whose use is restricted by copyright is used without consent. Plagiarism, in contrast, is concerned with the unearned increment to the plagiarizing author’s reputation that is achieved through false claims of authorship. Thus, plagiarism is considered a moral offense against the plagiarist’s audience (for example, a reader, listener, or teacher).

Plagiarism is also considered a moral offense against anyone who has provided the plagiarist with a benefit in exchange for what is specifically supposed to be original content (for example, the plagiarist’s publisher, employer, or teacher). In such cases, acts of plagiarism may sometimes also form part of a claim for breach of the plagiarist’s contract, or, if done knowingly, for a civil wrong.

In academia and journalism

Within academia, plagiarism by students, professors, or researchers is considered academic dishonesty or academic fraud, and offenders are subject to academic censure, up to and including expulsion. Many institutions use plagiarism detection software to uncover potential plagiarism and to deter students from plagiarizing. However, the practice of plagiarizing by use of sufficient word substitutions to elude detention software, known as rogeting, has rapidly evolved as students and unethical academics seek to stay ahead of detection software.

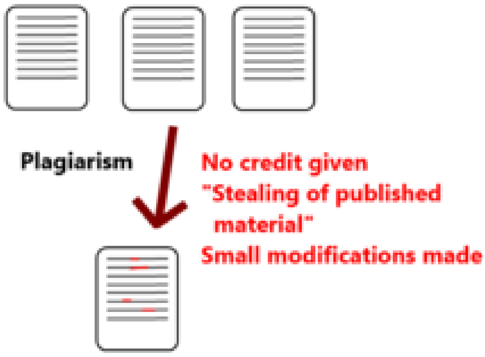
In journalism, plagiarism is considered a breach of journalistic ethics, and reporters caught plagiarizing typically face disciplinary measures ranging from suspension to termination of employment. Some individuals caught plagiarizing in academic or journalistic contexts claim that they plagiarized unintentionally, by failing to include quotations or give the appropriate citation. While plagiarism in scholarship and journalism has a centuries-old history, the development of the Internet, where articles appear as electronic text, has made the physical act of copying the work of others much easier.

Predicated upon an expected level of learning/comprehension having been achieved, all associated academic accreditation becomes seriously undermined if plagiarism is allowed to become the norm within academic submissions.

For professors and researchers, plagiarism is punished by sanctions ranging from suspension to termination, along with the loss of credibility and perceived integrity. Charges of plagiarism against students and professors are typically heard by internal disciplinary committees, by which students and professors have agreed to be bound.

Academia

No universally adopted definition of academic plagiarism exists; however, this section provides several definitions to exemplify the most common characteristics of academic plagiarism.



One form of academic plagiarism involves appropriating a published article and modifying it slightly to avoid suspicion.

According to Bela Gipp academic plagiarism encompasses:

> "The use of ideas, concepts, words, or structures
> without appropriately acknowledging the source
> to benefit in a setting where originality is expected."

The definition by B. Gipp is an abridged version of Teddi Fishman's definition of plagiarism, which proposed five elements characteristic of plagiarism. According to T. Fishman, plagiarism occurs when someone:

1. Uses words, ideas, or work products
2. Attributable to another identifiable person or source
3. Without attributing the work to the source from which it was obtained
4. In a situation in which there is a legitimate expectation of original authorship
5. In order to obtain some benefit, credit, or gain which need not be monetary

Furthermore, plagiarism is defined differently among institutions of higher learning and universities:

- Stanford sees plagiarism as the "use, without giving reasonable and appropriate credit to or acknowledging the author or source, of another person's original work, whether such work is made up of code, formulas, ideas, language, research, strategies, writing or other form."
- Yale views plagiarism as the "use of another's work, words, or ideas without attribution," which includes "using a source's language without quoting, using information from a source without attribution, and paraphrasing a source in a form that stays too close to the original."
- Princeton perceives plagiarism as the "deliberate" use of "someone else's language, ideas, or other original (not common-knowledge) material without acknowledging its source."
- Oxford College of Emory University characterizes plagiarism as the use of "a writer's ideas or phraseology without giving due credit."
- Brown defines plagiarism as "appropriating another person's ideas or words (spoken or written) without attributing those word or ideas to their true source."

Common forms of student plagiarism

According to "The Reality and Solution of College Plagiarism" created by the Health Informatics department of the University of Illinois at Chicago there are 10 main forms of plagiarism that students commit:

1. Submitting someone's work as their own.

2. Taking passages from their own previous work without adding citations.
3. Re-writing someone’s work without properly citing sources.
4. Using quotations, but not citing the source.
5. Interweaving various sources together in the work without citing.
6. Citing some, but not all passages that should be cited.
7. Melding together cited and uncited sections of the piece.
8. Providing proper citations, but fails to change the structure and wording of the borrowed ideas enough.
9. Inaccurately citing the source.
10. Relying too heavily on other people’s work. Fails to bring original thought into the text.

Sanctions for student plagiarism

In the academic world, plagiarism by students is usually considered a very serious offense that can result in punishments such as a failing grade on the particular assignment, the entire course, or even being expelled from the institution. Generally, the punishment increases as a person enters higher institutions of learning. For cases of repeated plagiarism, or for cases in which a student commits severe plagiarism (e.g., submitting a copied piece of writing as original work), suspension or expulsion is likely. A plagiarism tariff has been devised for UK higher education institutions in an attempt to encourage some standardization of this academic problem.

Plagiarism education

Given the serious consequences that plagiarism has for students there has been a call for a greater emphasis on learning in order to help students avoid committing plagiarism. This is especially important when students move to a new institution that may have a different view of the concept when compared the with view previously developed by the student. Indeed, given the seriousness of plagiarism accusations for a student’s future, the pedagogy of plagiarism education may need to be considered ahead of the pedagogy of the discipline being studied.

Journalism

Since journalism relies on the public trust, a reporter’s failure to honestly acknowledge their sources undercuts a newspaper or television news show’s integrity and undermines its credibility. Journalists accused of plagiarism are often suspended from their reporting tasks while the charges are being investigated by the news organization.

The ease with which electronic text can be reproduced from online sources has lured a number of reporters into acts of plagiarism. Journalists have been caught “copying and pasting” articles and text from a number of websites.

Self-plagiarism

Self-plagiarism (also known as “recycling fraud”) is the reuse of significant, identical, or nearly identical portions of one’s own work without acknowledging that one is doing so or without citing the original work. Articles of this nature are often referred to as duplicate or multiple publication. In addition there can be a copyright issue if copyright of the prior work has been transferred to another entity. Typically, self-plagiarism is only considered a serious ethical issue in settings where someone asserts that a publication consists of new material, such as in publishing or factual documentation. It does not apply to public-interest texts, such as social, professional, and cultural opinions usually published in newspapers and magazines.

In academic fields, self-plagiarism occurs when an author reuses portions of his own published and copyrighted work in subsequent publications, but without attributing the previous publication. Identifying self-plagiarism is often difficult because limited reuse of material is accepted both legally (as fair use) and ethically.

It is common for university researchers to rephrase and republish their own work, tailoring it for different academic journals and newspaper articles, to disseminate their work to the widest possible interested public. However, these researchers also obey limits: If half an article is the same as a previous one, it is usually rejected. One of the functions of the process of peer review in academic writing is to prevent this type of “recycling.”

The concept of self-plagiarism

The concept of "self-plagiarism" has been challenged as being self-contradictory, an oxymoron, and on other grounds.

For example, Stephanie J. Bird argues that self-plagiarism is a misnomer, since by definition plagiarism concerns the use of others' material.

However, the phrase is used to refer to specific forms of unethical publication. Bird identifies the ethical issues of "self-plagiarism" as those of "dual or redundant publication." She also notes that in an educational context, "self-plagiarism" refers to the case of a student who resubmits "the same essay for credit in two different courses." As David B. Resnik clarifies, "Self-plagiarism involves dishonesty but not intellectual theft."

According to Patrick M. Scanlon:

> "Self-plagiarism" is a term with some specialized currency. Most prominently, it is used in discussions of research and publishing integrity in biomedicine, where heavy publish-or-perish demands have led to a rash of duplicate and "salami-slicing" publication, the reporting of a single study's results in "least publishable units" within multiple articles (Blancett, Flanagin, & Young, 1995; Jefferson, 1998; Kassirer & Angell, 1995; Lowe, 2003; McCarthy, 1993; Schein & Paladugu, 2001; Wheeler, 1989). Roig (2002) offers a useful classification system including four types of self-plagiarism: duplicate publication of an article in more than one journal; partitioning of one study into multiple publications, often called salami-slicing; text recycling; and copyright infringement.

Self-plagiarism and codes of ethics

Some academic journals have codes of ethics that specifically refer to self-plagiarism. For example, the *Journal of International Business Studies*.

Some professional organizations like the Association for Computing Machinery (ACM) have created policies that deal specifically with self-plagiarism.

Other organizations do not make specific reference to self-plagiarism:

The American Political Science Association (APSA) published a code of ethics that describes plagiarism as "…deliberate appropriation of the works of others represented as one's own." It does not make any reference to self-plagiarism. It does say that when a thesis or dissertation is published "in whole or in part", the author is "not ordinarily under an ethical obligation to acknowledge its origins."

The American Society for Public Administration (ASPA) published a code of ethics that says its members are committed to: "Ensure that others receive credit for their work and contributions," but it makes no reference to self-plagiarism.

Factors that justify reuse

Pamela Samuelson, in 1994, identified several factors she says excuse reuse of one's previously published work, that make it not self-plagiarism. She relates each of these factors specifically to the ethical issue of self-plagiarism, as distinct from the legal issue of fair use of copyright, which she deals with separately. Among other factors that may excuse reuse of previously published material Samuelson lists the following:

1. The previous work must be restated to lay the groundwork for a new contribution in the second work.
2. Portions of the previous work must be repeated to deal with new evidence or arguments.
3. The audience for each work is so different that publishing the same work in different places is necessary to get the message out.
4. The author thinks they said it so well the first time that it makes no sense to say it differently a second time.

Samuelson states she has relied on the “different audience” rationale when attempting to bridge interdisciplinary communities. She refers to writing for different legal and technical communities, saying: “there are often paragraphs or sequences of paragraphs that can be bodily lifted from one article to the other. And, in truth, I lift them.” She refers to her own practice of converting “a technical article into a law review article with relatively few changes—adding footnotes and one substantive section” for a different audience.

Samuelson describes misrepresentation as the basis of self-plagiarism. She also states “Although it seems not to have been raised in any of the self-plagiarism cases, copyrights law’s fair use defense would likely provide a shield against many potential publisher claims of copyright infringement against authors who reused portions of their previous works.”

Organizational publications

Plagiarism is presumably not an issue when organizations issue collective unsigned works since they do not assign credit for originality to particular people. For example, the American Historical Association’s “Statement on Standards of Professional Conduct” (2005) regarding textbooks and reference books states that, since textbooks and encyclopedias are summaries of other scholars’ work, they are not bound by the same exacting standards of attribution as original research and may be allowed a greater “extent of dependence” on other works. However, even such a book does not make use of words, phrases, or paragraphs from another text or follow too closely the other text’s arrangement and organization, and the authors of such texts are also expected to “acknowledge the sources of recent or distinctive findings and interpretations, those not yet a part of the common understanding of the profession.”

In the arts

Plagiarism and the history of art

Through all of the history of literature and of the arts in general, works of art are for a large part repetitions of the tradition; to the entire history of artistic creativity belong plagiarism, literary theft,appropriation, incorporation, retelling, rewriting, recapitulation, revision, reprise, thematic variation, ironic retake, parody, imitation, stylistic theft, pastiches, collages, and deliberate assemblages. There is no rigorous and precise distinction between practices like imitation, stylistic plagiarism, copy, replica and forgery. These appropriation procedures are the main axis of a literate culture, in which the tradition of the canonic past is being constantly rewritten.

Ruth Graham quotes T.S. Eliot—”Immature poets imitate; mature poets steal. Bad poets deface what they take.”—she notes that despite the “taboo” of plagiarism, the ill-will and embarrassment it causes in the modern context, readers seem to often forgive the past excesses of historic literary offenders.

L.H.O.O.Q. (1919), one of Marcel Duchamp’s readymades.

Praisings of artistic plagiarism

A passage of Laurence Sterne’s 1767 *Tristram Shandy*, condemns plagiarism by resorting to plagiarism. Oliver Goldsmith commented:

> Sterne’s Writings, in which it is clearly shewn, that he, whose manner and style were so long thought original, was, in fact, the most unhesitating plagiarist who ever cribbed from his predecessors in order to garnish his own pages. It must be owned, at the same time, that Sterne selects the materials of his mosaic work with so much art, places them so well, and polishes them so

highly, that in most cases we are disposed to pardon the want of originality, in consideration of the exquisite talent with which the borrowed materials are wrought up into the new form.

In other contexts

Plagiarism on the Internet

Content scraping is copying and pasting from websites and blogs.

Free online tools are becoming available to help identify plagiarism, and there are a range of approaches that attempt to limit online copying, such as disabling right clicking and placing warning banners regarding copyrights on web pages. Instances of plagiarism that involve copyright violation may be addressed by the rightful content owners sending a DMCA removal notice to the offending site-owner, or to the ISP that is hosting the offending site.

Licensing & Attributions

CC licensed content, Shared previously

- Plagiarism. **Provided by:** Wikipedia. **Located at:** https://en.wikipedia.org/wiki/Plagiarism. **License:** *CC BY-SA: Attribution-ShareAlike*
- An interesting case of early plagiarism. **Authored by:** Mario Klingemann. **Located at:** https://www.flickr.com/photos/quasimondo/14660625844/. **License:** *CC BY-NC: Attribution-NonCommercial*
- Staff and student perceptions of plagiarism. **Authored by:** Jobadge. **Located at:** https://www.flickr.com/photos/24612276@N05/5902067107/. **License:** *CC BY-NC: Attribution-NonCommercial*

READING: INFORMATION PRIVACY

Information privacy, or **data privacy (or data protection)**, is the relationship between collection and dissemination of data, technology, the public expectation of privacy, and the legal and political issues surrounding them.

Privacy concerns exist wherever personally identifiable information or other sensitive information is collected and stored—in digital form or otherwise. Improper or non-existent disclosure control can be the root cause for privacy issues. Data privacy issues can arise in response to information from a wide range of sources, such as:

- Healthcare records
- Criminal justice investigations and proceedings
- Financial institutions and transactions
- Biological traits, such as genetic material
- Residence and geographic records
- Ethnicity
- Privacy breach
- Location-based service and geolocation

The challenge in data privacy is to share data while protecting personally identifiable information. The fields of data security and information security design and utilize software, hardware and human resources to address this issue. As the laws and regulations related to Data Protection are constantly changing, it is important to keep abreast of any changes in the law and continually reassess your compliance with data privacy and security regulations.

Information types

Various types of personal information often come under privacy concerns.

Internet

The ability to control the information one reveals about oneself over the Internet, and who can access that information, has become a growing concern. These concerns include whether email can be stored or read by third parties without consent, or whether third parties can continue to track the web sites someone has visited. Another concern is web sites which are visited collect, store, and possibly share personally identifiable information about users.

The advent of various search engines and the use of data mining created a capability for data about individuals to be collected and combined from a wide variety of sources very easily. The FTC has provided a set of guidelines that represent widely accepted concepts concerning fair information practices in an electronic marketplace called the Fair Information Practice Principles.

In order not to give away too much personal information, e-mails should be encrypted and browsing of webpages as well as other online activities should be done trace-less via anonymizers, or, in cases those are not trusted, by open source distributed anonymizers, so-called mix nets, such as I2P – The Onion Router or Tor.

Email isn't the only Internet use with concern of privacy. Everything is accessible over the Internet nowadays. However a major issue with privacy relates back to social networking. For example, there are millions of users on Facebook, and regulations have changed. People may be tagged in photos or have valuable information exposed about themselves either by choice or most of the time unexpectedly by others. It is important to be cautious of what is being said over the Internet and what information is being displayed as well as photos because this all can searched across the web and used to access private databases making it easy for anyone to quickly go online and profile a person.

Cable television

The ability to control what information one reveals about oneself over cable television, and who can access that information. For example, third parties can track IP TV programs someone has watched at any given time.

> The addition of any information in a broadcasting stream is not required for an audience rating survey, additional devices are not requested to be installed in the houses of viewers or listeners, and without the necessity of their cooperation, audience ratings can be automatically performed in real-time.

Medical

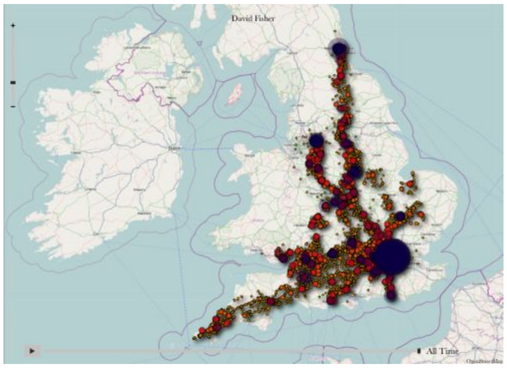
A person may not wish for their medical records to be revealed to others. This may be because they have concern that it might affect their insurance coverages or employment. Or it may be because they would not wish for others to know about medical or psychological conditions or treatments which would be embarrassing. Revealing medical data could also reveal other details about one's personal life. Privacy Breach There are three major categories of medical privacy: informational (the degree of control over personal information), physical (the degree of physical inaccessibility to others), and psychological (the extent to which the doctor respects patients' cultural beliefs, inner thoughts, values, feelings, and religious practices and allows them to make personal decisions). Physicians and psychiatrists in many cultures and countries have standards for doctor-patient relationships which include maintaining confidentiality. In some cases, the physician-patient privilege is legally protected. These practices are in place to protect the dignity of patients, and to ensure that patients will feel free to reveal complete and accurate information required for them to receive the correct treatment. The United States has laws governing privacy of private health information, see HIPAA and the HITECH Act.

Financial

Information about a person's financial transactions, including the amount of assets, positions held in stocks or funds, outstanding debts, and purchases can be sensitive. If criminals gain access to information such as a person's accounts or credit card numbers, that person could become the victim of fraud or identity theft. Information about a person's purchases can reveal a great deal about that person's history, such as places he/she has visited, whom he/she has contacted with, products he/she has used, his/her activities and habits, or medications he/she has used. In some cases corporations might wish to use this information to target individuals with marketing customized towards those individual's personal preferences, something which that person may or may not approve.

Locational

As location tracking capabilities of mobile devices are increasing (Location-based service), problems related to user privacy arise. Location data is indeed among the most sensitive data currently being collected. A list of potentially sensitive professional and personal information that could be inferred about an individual knowing only his mobility trace was published recently by the Electronic Frontier Foundation. These include the movements of a competitor sales force, attendance of a particular church or an individual's presence in a motel or at an abortion clinic. A recent MIT study by de Montjoye et al. showed that 4 spatio-temporal points, approximate places and times, are enough to uniquely identify 95% of 1.5M people in a mobility database. The study further shows that these constraints hold even when the resolution of the dataset is low. Therefore, even coarse or blurred datasets provide little anonymity.



The dots show all the locations logged by an iPhone without the user's knowledge.

Political

Political privacy has been a concern since voting systems emerged in ancient times. The secret ballot is the simplest and most widespread measure to ensure that political views are not known to anyone other than the voter him/herself—it is nearly universal in modern democracy, and considered to be a basic right of citizenship. In fact even where other rights of privacy do not exist, this type of privacy very often does.

Educational

In the United Kingdom, in 2012 the Education Secretary Michael Gove described the National Pupil Database as a "rich dataset" whose value could be "maximized" by making it more openly accessible, including to private companies. Kelly Fiveash of *The Register* said that this could mean "a child's school life including exam results, attendance, teacher assessments and even characteristics" could be available, with third-party organizations being responsible for anonymizing any publications themselves, rather than the data being anonymized by the government before being handed over. An example of a data request that Gove indicated had been rejected in the past, but might be possible under an improved version of privacy regulations, was for "analysis on sexual exploitation."

Legality

The legal protection of the right to privacy in general – and of data privacy in particular – varies greatly around the world.

> No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks upon his honor and reputation. Everyone has the right to the protection of the law against such interference or attacks.—Universal Declaration of Human Rights, Article 12

There is a significant challenge for organizations that hold sensitive data to achieve and maintain compliance with so many regulations that have relevance to information privacy.

Safe Harbor Program and Passenger Name Record issues

The United States Department of Commerce created the International Safe Harbor Privacy Principles certification program in response to the1995 Directive on Data Protection (Directive 95/46/EC) of the European Commission. Directive 95/46/EC declares in Chapter IV Article 25 that personal data may only be transferred from the countries in the European Economic Area to countries which provide adequate privacy protection. Historically, establishing adequacy required the creation of national laws broadly equivalent to those implemented by Directive 95/46/EU. Although there are exceptions to this blanket prohibition – for example where the disclosure to a country outside the EEA is made with the consent of the relevant individual (Article 26(1)(a)) – they are limited in practical scope. As a result, Article 25 created a legal risk to organizations which transfer personal data from Europe to the United States.

Commandant of the Coast Guard Adm. Thad W. Allen, right, delivers remarks as Chief of Naval Operations Adm. Gary Roughead looks on after the signing of the memorandum of agreement for the Safe Harbor program during a signing ceremony at the Pentagon.

The program has an important issue on the exchange of Passenger Name Record information between the EU and the US. According to the EU directive, personal data may only be transferred to third countries if that country provides an adequate level of protection. Some exceptions to this rule are provided, for instance when the controller himself can guarantee that the recipient will comply with the data protection rules.

The European Commission has set up the “Working party on the Protection of Individuals with regard to the Processing of Personal Data,” commonly known as the “Article 29 Working Party”. The Working Party gives advice about the level of protection in the European Union and third countries.

The Working Party negotiated with U.S. representatives about the protection of personal data, the Safe Harbor Principles were the result. Notwithstanding that approval, the self-assessment approach of the Safe Harbor remains controversial with a number of European privacy regulators and commentators.

The Safe Harbor program addresses this issue in a unique way: rather than a blanket law imposed on all organizations in the United States, a voluntary program is enforced by the FTC. U.S. organizations which register with this program, having self-assessed their compliance with a number of standards, are “deemed adequate” for the purposes of Article 25. Personal information can be sent to such organizations from the EEA without the sender being in breach of Article 25 or its EU national equivalents. The Safe Harbor was approved as providing adequate protection for personal data, for the purposes of Article 25(6), by the European Commission on 26 July 2000.

The Safe Harbor is not a perfect solution to the challenges posed by Article 25. In particular, adoptee organisations need to carefully consider their compliance with the *onward transfer obligations*, where personal data originating in the EU is transferred to the US Safe Harbor, and then onward to a third country. The alternative compliance approach of “binding corporate rules”, recommended by many EU privacy regulators, resolves this issue. In addition, any dispute arising in relation to the transfer of HR data to the US Safe Harbor must be heard by a panel of EU privacy regulators.

In July 2007, a new, controversial, Passenger Name Record agreement between the US and the EU was undersigned. A short time afterwards, the Bush administration gave exemption for the Department of Homeland Security, for the Arrival and Departure Information System (ADIS) and for the Automated Target System from the 1974 Privacy Act.

In February 2008, Jonathan Faull, the head of the EU’s Commission of Home Affairs, complained about the US bilateral policy concerning PNR. The US had signed in February 2008 a memorandum of understanding (MOU) with the Czech Republic in exchange of a VISA waiver scheme, without concerting before with Brussels. The tensions between Washington and Brussels are mainly caused by a lesser level of data protection in the US, especially since foreigners do not benefit from the US Privacy Act of 1974. Other countries approached for bilateral MOU included the United Kingdom, Estonia, Germany and Greece.

Protecting privacy in information systems

As heterogeneous information systems with differing privacy rules are interconnected and information is shared, policy appliances will be required to reconcile, enforce and monitor an increasing amount of privacy policy rules (and laws). There are two categories of technology to address privacy protection in commercial IT systems: communication and enforcement.

Policy Communication

- P3P—The Platform for Privacy Preferences. P3P is a standard for communicating privacy practices and comparing them to the preferences of individuals.

Policy Enforcement

- XACML—The Extensible Access Control Markup Language together with its Privacy Profile is a standard for expressing privacy policies in a machine-readable language which a software system can use to enforce the policy in enterprise IT systems.
- EPAL—The Enterprise Privacy Authorization Language is very similar to XACML, but is not yet a standard.
- WS-Privacy—"Web Service Privacy" will be a specification for communicating privacy policy in web services. For example, it may specify how privacy policy information can be embedded in the SOAP envelope of a web service message.

Protecting Privacy on the Internet

On the Internet you almost always give away a lot of information about yourself: Unencrypted e-mails can be read by the administrators of the e-mail server, if the connection is not encrypted (no https), and also the Internet service provider and other parties sniffing the traffic of that connection are able to know the contents. Furthermore, the same applies to any kind of traffic generated on the Internet (web-browsing,instant messaging, among others) In order not to give away too much personal information, e-mails can be encrypted and browsing of webpages as well as other online activities can be done traceless via anonymizers, or, in cases those are not trusted, by open source distributed anonymizers, so called mix nets. Renowned open-source mix nets are I2P—The Anonymous Network or tor.

Licensing & Attributions

CC licensed content, Shared previously

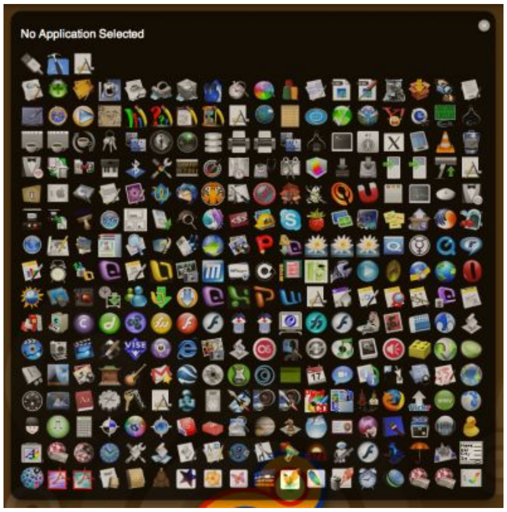
- Information Privacy. **Provided by:** Wikipedia. **Located at:** https://en.wikipedia.org/wiki/Information_privacy. **License:** *CC BY-SA: Attribution-ShareAlike*
- Your data for sale. **Authored by:** Anita Hart. **Located at:** https://www.flickr.com/photos/anitakhart/5562739925/. **License:** *CC BY-SA: Attribution-ShareAlike*
- iPhone locations. **Authored by:** David Fisher. **Located at:** https://www.flickr.com/photos/dave_fisher/5639609289/. **License:** *CC BY: Attribution*

Public domain content

- Safe Harbor Agreement. **Authored by:** Anonymous. **Provided by:** U.S. Navy. **Located at:** https://en.wikipedia.org/wiki/Software_development. **License:** *Public Domain: No Known Copyright*

READING: SOFTWARE DEVELOPMENT

Introduction



Software development is the computer programming, documenting, testing, and bug fixing involved in creating and maintaining applications and frameworks involved in a software release life cycle and resulting in a software product. The term refers to a process of writing and maintaining the source code, but in a broader sense of the term it includes all that is involved between the conception of the desired software through to the final manifestation of the software, ideally in a planned and structured process. Therefore, software development may include research, new development, prototyping, modification, reuse, re-engineering, maintenance, or any other activities that result in software products.

Software can be developed for a variety of purposes, the three most common being to meet specific needs of a specific client/business (the case with custom software), to meet a perceived need of some set of potential users (the case with commercial and open source software), or for personal use (e.g. a scientist may write software to automate a mundane task). Embedded software development, that is, the development of embedded software such as used for controlling consumer products, requires the development process to be integrated with the development of the controlled physical product. System software underlies applications and the programming process itself, and is often developed separately.

The need for better quality control of the software development process has given rise to the discipline of software engineering, which aims to apply the systematic approach exemplified in the engineering paradigm to the process of software development.

There are many approaches to software project management, known as software development life cycle models, methodologies, processes, or models. The waterfall model is a traditional version, contrasted with the more recent innovation of agile software development.

Methodologies

A software development methodology (also known as a software development process, model, or life cycle) is a framework that is used to structure, plan, and control the process of developing information systems. A wide variety of such frameworks have evolved over the years, each with its own recognized strengths and weaknesses. There are several different approaches to software development: some take a more structured, engineering-based approach to developing business solutions, whereas others may take a more incremental approach, where software evolves as it is developed piece-by-piece. One system development methodology is not necessarily suitable for use by all projects. Each of the available methodologies is best suited to specific kinds of projects, based on various technical, organizational, project and team considerations.

Most methodologies share some combination of the following stages of software development:

- Analyzing the problem
- Market research
- Gathering requirements for the proposed business solution
- Devising a plan or design for the software-based solution
- Implementation (coding) of the software
- Testing the software
- Deployment

- Maintenance and bug fixing

These stages are often referred to collectively as the software development lifecycle, or SDLC. Different approaches to software development may carry out these stages in different orders, or devote more or less time to different stages. The level of detail of the documentation produced at each stage of software development may also vary. These stages may also be carried out in turn (a “waterfall” based approach), or they may be repeated over various cycles or iterations (a more “extreme” approach). The more extreme approach usually involves less time spent on planning and documentation, and more time spent on coding and development of automated tests. More “extreme” approaches also promote continuous testing throughout the development lifecycle, as well as having a working (or bug-free) product at all times. More structured or “waterfall” based approaches attempt to assess the majority of risks and develop a detailed plan for the software before implementation (coding) begins, and avoid significant design changes and re-coding in later stages of the software development life cycle planning.

There are significant advantages and disadvantages to the various methodologies, and the best approach to solving a problem using software will often depend on the type of problem. If the problem is well understood and a solution can be effectively planned out ahead of time, the more “waterfall” based approach may work the best. If, on the other hand, the problem is unique (at least to the development team) and the structure of the software solution cannot be easily envisioned, then a more “extreme” incremental approach may work best.

Software development activities

Identification of need

The sources of ideas for software products are legion. These ideas can come from market research including the demographics of potential new customers, existing customers, sales prospects who rejected the product, other internal software development staff, or a creative third party. Ideas for software products are usually first evaluated by marketing personnel for economic feasibility, for fit with existing channels distribution, for possible effects on existing product lines, required features, and for fit with the company’s marketing objectives. In a marketing evaluation phase, the cost and time assumptions become evaluated. A decision is reached early in the first phase as to whether, based on the more detailed information generated by the marketing and development staff, the project should be pursued further.

In the book *“Great Software Debates”*, Alan M. Davis states in the chapter *“Requirements,”* subchapter *“The Missing Piece of Software Development”*

> Students of engineering learn engineering and are rarely exposed to finance or marketing. Students of marketing learn marketing and are rarely exposed to finance or engineering. Most of us become specialists in just one area. To complicate matters, few of us meet interdisciplinary people in the workforce, so there are few roles to mimic. Yet, software product planning is critical to the development success and absolutely requires knowledge of multiple disciplines.

Because software development may involve compromising or going beyond what is required by the client, a software development project may stray into less technical concerns such as human resources, risk management, intellectual property, budgeting, crisis management, etc. These processes may also cause the role of business development to overlap with software development.

Planning

Planning is an objective of each and every activity, where we want to discover things that belong to the project. An important task in creating a software program is extracting the requirements or requirements analysis. Customers typically have an abstract idea of what they want as an end result, but do not know what *software* should do. Skilled and experienced software engineers recognize incomplete, ambiguous, or even contradictory requirements at this point. Frequently demonstrating live code may help reduce the risk that the requirements are incorrect.

Once the general requirements are gathered from the client, an analysis of the scope of the development should be determined and clearly stated. This is often called a scope document.

Certain functionality may be out of scope of the project as a function of cost or as a result of unclear requirements at the start of development. If the development is done externally, this document can be considered a legal document so that if there are ever disputes, any ambiguity of what was promised to the client can be clarified.

Designing

Once the requirements are established, the design of the software can be established in a software design document. This involves a preliminary, or high-level design of the main modules with an overall picture (such as a block diagram) of how the parts fit together. The language, operating system, and hardware components should all be known at this time. Then a detailed or low-level design is created, perhaps with prototyping as proof-of-concept or to firm up requirements.

Implementation, testing and documenting

Implementation is the part of the process where software engineers actually program the code for the project.

Software testing is an integral and important phase of the software development process. This part of the process ensures that defects are recognized as soon as possible. In some processes, generally known as test-driven development, tests may be developed just before implementation and serve as a guide for the implementation's correctness.

Documenting the internal design of software for the purpose of future maintenance and enhancement is done throughout development. This may also include the writing of an API, be it external or internal. The software engineering process chosen by the developing team will determine how much internal documentation (if any) is necessary. Plan-driven models (e.g., Waterfall) generally produce more documentation than Agile models.

Deployment and maintenance

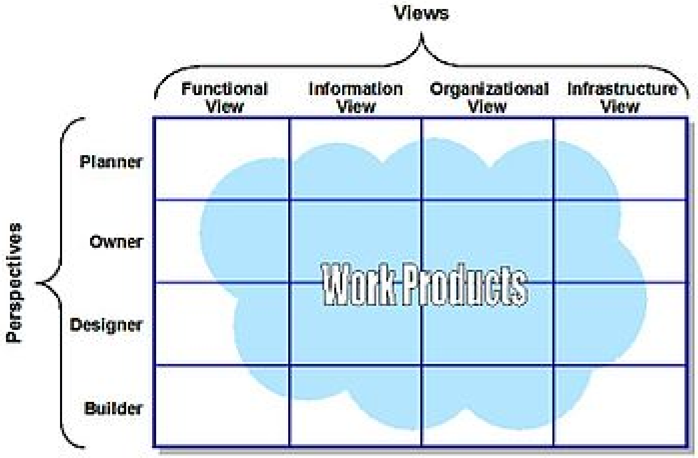
Deployment starts directly after the code is appropriately tested, approved for release, and sold or otherwise distributed into a production environment. This may involve installation, customization (such as by setting parameters to the customer's values), testing, and possibly an extended period of evaluation.

Software training and support is important, as software is only effective if it is used correctly.

Maintaining and enhancing software to cope with newly discovered faults or requirements can take substantial time and effort, as missed requirements may force redesign of the software.

Subtopics

View model



The TEAF Matrix of Views and Perspectives.

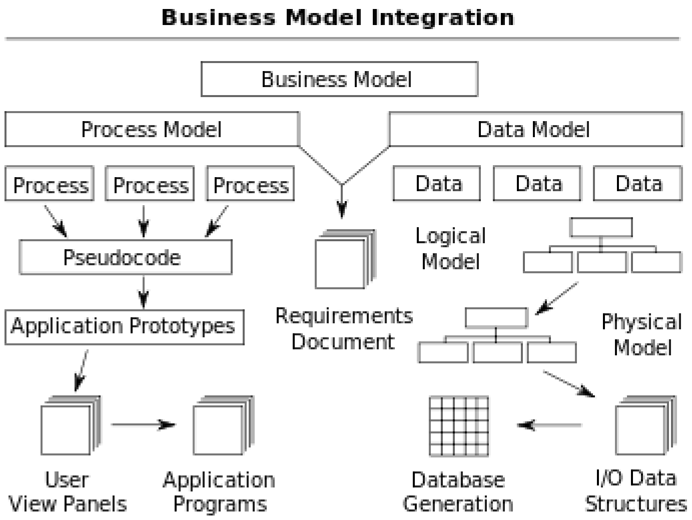
A view model is a framework that provides the viewpoints on the system and its environment, to be used in the software development process. It is a graphical representation of the underlying semantics of a view.

The purpose of viewpoints and views is to enable human engineers to comprehend very complex systems, and to organize the elements of the problem and the solution around domains of expertise. In the engineering of physically intensive systems, viewpoints often correspond to capabilities and responsibilities within the engineering organization.

Most complex system specifications are so extensive that no one individual can fully comprehend all aspects of the specifications. Furthermore, we all have different interests in a given system and different reasons for examining the system's specifications. A business executive will ask different questions of a system make-up than would a system implementer. The concept of viewpoints framework, therefore, is to provide separate viewpoints into the specification of a given complex system. These viewpoints each satisfy an audience with interest in some set of aspects of the system. Associated with each viewpoint is a viewpoint language that optimizes the vocabulary and presentation for the audience of that viewpoint.

Business process and data modeling

Graphical representation of the current state of information provides a very effective means for presenting information to both users and system developers.



Example of the interaction between business process and data models.

- A business model illustrates the functions associated with the business process being modeled and the organizations that perform these functions. By depicting activities and information flows, a foundation is created to visualize, define, understand, and validate the nature of a process.
- A data model provides the details of information to be stored, and is of primary use when the final product is the generation of computer software code for an application or the preparation of a functional specification to aid a computer software make-or-buy decision. See the figure on the right for an example of the interaction between business process and data models.

Usually, a model is created after conducting an interview, referred to asbusiness analysis. The interview consists of a facilitator asking a series of questions designed to extract required information that describes a process. The interviewer is called a facilitator to emphasize that it is the participants who provide the information. The facilitator should have some knowledge of the process of interest, but this is not as important as having a structured methodology by which the questions are asked of the process expert. The methodology is important because usually a team of facilitators is collecting information across the facility and the results of the information from all the interviewers must fit together once completed.

The models are developed as defining either the current state of the process, in which case the final product is called the “as-is” snapshot model, or a collection of ideas of what the process should contain, resulting in a “what-can-be” model. Generation of process and data models can be used to determine if the existing processes and information systems are sound and only need minor modifications or enhancements, or if re-engineering is required as a corrective action. The creation of business models is more than a way to view or automate your information process. Analysis can be used to fundamentally reshape the way your business or organization conducts its operations.

Computer-aided software engineering

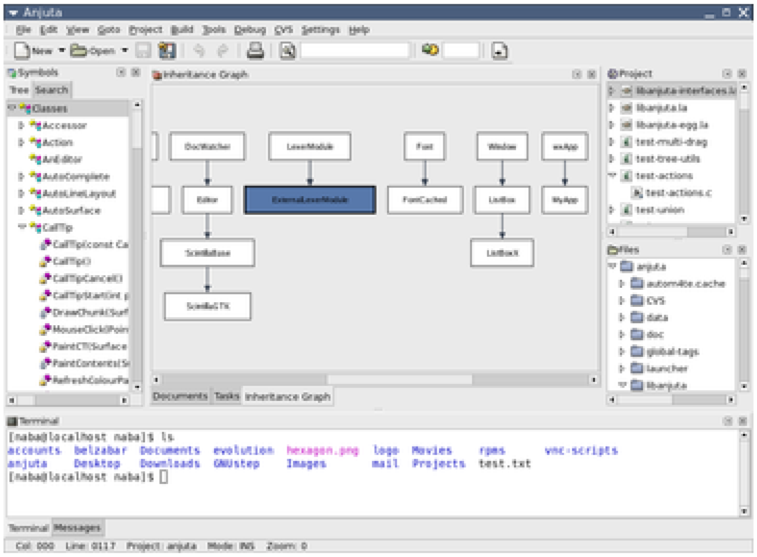
Computer-aided software engineering (CASE), in the field software engineering is the scientific application of a set of tools and methods to a software which results in high-quality, defect-free, and maintainable software products. It also refers to methods for the development of information systems together with automated tools that can be used in the software development process. The term “computer-aided software engineering” (CASE) can refer to the software used for the automated development of systems software, i.e., computer code. The CASE functions include analysis, design, and programming. CASE tools automate methods for designing, documenting, and producing structured computer code in the desired programming language.

Two key ideas of Computer-aided Software System Engineering (CASE) are:

- Foster computer assistance in software development and or software maintenance processes, and
- An engineering approach to software development and or maintenance.

Typical CASE tools exist for configuration management, data modeling, model transformation, refactoring, source code generation.

Integrated development environment



```
[naba@localhost naba]$ ls
accounts  belzabar  Documents  evolution  hexagon.png  logo  Movies    rpms      vnc-scripts
anjuta    Desktop   Downloads  GNUstep    Images       mail  Projects  test.txt
[naba@localhost naba]$ 
```

Anjuta, a C and C++ IDE for the GNOME environment

An integrated development environment (IDE) also known as *integrated design environment* or *integrated debugging environment* is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of a:

- source code editor,
- compiler and/or interpreter,
- build automation tools, and
- debugger (usually).

IDEs are designed to maximize programmer productivity by providing tight-knit components with similar user interfaces. Typically an IDE is dedicated to a specific programming language, so as to provide a feature set which most closely matches the programming paradigms of the language.

Modeling language

A modeling language is any artificial language that can be used to express information or knowledge or systems in a structure that is defined by a consistent set of rules. The rules are used for interpretation of the meaning of components in the structure. A modeling language can be graphical or textual. Graphical modeling languages use a diagram techniques with named symbols that represent concepts and lines that connect the symbols and that represent relationships and various other graphical annotation to represent constraints. Textual modeling languages typically use standardized keywords accompanied by parameters to make computer-interpretable expressions.

Example of graphical modeling languages in the field of software engineering are:

- Business Process Modeling Notation (BPMN, and the XML form BPML) is an example of a process modeling language.
- EXPRESS and EXPRESS-G (ISO 10303-11) is an international standard general-purpose data modeling language.
- Extended Enterprise Modeling Language (EEML) is commonly used for business process modeling across layers.
- Flowchart is a schematic representation of an algorithm or a stepwise process,
- Fundamental Modeling Concepts (FMC) modeling language for software-intensive systems.

- IDEF is a family of modeling languages, the most notable of which include IDEF0 for functional modeling, IDEF1X for information modeling, and IDEF5 for modeling ontologies.
- LePUS3 is an object-oriented visual Design Description Language and a formal specification language that is suitable primarily for modelling large object-oriented (Java, C++, C#) programs and design patterns.
- Specification and Description Language(SDL) is a specification language targeted at the unambiguous specification and description of the behaviour of reactive and distributed systems.
- Unified Modeling Language (UML) is a general-purpose modeling language that is an industry standard for specifying software-intensive systems. UML 2.0, the current version, supports thirteen different diagram techniques, and has widespread tool support.

Not all modeling languages are executable, and for those that are, using them doesn't necessarily mean that programmers are no longer needed. On the contrary, executable modeling languages are intended to amplify the productivity of skilled programmers, so that they can address more difficult problems, such as parallel computing and distributed systems.

Programming paradigm

A programming paradigm is a fundamental style of computer programming, which is not generally dictated by the project management methodology (such as waterfall or agile). Paradigms differ in the concepts and abstractions used to represent the elements of a program (such as objects, functions, variables, constraints) and the steps that comprise a computation (such as assignations, evaluation, continuations, data flows). Sometimes the concepts asserted by the paradigm are utilized cooperatively in high-level system architecture design; in other cases, the programming paradigm's scope is limited to the internal structure of a particular program or module.

A programming language can support multiple paradigms. For example programs written in C++ or Object Pascal can be purely procedural, or purely object-oriented, or contain elements of both paradigms. Software designers and programmers decide how to use those paradigm elements. In object-oriented programming, programmers can think of a program as a collection of interacting objects, while in functional programming a program can be thought of as a sequence of stateless function evaluations. When programming computers or systems with many processors, process-oriented programming allows programmers to think about applications as sets of concurrent processes acting upon logically shared data structures.

Just as different groups in software engineering advocate different *methodologies*, different programming languages advocate different *programming paradigms*. Some languages are designed to support one paradigm (Smalltalk supports object-oriented programming, Haskellsupports functional programming), while other programming languages support multiple paradigms (such as Object Pascal, C++, C#, Visual Basic, Common Lisp, Scheme, Python, Ruby, and Oz).

Many programming paradigms are as well known for what methods they *forbid* as for what they enable. For instance, pure functional programming forbids using side-effects; structured programming forbids using goto statements. Partly for this reason, new paradigms are often regarded as doctrinaire or overly rigid by those accustomed to earlier styles. Avoiding certain methods can make it easier to prove theorems about a program's correctness, or simply to understand its behavior.

Examples of high-level paradigms include:

- Aspect-oriented software development
- Domain-specific modeling
- Model-driven engineering
- Object-oriented programming methodologies
    - Grady Booch's object-oriented design (OOD), also known as object-oriented analysis and design (OOAD). The Booch model includes six diagrams: class, object, state transition, interaction, module, and process.
- Search-based software engineering
- Service-oriented modeling
- Structured programming
- Top-down and bottom-up design
    - Top-down programming: evolved in the 1970s by IBM researcher Harlan Mills (and Niklaus Wirth) in developed structured programming.

Software framework

A software framework is a re-usable design for a software system or subsystem. A software framework may include support programs, code libraries, a scripting language, or other software to help develop and *glue together* the different components of a software project. Various parts of the framework may be exposed via an API.

Licensing & Attributions

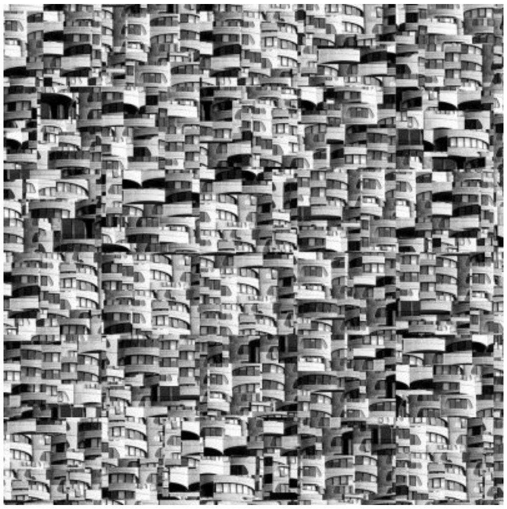
CC licensed content, Shared previously

- Software Development. **Provided by:** Wikipedia. **Located at:** https://en.wikipedia.org/wiki/Software_development. **License:** *CC BY-SA: Attribution-ShareAlike*
- My Apps on Todos. **Authored by:** Adam Schilling. **Located at:** https://www.flickr.com/photos/schilling/198887098/. **License:** *CC BY-NC-ND: Attribution-NonCommercial-NoDerivatives*
- Idea. **Authored by:** thesaltr. **Located at:** https://www.flickr.com/photos/thesaltr/7941318940/. **License:** *CC BY-NC-ND: Attribution-NonCommercial-NoDerivatives*

Public domain content

- **Authored by:** Anonymous. **Provided by:** U.S. Navy. **Located at:** https://commons.wikimedia.org/wiki/File:Safe_Harbor_agreement_DVIDS161960.jpg. **License:** *Public Domain: No Known Copyright*

READING: STRUCTURED PROGRAMMING



Structured programming is a programming paradigm aimed at improving the clarity, quality, and development time of a computer program by making extensive use of subroutines, block structures and for and while loops—in contrast to using simple tests and jumps such as the *goto* statement which could lead to “spaghetti code” which is difficult both to follow and to maintain

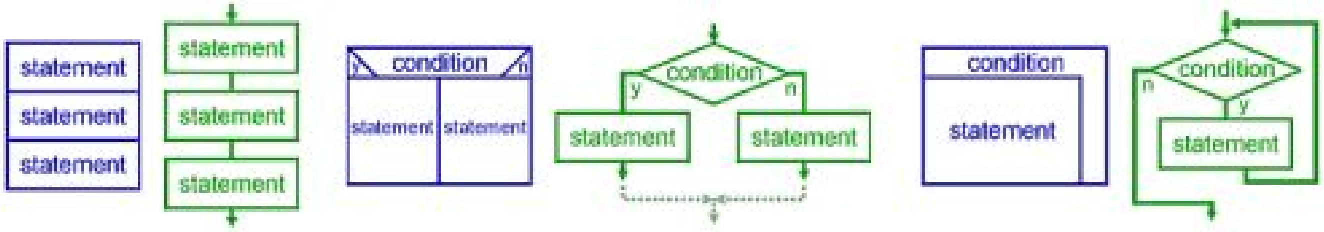
It emerged in the 1960s—particularly from a famous letter, *Go To Statement Considered Harmful.*— and was bolstered theoretically by the structured program theorem, and practically by the emergence of languages such as ALGOL with suitably rich control structures.

Elements

Control structures

Following the structured program theorem, all programs are seen as composed of three control structures:

- “Sequence”; ordered statements or subroutines executed in sequence.
- “Selection”; one or a number of statements is executed depending on the state of the program. This is usually expressed with keywords such as `if..then..else..endif`.
- “Iteration”; a statement or block is executed until the program reaches a certain state, or operations have been applied to every element of a collection. This is usually expressed with keywords such as `while`, `repeat`, `for` or `do..until`. Often it is recommended that each loop should only have one entry point (and in the original structural programming, also only one exit point, and a few languages enforce this).



Graphical representations of the three basic patterns using NS diagrams (blue) and flow charts (green).

Subroutines

Subroutines; callable units such as procedures, functions, methods, or subprograms are used to allow a sequence to be referred to by a single statement.

Blocks

Blocks are used to enable groups of statements to be treated as if they were one statement. *Block-structured* languages have a syntax for enclosing structures in some formal way, such as an if-statement bracketed by `if..fi` as in ALGOL 68, or a code section bracketed by`BEGIN..END`, as in PL/I, whitespace indentation as in Python – or the curly braces `{...}` of C and many later languages.

Structured programming languages

It is possible to do structured programming in any programming language, though it is preferable to use something like a procedural programming language. Some of the languages initially used for structured programming include: ALGOL, Pascal, PL/I and Ada—but most new procedural programming languages since that time have included features to encourage structured programming, and sometimes deliberately left out features—notably GOTO—in an effort to make unstructured programming more difficult. *Structured programming* (sometimes known as modular programming) is a subset of procedural programming that enforces a logical structure on the program being written to make it more efficient and easier to understand and modify.

History

Theoretical foundation

The structured program theorem provides the theoretical basis of structured programming. It states that three ways of combining programs—sequencing, selection, and iteration—are sufficient to express any computable function. This observation did not originate with the structured programming movement; these structures are sufficient to describe the instruction cycle of a central processing unit, as well as the operation of a Turing machine. Therefore a processor is always executing a “structured program” in this sense, even if the instructions it reads from memory are not part of a structured program. However, authors usually credit the result to a 1966 paper by Böhm and Jacopini, possibly because Dijkstra cited this paper himself. The structured program theorem does not address how to write and analyze a usefully structured program. These issues were addressed during the late 1960s and early 1970s, with major contributions by Dijkstra, Robert W. Floyd, Tony Hoare, Ole-Johan Dahl, and David Gries.

Debate

P. J. Plauger, an early adopter of structured programming, described his reaction to the structured program theorem:

> Us converts waved this interesting bit of news under the noses of the unreconstructed assembly-language programmers who kept trotting forth twisty bits of logic and saying, ‘I betcha can’t structure this.’ Neither the proof by Böhm and Jacopini nor our repeated successes at writing structured code brought them around one day sooner than they were ready to convince themselves.

Donald Knuth accepted the principle that programs must be written with provability in mind, but he disagreed (and still disagrees) with abolishing the GOTO statement. In his 1974 paper, “Structured Programming with Goto Statements”, he gave examples where he believed that a direct jump leads to clearer and more efficient code without sacrificing provability. Knuth proposed a looser structural constraint: It should be possible to draw a program's flow chart with all forward branches on the left, all backward branches on the right, and no branches crossing each other. Many of those knowledgeable in compilers and graph theory have advocated allowing only reducible flow graphs.

Structured programming theorists gained a major ally in the 1970s after IBM researcher Harlan Mills applied his interpretation of structured programming theory to the development of an indexing system for the *New York Times* research file. The project was a great engineering success, and managers at other companies cited it in support of adopting structured programming, although Dijkstra criticized the ways that Mills's interpretation differed from the published work.

As late as 1987 it was still possible to raise the question of structured programming in a computer science journal. Dijkstra did so in that year with a letter, "GOTO considered harmful." Numerous objections followed, including a response from Frank Rubin that sharply criticized both Dijkstra and the concessions other writers made when responding to him.

Outcome

By the end of the 20th century nearly all computer scientists were convinced that it is useful to learn and apply the concepts of structured programming. High-level programming languages that originally lacked programming structures, such as FORTRAN, COBOL, and BASIC, now have them.

Common deviations

While goto has now largely been replaced by the structured constructs of selection (if/then/else) and repetition (while and for), few languages are purely structured. The most common deviation, found in many languages, is the use of a return statement for early exit from a subroutine. This results in multiple exit points, instead of the single exit point required by structured programming. There are other constructions to handle cases that are awkward in purely structured programming.

Early exit

The most common deviation from structured programming is early exit from a function or loop. At the level of functions, this is a `return` statement. At the level of loops, this is a `break` statement (terminate the loop) or `continue` statement (terminate the current iteration, proceed with next iteration). In structured programming, these can be replicated by adding additional branches or test, but for returns from nested code this can add significant complexity. C is an early and prominent example of these constructs. Some newer languages also have "labeled breaks", which allow breaking out of more than just the innermost loop. Exceptions also allow early exit, but have further consequences, and thus are treated below.

Multiple exits can arise for a variety of reasons, most often either that the subroutine has no more work to do (if returning a value, it has completed the calculation), or has encountered "exceptional" circumstances that prevent it from continuing, hence needing exception handling.

The most common problem in early exit is that cleanup or final statements are not executed – for example, allocated memory is not unallocated, or open files are not closed, causing memory leaks or resource leaks. These must be done at each return site, which is brittle and can easily result in bugs. For instance, in later development, a return statement could be overlooked by a developer, and an action which should be performed at the end of a subroutine (e.g., a trace statement) might not be performed in all cases. Languages without a return statement, such as standard Pascal don't have this problem.

Most modern languages provide language-level support to prevent such leaks; see detailed discussion at resource management. Most commonly this is done via unwind protection, which ensures that certain code is guaranteed to be run when execution exits a block; this is a structured alternative to having a cleanup block and a `goto`. This is most often known as `try...finally,` and considered a part of exception handling. Various techniques exist to encapsulate resource management. An alternative approach, found primarily in C++, isResource Acquisition Is Initialization, which uses normal stack unwinding (variable deallocation) at function exit to call destructors on local variables to deallocate resources.

Kent Beck, Martin Fowler and co-authors have argued in their refactoring books that nested conditionals may be harder to understand than a certain type of flatter structure using multiple exits predicated by guard clauses. Their 2009 book flatly states that “one exit point is really not a useful rule. Clarity is the key principle: If the method is clearer with one exit point, use one exit point; otherwise don’t”. They offer a cookbook solution for transforming a function consisting only of nested conditionals into a sequence of guarded return (or throw) statements, followed by a single unguarded block, which is intended to contain the code for the common case, while the guarded statements are supposed to deal with the less common ones (or with errors). Herb Sutter and Andrei Alexandrescu also argue in their 2004 C++ tips book that the single-exit point is an obsolete requirement.

In his 2004 textbook, David Watt writes that “single-entry multi-exit control flows are often desirable”. Using Tennent’s framework notion of sequencer, Watt uniformly describes the control flow constructs found in contemporary programming languages and attempts to explain why certain types of sequencers are preferable to others in the context of multi-exit control flows. Watt writes that unrestricted gotos (jump sequencers) are a bad because the destination of the jump is not self-explanatory to the reader of a program until the reader finds and examines the actual label or address that is the target of the jump. In contrast, Watt argues that the conceptual intent of a return sequencer is clear from its own context, without having to examine its destination. Watt writes that a class of sequencers known as *escape sequencers*, defined as “sequencer that terminates execution of a textually enclosing command or procedure”, encompasses both breaks from loops (including multi-level breaks) and return statements. Watt also notes that while jump sequencers (gotos) have been somewhat restricted in languages like C, where the target must be an inside the local block or an encompassing outer block, that restriction alone is not sufficient to make the intent of gotos in C self-describing and so they can still produce “spaghetti code”. Watt also examines how exception sequencers differ from escape and jump sequencers; this is explained in the next section of this article.

In contrast to the above, Bertrand Meyer wrote in his 2009 textbook that instructions like `break` and `continue` “are just the old `goto` in sheep’s clothing” and strongly advised against their use.

Exception handling

Based on the coding error from the Ariane 501 disaster, software developer Jim Bonang argues that any exceptions thrown from a function violate the single-exit paradigm, and propose that all inter-procedural exceptions should be forbidden. In C++ syntax, this is done by declaring all function signatures as `throw()` Bonang proposes that all single-exit conforming C++ should be written along the lines of:

```
bool myCheck1() throw()
{
  bool success = false;
  try {
    // do something that may throw exceptions
    if(myCheck2() == false) {
      throw SomeInternalException();
    }
    // other code similar to the above
    success = true;
  }
  catch(...) { // all exceptions caught and logged
  }

  return success;

}
```

Peter Ritchie also notes that, in principle, even a single `throw` right before the `return` in a function constitutes a violation of the single-exit principle, but argues that Dijkstra's rules were written in a time before exception handling became a paradigm in programming languages, so he proposes to allow any number of throw points in addition to a single return point. He notes that solutions which wrap exceptions for the sake of creating a single-exit have higher nesting depth and thus are more difficult to comprehend, and even accuses those who propose to apply such solutions to programming languages which support exceptions of engaging in cargo cult thinking.

David Watt also analyzes exception handling in the framework of sequencers (introduced in this article in the previous section on early exits.) Watt notes that an abnormal situation (generally exemplified with arithmetic overflows or input/output failures like file not found) is a kind of error that "is detected in some low-level program unit, but [for which] a handler is more naturally located in a high-level program unit." For example, a program might contain several calls to read files, but the action to perform when a file is not found depends on the meaning (purpose) of the file in question to the program and thus a handling routine for this abnormal situation cannot be located in low-level system code. Watts further notes that introducing status flags testing in the caller, as single-exit structured programming or even (multi-exit) return sequencers would entail, results in a situation where "the application code tends to get cluttered by tests of status flags" and that "the programmer might forgetfully or lazily omit to test a status flag. In fact, abnormal situations represented by status flags are by default ignored!" He notes that in contrast to status flags testing, exceptions have the opposite default behavior, causing the program to terminate unless the programmer explicitly deals with the exception in some way, possibly by adding code to willfully ignore it. Based on these arguments, Watt concludes that jump sequencers or escape sequencers (discussed in the previous section) aren't as suitable as a dedicated exception sequencer with the semantics discussed above.

The textbook by Louden and Lambert emphasizes that exception handling differs from structured programming constructs like `while` loops because the transfer of control "is set up at a different point in the program than that where the actual transfer takes place. At the point where the transfer actually occurs, there may be no syntactic indication that control will in fact be transferred." Computer science professor Arvind Kumar Bansal also notes that in languages which implement exception handling, even control structures like `for`, which have the single-exit property in absence of exceptions, no longer have it in presence of exceptions, because an exception can prematurely cause an early exit in any part of the control structure; for instance if `init()` throws an exception in `for (init(); check(); increm())`, then the usual exit point after check() is not reached. Citing multiple prior studies by others (1999-2004) and their own results, Westley Weimer and George Necula wrote that a significant problem with exceptions is that they "create hidden control-flow paths that are difficult for programmers to reason about."

The necessity to limit code to single-exit points appears in some contemporary programming environments focused on parallel computing, such as OpenMP. The various parallel constructs from OpenMP, like `parallel do`, do not allow early exits from inside to the outside of the parallel construct; this restriction includes all manner of exits, from `break` to C++ exceptions, but all of these are permitted inside the parallel construct if the jump target is also inside it.

Multiple entry

More rarely, subprograms allow multiple *entry.* This is most commonly only *re*-entry into a coroutine (or generator/semicoroutine), where a subprogram yields control (and possibly a value), but can then be resumed where it left off. There are a number of common uses of such programming, notably for streams (particularly input/output), state machines, and concurrency. From a code execution point of view, yielding from a coroutine is closer to structured programming than returning from a subroutine, as the subprogram has not actually terminated, and will continue when called again—it is not an early exit. However, coroutines mean that multiple subprograms have execution state – rather than a single call stack of subroutines—and thus introduce a different form of complexity.

It is very rare for subprograms to allow entry to an arbitrary position in the subprogram, as in this case the program state (such as variable values) is uninitialized or ambiguous, and this is very similar to a goto.

State machines

Some programs, particularly parsers and communications protocols, have a number of states that follow each other in a way that is not easily reduced to the basic structures, and some programmers (including Knuth) implement the state-changes with a jump to the new state. This type of state-switching is often used in the Linux kernel.

However, it is possible to structure these systems by making each state-change a separate subprogram and using a variable to indicate the active state (see trampoline). Alternatively, these can be implemented via coroutines, which dispense with the trampoline.

Licensing & Attributions

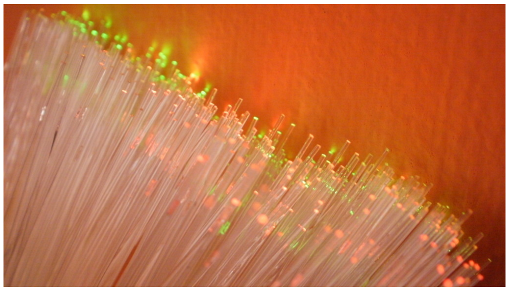
CC licensed content, Shared previously

- Structured Programming. **Provided by:** Wikipedia. **Located at:** https://en.wikipedia.org/wiki/Structured_programming. **License:** *CC BY-SA: Attribution-ShareAlike*
- Building. **Authored by:** Nick Hughes. **Located at:** https://www.flickr.com/photos/enhues/7735138488/. **License:** *CC BY: Attribution*
- Modular. **Authored by:** Goran Konjevod. **Located at:** https://www.flickr.com/photos/23913057@N05/3183895969/. **License:** *CC BY-NC: Attribution-NonCommercial*

MODULE 7: NETWORKS AND SECURITY

READING: NETWORKS & TELECOMMUNICATIONS

Introduction



One of the fastest growing technology areas is that of telecommunications (often referred to as telecoms). Organizations have realized that stand-alone computers present many problems: fragmentation of data, lack of control, insufficient integration and limited opportunity for teamwork. One of the major trends over the last decade has been the move not only to have a personal computer on the desk of virtually every knowledge worker, but to have that computer linked to the other computers in the organization.

This chapter deals with the basic telecommunication devices, the types of computer networks and the telecommunications services available in South Africa. We conclude with a discussion of arguably the most interesting development in information systems of the last decade: the Internet.

Computer Networks

When a numbers of computers are connected together, they form a computer network. There are many ways of classifying computer networks.

Networks according to size

Networks sizes can range from tiny to very large.

- Personal Area Network (PAN): consists of two to five computing devices. This not very common term would apply to the network typically found in the home, and may be based on wireless technology e.g. Bluetooth.
- Local Area Network (LAN): the most common type of network. It consists of from about four up to as many as a couple of hundred of computers linked together with one set of cables, usually within the same building. Most LANs are controlled by a central file server that takes care of network communications, security control and the storage of data files. A student computer laboratory typically constitutes one LAN.
- Metropolitan Area Network (MAN): a network infrastructure linking various local businesses within a large city area. This is now almost completely superseded by the Internet.
- Wide Area Network (WAN): the opposite of the LAN. It links computers over large geographical areas. This network usually makes use of the public telecommunications network. The widely dispersed Automatic Teller Machine (ATM) network of a commercial bank is typically part of the bank's WAN.
- Value-Added Network (VAN): although not relating to size (but it rhymes with the others!), it refers to the provision of a network infrastructure service to other businesses. The service goes beyond the physical cabling and includes "value-added services" such as limited data and transaction processing or message

routing. An example for the banking industry is the provision of an inter-bank Electronic Funds Transfer (EFT) and clearing service, linking the computers of different commercial banks (and, possibly, retailers) together.

Network Topologies

The network topology refers to the physical and logical way in which the computers in a network are connected together. Although there are a number of proprietary ways, the following three are the main topologies in common use (refer to Figure 1). Note that these topologies usually refer to a LAN configuration.

- The star network is driven by one central computer to and through which all other computers communicate. Although this allows for central co-ordination and control, it requires a very reliable central computer and lots of cables.
- The ring network consists of a continuous loop connecting all computers. Signals travel in a given direction and all computers have equal access to the data. A special version of the ring network is the token ring whereby a special code, the token, is passed around the ring. This token serves as the data holder and computers can send information only after grabbing an available i.e. empty token, adding their data and passing the token back onto the network.
- The bus network is currently the most popular configuration. A central data cable is used, to which each computer (and other devices such as printers and routers) can be attached. Although bottlenecks can occur, its popularity stems from its inherent robustness: devices can be added or removed without affecting the rest of the network. Data clashes (two computers attempting to send information simultaneously) can prevented by a variety of means.

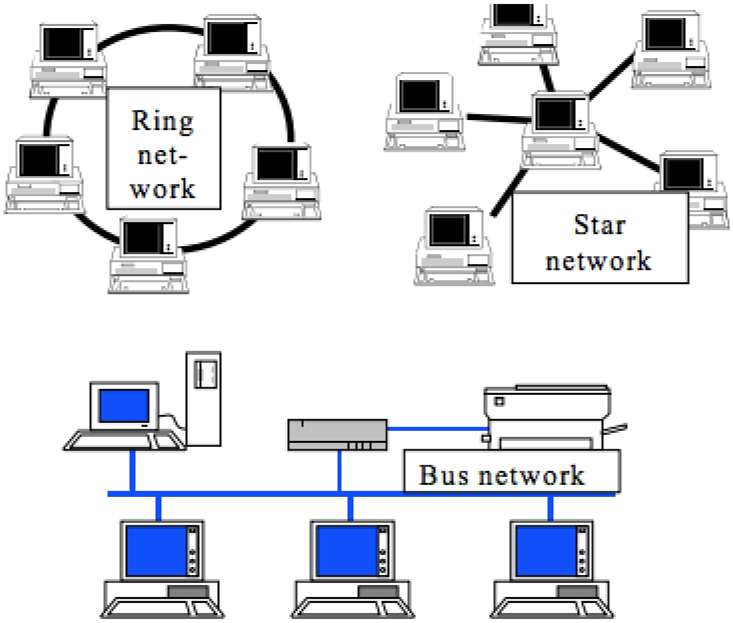


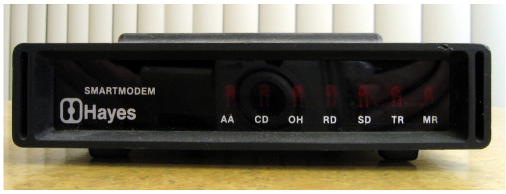
Figure 1: Network topologies

Telecommunication Devices

Regardless of the network topology that has been implemented, the same basic equipment is used to connect the different computers and to ensure error-free data transmission between them.

- Network cables are the physical wires by which computers are linked together. The most common types are:
    - Twisted pair: thin insulated copper wires, combined in one single cable. This is similar to the wire used for voice telephone connections.
    - Coaxial cable: (or coax) a thin copper wire inside a tube of insulation material, surrounded by a sheath or mesh of conducting wire, again insulated on the outside. This is similar to the wire

    - used to connect antennas to video or TV equipment. Because there is less possibility of interference, it allows greater volumes of data to be transmitted in a given time – the amount or volume of data that can be transmitted over a network connection is referred to as bandwidth.
    - Optical fibre or fibre-optic: a translucent and flexible material through which laser light can travel over long distances. This fibre is much more difficult to work: it requires special connectors as well as lasers and sensors (with electronics-to-light converters) at each terminal. Although this technology is more expensive, laser light can be switched on and off a lot faster than electricity (and it travels ten times faster), resulting again in a much greater bandwidth.
    - Wireless: not all computer devices need a physical cable connection. Because of the cabling costs and hassles, engineers have explored many methods of transmitting data without the use of wires. For short distances, infra-red signals work well albeit slowly – the same technology as your VCR remote control. For longer distances and higher bandwidths, radio frequencies or other parts of the electro-magnetic spectrum are used. Satellite technology is increasingly being used for digital data transmissions, especially in conjunction with Global Positioning Systems (GPS).
- Network interface cards (NICs) are necessary when computers are connected directly to other computers by means of digital network cables (as opposed to the situation when two computers are connected to each other via a telephone link). Their primary function is to make sure that there are no transmission conflicts with the other computers linked to the network, since data may be simultaneously sent and received by many different computers all linked to the same network. In addition, the network card usually fulfills an error-checking function, to ensure that uncorrupted data is received at its destination.
- Multiplexers allow a single channel to carry data transmissions simultaneously from many sources, by merging them at one end of the channel and then separating the individual transmissions at the receiving end of the channel.
- Front-end processors (FEP) are used in bigger networks that are centrally controlled by large computers – often mainframes. In order to give the expensive mainframe more "time" to concentrate on application processing, it needs to be relieved from the rather mundane task of network control. FEPs handle all or most communication processing such as error-checking, data conversion, packaging and transmission control.
- Routers and bridges are computers dedicated to the translation of network protocols and standards between different networks. They are becoming important as more and more organizations are linking their own networks to those of other organizations. They may be using
    - different operating systems (Novell, Unix or Windows NT),
    - other technologies (coax or fibre-optic),
    - or different protocols (proprietary or public standards set for computer communications).
- Finally, the modem allows a computer to communicate with another computer by means of the public voice telephone network, rather than by using digital cabling.

  The original model 300-baud Smartmodem

  This requires the conversion of digital computer signals (used inside the computer) into analogue sound signals (that can travel over the voice telephone lines) – this process is called modulation. At the other end of the line, these sound signals are converted back into digital signals – or demodulated. The word modem refers to this modulation/demodulation process. You may have heard this "modulated signal" when listening to a fax machine, which is really a scanner/printer/modem in one. Since the modem replaces the network card, it usually carries out similar error-checking functions to ensure the correct transmission of data.

Trivial fact: More than 5000 satellites are orbiting the earth and most of them are involved in telecommunications.

SA Public Telecommunications Services

Because telecommunication services are a critical part of any country's infrastructure, most governments have been very protective towards their telecoms. Paradoxically, this protectionism often resulted in high tariffs (monopolies!), thus reducing the overall competitiveness of local businesses. Recently more and more countries have started to privatize these services and allowed competition to drive prices down. The South African public telecommunication services are controlled by Telkom, although its legal monopoly is being phased out. The following are the main data network services provided by Telkom

Public Switched Telephone Network (PSTN) Services

The oldest data service provided by Telkom is the Datel service, which provides a connection between computers by means of the standard Public Switched Telephone Network (PSTN)—i.e., the same as the normal voice telephone traffic. This requires the use of built-in or external modem equipment that modulates the digital signal into an analogue audio signal (and demodulates it back at the receiving end). This service is quick and easy to set up since it is available anywhere where there is an ordinary voice telephone point. The main drawbacks are the limited transmission speed, high error rate and the lack of security. Customers may choose between a dial-up or leased line connection.

Diginet

Diginet is a dedicated digital data service from Telkom that provides reliable and efficient point-to-point (i.e. not switched) data connections. It differs from the Datel network in that the transmission path is entirely digital: a combination of fibre-optic, microwave and coaxial cable. Because the signal does not have to be translated into analogue form, no modem is required, resulting in a cost saving. However, its main advantages are the higher transmission rates and a substantial reduction in transmission errors. The standard Diginet service allows for 64 kpbs (kilobits per second) though an enhanced service called Diginet-Plus has been designed to transfer up to 1920 kpbs, which allows slow-scan TV and video conferencing signals to be transmitted in real-time.

Public Switched Data Network (PSDN) Services

Saponet is Telkom's Public Switched Data Network (PSDN). The Saponet-P service relies on a packet-switching mechanism whereby all data transmissions are broken up into smaller, standard-size units or data packets. Each of these packets is then routed independently to their destination. The path traveled by the packet depends on the available capacity and bottlenecks. At the destination, the original transmission is reassembled out of the constituent packets. A Packet Assembler & Disassembler (PAD) is responsible for the breaking up of a message into packets and the opposite process of reassembling packets into a message at the destination. This PAD can be a separate hardware device or a software program.

X.400 and Telkom400

Telkom400 is a VAN on top of the X.400 infrastructure. It supports electronic message handling and electronic data interchange (EDI). EDI is the automated computer to computer application exchange of structured business data between different organizations. An international standard defines common business documents such as order forms, invoices or electronic funds transfer documents that are exchanged directly between the computers of the respective business partners.

ISDN and ADSL

Now that most of Telkom's telephone exchanges have become digital, Telkom is able to provide new functions and services. One all-digital connection that has sufficient capacity (bandwidth) to support speech, video conferencing, facsimile, data and image transfer. This connection is called an ISDN line (Integrated Services Digital Network) and is currently available in selected metropolitan areas. Much more popular is the newer Asymmetric Digital Subscriber Line (ADSL) which allows a broadband connection (at least several hundred kilobits/second) over your standard telephone line while keeping the line available for voice telephone calls. It is called "asymmetrical" because the standard allows for much greater "download" than "upload" speeds; this reflects the typical home user pattern. Higher volumes and transmission speeds of up to 150 mbs – typically needed by mid-size and larger businesses – are available through Telkom's ATM Express service.

Licensing & Attributions

CC licensed content, Shared previously

- Discovering Information Systems Chapter 6.1-6.3. **Authored by**: Jean-Paul Van Belle, Mike Eccles, and Jane Nash. **Located at:** https://archive.org/details/ost-computer-science-discoveringinformationsystems. **License:** *CC BY-NC-ND: Attribution-NonCommercial-NoDerivatives*
- Smartmodem. **Authored by:** Michael Pereckas. **Located at:** https://en.wikipedia.org/wiki/Modem#/media/File:Hayes_300_Baud_Smartmodem_02.jpg. **License:** *CC BY-SA: Attribution-ShareAlike*
- Fiber Optic Cables. **Authored by:** Roshan Nikam. **Located at:** https://www.flickr.com/photos/31916678@N07/4753800195/. **License:** *CC BY: Attribution*

READING: NETWORKING

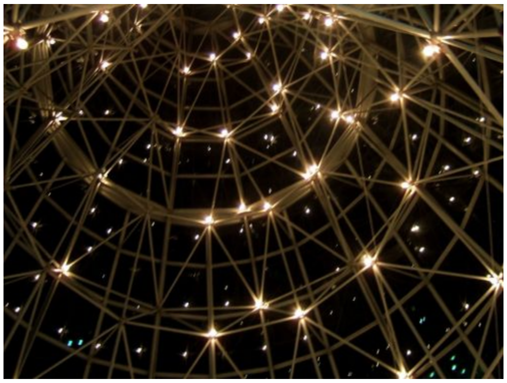
https://learn.saylor.org/course/view.php?id=94&sectionid=975

Licensing & Attributions

Public domain content

- License: *Public Domain: No Known Copyright*

READING: COMPUTER NETWORK



A **computer network** or **data network** is a telecommunications network which allows computers to exchange data. In computer networks, networked computing devices pass data to each other along network links (data connections). The connections between nodes are established using either cable media or wireless media. The best-known computer network is the Internet.

Network computer devices that originate, route and terminate the data are called network nodes. Nodes can include hosts such as personal computers, phones, servers as well as networking hardware. Two such devices can be said to be networked together when one device is able to exchange information with the other device, whether or not they have a direct connection to each other.

Computer networks differ in the transmission media used to carry their signals, the communications protocols to organize network traffic, the network's size, topology and organizational intent. In most cases, communications protocols are layered on (i.e. work using) other more specific or more general communications protocols, except for the *physical layer* that directly deals with the transmission media.

Computer networks support applications such as access to the World Wide Web, shared use of application and storage servers, printers, and fax machines, and use of email and instant messaging applications.

History

The chronology of significant computer-network developments includes:

- In the late 1950s early networks of communicating computers included the military radar system Semi-Automatic Ground Environment (SAGE).
- In 1959 Anatolii Ivanovich Kitov proposed to the Central Committee of the Communist Party of the Soviet Union a detailed plan for the re-organisation of the control of the Soviet armed forces and of the Soviet economy on the basis of a network of computing centres.
- In 1960 the commercial airline reservation system semi-automatic business research environment (SABRE) went online with two connected mainframes.
- In 1962 J.C.R. Licklider developed a working group he called the “Intergalactic Computer Network”, a precursor to the ARPANET, at the Advanced Research Projects Agency (ARPA).
- In 1964 researchers at Dartmouth College developed the Dartmouth Time Sharing System for distributed users of large computer systems. The same year, at Massachusetts Institute of Technology, a research group supported by General Electric and Bell Labs used a computer to route and manage telephone connections.

- Throughout the 1960s, Leonard Kleinrock, Paul Baran, and Donald Davies independently developed network systems that used packets to transfer information between computers over a network.
- In 1965, Thomas Marill and Lawrence G. Roberts created the first wide area network (WAN). This was an immediate precursor to the ARPANET, of which Roberts became program manager.
- Also in 1965, Western Electric introduced the first widely used telephone switch that implemented true computer control.
- In 1969 the University of California at Los Angeles, the Stanford Research Institute, the University of California at Santa Barbara, and the University of Utah became connected as the beginning of the ARPANET network using 50 kbit/s circuits.
- In 1972 commercial services using X.25 were deployed, and later used as an underlying infrastructure for expanding TCP/IP networks.
- In 1973, Robert Metcalfe wrote a formal memo at Xerox PARC describing Ethernet, a networking system that was based on the Aloha network, developed in the 1960s by Norman Abramson and colleagues at the University of Hawaii. In July 1976, Robert Metcalfe and David Boggs published their paper “Ethernet: Distributed Packet Switching for Local Computer Networks” and collaborated on several patents received in 1977 and 1978. In 1979 Robert Metcalfe pursued making Ethernet an open standard.
- In 1976 John Murphy of Datapoint Corporation created ARCNET, a token-passing network first used to share storage devices.
- In 1995 the transmission speed capacity for Ethernet increased from 10 Mbit/s to 100 Mbit/s. By 1998, Ethernet supported transmission speeds of a Gigabit. The ability of Ethernet to scale easily (such as quickly adapting to support new fiber optic cable speeds) is a contributing factor to its continued use as of 2015.

Properties

Computer networking may be considered a branch of electrical engineering, telecommunications, computer science, information technology or computer engineering, since it relies upon the theoretical and practical application of the related disciplines.

A computer network facilitates interpersonal communications allowing people to communicate efficiently and easily via email, instant messaging, chat rooms, telephone, video telephone calls, and video conferencing. Providing access to information on shared storage devices is an important feature of many networks. A network allows sharing of files, data, and other types of information giving authorized users the ability to access information stored on other computers on the network. A network allows sharing of network and computing resources. Users may access and use resources provided by devices on the network, such as printing a document on a shared network printer. Distributed computing uses computing resources across a network to accomplish tasks. A computer network may be used by computer crackers to deploy computer viruses or computer worms on devices connected to the network, or to prevent these devices from accessing the network via a denial of service attack.

Network packet

Computer communication links that do not support packets, such as traditional point-to-point telecommunication links, simply transmit data as a bit stream. However, most information in computer networks is carried in *packets*. A network packet is a formatted unit of data (a list of bits or bytes, usually a few tens of bytes to a few kilobytes long) carried by a packet-switched network.

In packet networks, the data is formatted into packets that are sent through the network to their destination. Once the packets arrive they are reassembled into their original message. With packets, the bandwidth of the transmission medium can be better shared among users than if the network were circuit switched. When one user is not sending packets, the link can be filled with packets from others users, and so the cost can be shared, with relatively little interference, provided the link isn’t overused.

Packets consist of two kinds of data: control information and user data (also known as payload). The control information provides data the network needs to deliver the user data, for example: source and destination network addresses, error detection codes, and sequencing information. Typically, control information is found in packet headers and trailers, with payload data in between.

Often the route a packet needs to take through a network is not immediately available. In that case the packet is queued and waits until a link is free.

Network topology

The physical layout of a network is usually less important than the topology that connects network nodes. Most diagrams that describe a physical network are therefore topological, rather than geographic. The symbols on these diagrams usually denote network links and network nodes.

Network links

The transmission media (often referred to in the literature as the *physical media*) used to link devices to form a computer network include electrical cable(Ethernet, HomePNA, power line communication, G.hn), optical fiber (fiber-optic communication), and radio waves (wireless networking). In the OSI model, these are defined at layers 1 and 2 — the physical layer and the data link layer.

A widely adopted *family* of transmission media used in local area network (LAN) technology is collectively known as Ethernet. The media and protocol standards that enable communication between networked devices over Ethernet are defined by IEEE 802.3. Ethernet transmits data over both copper and fiber cables. Wireless LAN standards (e.g. those defined by IEEE 802.11) use radio waves, or others use infrared signals as a transmission medium.Power line communication uses a building's power cabling to transmit data.

Wired technologies

The orders of the following wired technologies are, roughly, from slowest to fastest transmission speed.

Fiber optic cables are used to transmit light from one computer/network node to another.

- *Coaxial cable* is widely used for cable television systems, office buildings, and other work-sites for local area networks. The cables consist of copper or aluminum wire surrounded by an insulating layer (typically a flexible material with a high dielectric constant), which itself is surrounded by a conductive layer. The insulation helps minimize interference and distortion. Transmission speed ranges from 200 million bits per second to more than 500 million bits per second.
- ITU-T G.hn technology uses existing home wiring (coaxial cable, phone lines and power lines) to create a high-speed (up to 1 Gigabit/s) local area network
- *Twisted pair wire* is the most widely used medium for all telecommunication. Twisted-pair cabling consist of copper wires that are twisted into pairs. Ordinary telephone wires consist of two insulated copper wires twisted into pairs. Computer network cabling (wired Ethernet as defined by IEEE 802.3) consists of 4 pairs of copper cabling that can be utilized for both voice and data transmission. The use of two wires twisted together helps to reduce crosstalk and electromagnetic induction. The transmission speed ranges from 2 million bits per second to 10 billion bits per second. Twisted pair cabling comes in two forms: unshielded twisted pair (UTP) and shielded twisted-pair (STP). Each form comes in several category ratings, designed for use in various scenarios.
- An *optical fiber* is a glass fiber. It carries pulses of light that represent data. Some advantages of optical fibers over metal wires are very low transmission loss and immunity from electrical interference. Optical fibers can simultaneously carry multiple wavelengths of light, which greatly increases the rate that data can be sent, and helps enable data rates of up to trillions

of bits per second. Optic fibers can be used for long runs of cable carrying very high data rates, and are used for undersea cables to interconnect continents.

Price is a main factor distinguishing wired- and wireless-technology options in a business. Wireless options command a price premium that can make purchasing wired computers, printers and other devices a financial benefit. Before making the decision to purchase hard-wired technology products, a review of the restrictions and limitations of the selections is necessary. Business and employee needs may override any cost considerations.

Wireless technologies

Computers are very often connected to networks using wireless links.

- *Terrestrial microwave*—Terrestrial microwave communication uses Earth-based transmitters and receivers resembling satellite dishes. Terrestrial microwaves are in the low-gigahertz range, which limits all communications to line-of-sight. Relay stations are spaced approximately 48 km (30 mi) apart.
- *Communications satellites*—Satellites communicate via microwave radio waves, which are not deflected by the Earth's atmosphere. The satellites are stationed in space, typically in geosynchronous orbit 35,400 km (22,000 mi) above the equator. These Earth-orbiting systems are capable of receiving and relaying voice, data, and TV signals.
- *Cellular and PCS systems* use several radio communications technologies. The systems divide the region covered into multiple geographic areas. Each area has a low-power transmitter or radio relay antenna device to relay calls from one area to the next area.
- *Radio and spread spectrum technologies*—Wireless local area networks use a high-frequency radio technology similar to digital cellular and a low-frequency radio technology. Wireless LANs use spread spectrum technology to enable communication between multiple devices in a limited area. IEEE 802.11 defines a common flavor of open-standards wireless radio-wave technology known as Wifi.
- *Free-space optical communication* uses visible or invisible light for communications. In most cases, line-of-sight propagation is used, which limits the physical positioning of communicating devices.

Exotic technologies

There have been various attempts at transporting data over exotic media:

- IP over Avian Carriers was a humorous April fool's Request for Comments, issued as RFC 1149. It was implemented in real life in 2001.
- Extending the Internet to interplanetary dimensions via radio waves.

Both cases have a large round-trip delay time, which gives slow two-way communication, but doesn't prevent sending large amounts of information.

Network nodes

Apart from any physical transmission medium there may be, networks comprise additional basic system building blocks, such as network interface controller (NICs), repeaters, hubs, bridges, switches, routers, modems, and firewalls.

Network interfaces

A network interface controller (NIC) is computer hardware that provides a computer with the ability to access the transmission media, and has the ability to process low-level network information. For example the NIC may have a connector for accepting a cable, or an aerial for wireless transmission and reception, and the associated circuitry.

The NIC responds to traffic addressed to a network address for either the NIC or the computer as a whole.

In Ethernet networks, each network interface controller has a unique Media Access Control (MAC) address—usually stored in the controller's permanent memory. To avoid address conflicts between network devices, the Institute of Electrical and Electronics Engineers (IEEE) maintains and administers MAC address uniqueness. The size of an Ethernet MAC address is six octets. The three most significant octets are reserved to identify NIC manufacturers. These manufacturers, using only their assigned prefixes, uniquely assign the three least-significant octets of every Ethernet interface they produce.

An ATM network interface in the form of an accessory card. A lot of network interfaces are built-in.

Repeaters and hubs

A repeater is an electronic device that receives a network signal, cleans it of unnecessary noise, and regenerates it. The signal is retransmitted at a higher power level, or to the other side of an obstruction, so that the signal can cover longer distances without degradation. In most twisted pair Ethernet configurations, repeaters are required for cable that runs longer than 100 meters. With fiber optics, repeaters can be tens or even hundreds of kilometers apart.

A repeater with multiple ports is known as a hub. Repeaters work on the physical layer of the OSI model. Repeaters require a small amount of time to regenerate the signal. This can cause a propagation delay that affects network performance. As a result, many network architectures limit the number of repeaters that can be used in a row, e.g., the Ethernet 5-4-3 rule.

Hubs have been mostly obsoleted by modern switches; but repeaters are used for long distance links, notably undersea cabling.

Bridges

A network bridge connects and filters traffic between two network segments at the data link layer (layer 2) of the OSI model to form a single network. This breaks the network's collision domain but maintains a unified broadcast domain. Network segmentation breaks down a large, congested network into an aggregation of smaller, more efficient networks.

Bridges come in three basic types:

- Local bridges: Directly connect LANs
- Remote bridges: Can be used to create a wide area network (WAN) link between LANs. Remote bridges, where the connecting link is slower than the end networks, largely have been replaced with routers.
- Wireless bridges: Can be used to join LANs or connect remote devices to LANs.

Switches

A network switch is a device that forwards and filters OSI layer 2 datagrams between ports based on the MAC addresses in the packets. A switch is distinct from a hub in that it only forwards the frames to the physical ports involved in the communication rather than all ports connected. It can be thought of as a multi-port bridge. It learns to associate physical ports to MAC addresses by examining the source addresses of received frames. If an unknown destination is targeted, the switch broadcasts to all ports but the source. Switches normally have numerous ports, facilitating a star topology for devices, and cascading additional switches.

Multi-layer switches are capable of routing based on layer 3 addressing or additional logical levels. The term *switch* is often used loosely to include devices such as routers and bridges, as well as devices that may distribute traffic based on load or based on application content (e.g., a Web URL identifier).

Routers

A router is an internet working device that forwards packets between networks by processing the routing information included in the packet or datagram (Internet protocol information from layer 3). The routing information is often processed in conjunction with the routing table (or forwarding table). A router uses its routing table to determine where to forward packets. (A destination in a routing table can include a “null” interface, also known as the “black hole” interface because data can go into it, however, no further processing is done for said data.)

A typical home or small office router showing the ADSL telephone line and Ethernet network cable connections

Modems

Modems (MOdulator-DEModulator) are used to connect network nodes via wire not originally designed for digital network traffic, or for wireless. To do this one or more carrier signals are modulated by the digital signal to produce an analog signal that can be tailored to give the required properties for transmission. Modems are commonly used for telephone lines, using a Digital Subscriber Line technology.

Firewalls

A firewall is a network device for controlling network security and access rules. Firewalls are typically configured to reject access requests from unrecognized sources while allowing actions from recognized ones. The vital role firewalls play in network security grows in parallel with the constant increase in cyber attacks.

Network structure

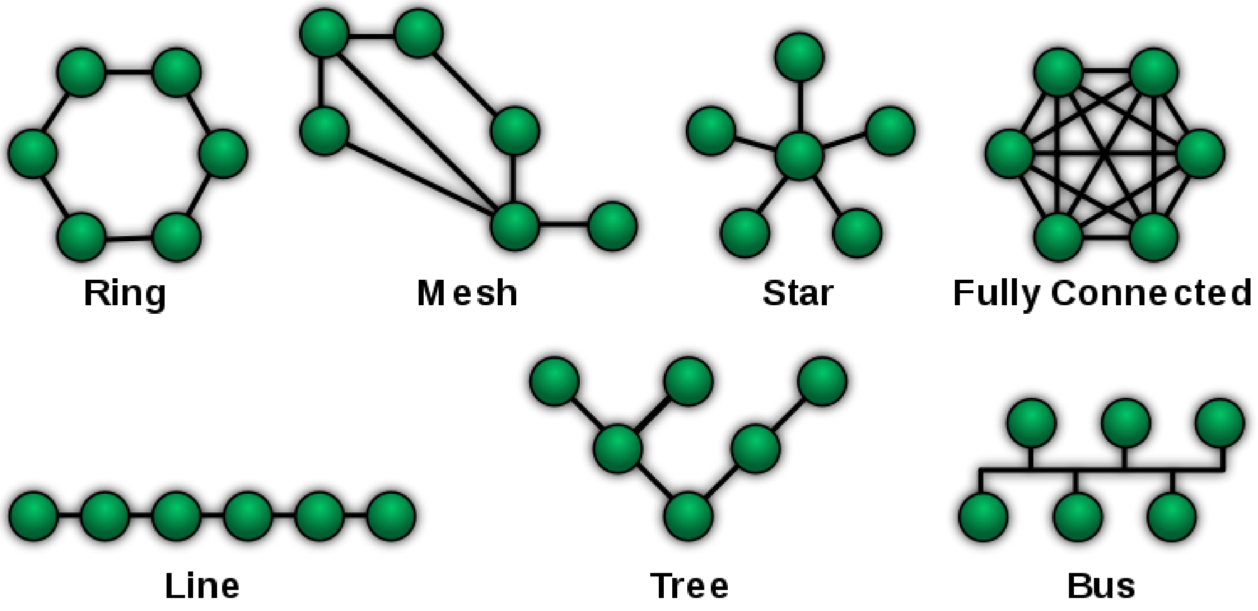
Network topology is the layout or organizational hierarchy of interconnected nodes of a computer network. Different network topologies can affect throughput, but reliability is often more critical. With many technologies, such as bus networks, a single failure can cause the network to fail entirely. In general the more interconnections there are, the more robust the network is; but the more expensive it is to install.

Common layouts

Common layouts are:

- A bus network: all nodes are connected to a common medium along this medium. This was the layout used in the original Ethernet, called 10BASE5 and 10BASE2.
- A star network: all nodes are connected to a special central node. This is the typical layout found in a Wireless LAN, where each wireless client connects to the central Wireless access point.
- A ring network: each node is connected to its left and right neighbor node, such that all nodes are connected and that each node can reach each other node by traversing nodes left- or rightwards. The Fiber Distributed Data Interface (FDDI) made use of such a topology.
- A mesh network: each node is connected to an arbitrary number of neighbours in such a way that there is at least one traversal from any node to any other.
- A fully connected network: each node is connected to every other node in the network.

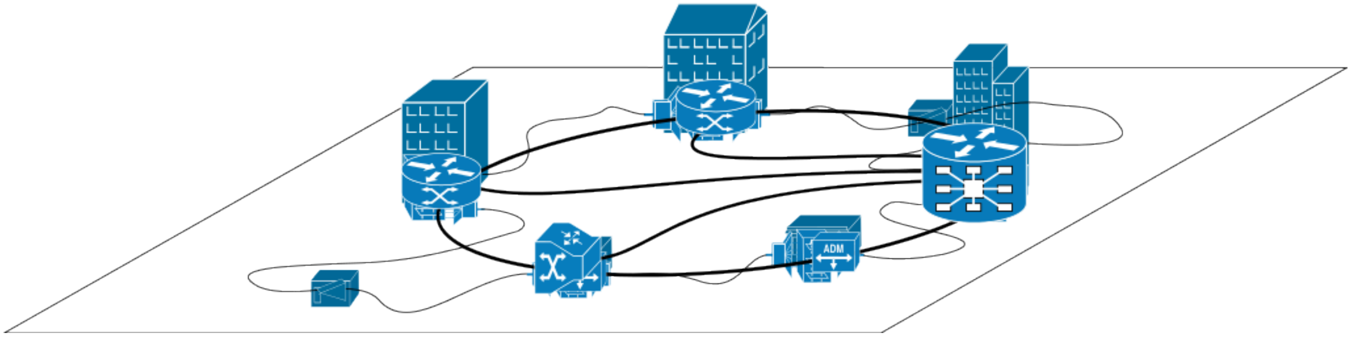
- A tree network: nodes are arranged hierarchically.



Common network topologies

Note that the physical layout of the nodes in a network may not necessarily reflect the network topology. As an example, with FDDI, the network topology is a ring (actually two counter-rotating rings), but the physical topology is often a star, because all neighboring connections can be routed via a central physical location.

Overlay network



A sample overlay network.

An overlay network is a virtual computer network that is built on top of another network. Nodes in the overlay network are connected by virtual or logical links. Each link corresponds to a path, perhaps through many physical links, in the underlying network. The topology of the overlay network may (and often does) differ from that of the underlying one. For example, many peer-to-peer networks are overlay networks. They are organized as nodes of a virtual system of links that run on top of the Internet.

Overlay networks have been around since the invention of networking when computer systems were connected over telephone lines using modems, before any data network existed.

The most striking example of an overlay network is the Internet itself. The Internet itself was initially built as an overlay on the telephone network. Even today, at the network layer, each node can reach any other by a direct connection to the desired IP address, thereby creating a fully connected network. The underlying network, however, is composed of a mesh-like interconnect of sub-networks of varying topologies (and technologies).

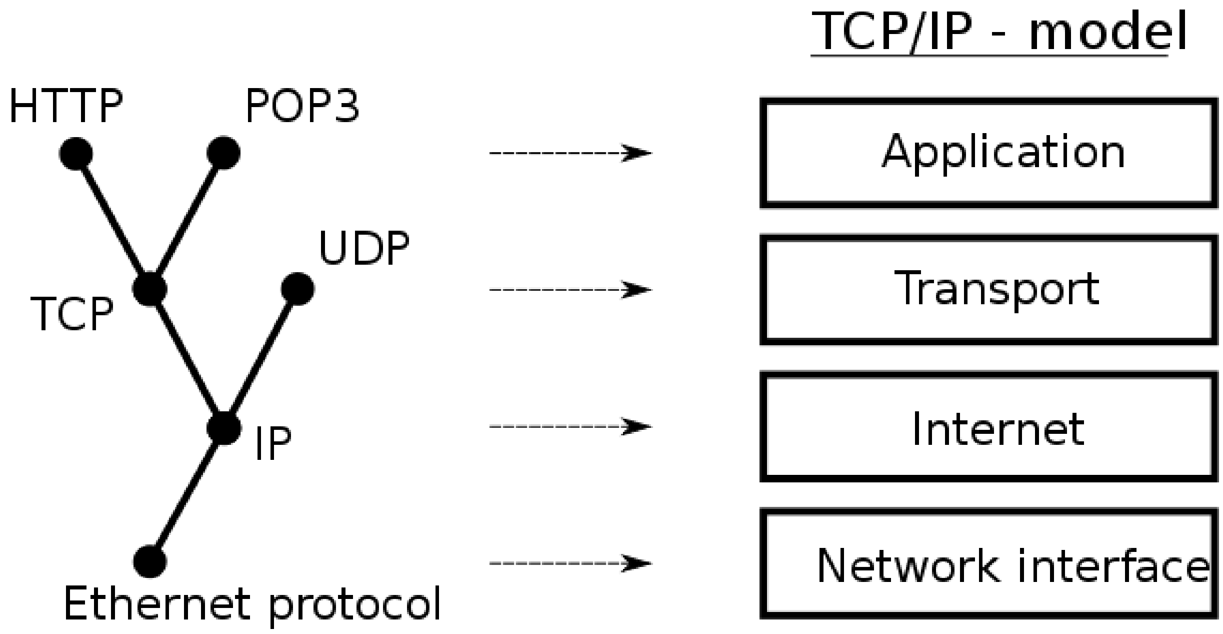
Address resolution and routing are the means that allow mapping of a fully connected IP overlay network to its underlying network.

Another example of an overlay network is a distributed hash table, which maps keys to nodes in the network. In this case, the underlying network is an IP network, and the overlay network is a table (actually a map) indexed by keys.

Overlay networks have also been proposed as a way to improve Internet routing, such as through quality of service guarantees to achieve higher-quality streaming media. Previous proposals such as IntServ, DiffServ, and IP Multicast have not seen wide acceptance largely because they require modification of all routers in the network. On the other hand, an overlay network can be incrementally deployed on end-hosts running the overlay protocol software, without cooperation from Internet service providers. The overlay network has no control over how packets are routed in the underlying network between two overlay nodes, but it can control, for example, the sequence of overlay nodes that a message traverses before it reaches its destination.

For example, Akamai Technologies manages an overlay network that provides reliable, efficient content delivery (a kind of multicast). Academic research includes end system multicast, resilient routing and quality of service studies, among others.

Communications protocols



The TCP/IP model or Internet layering scheme and its relation to common protocols often layered on top of it.

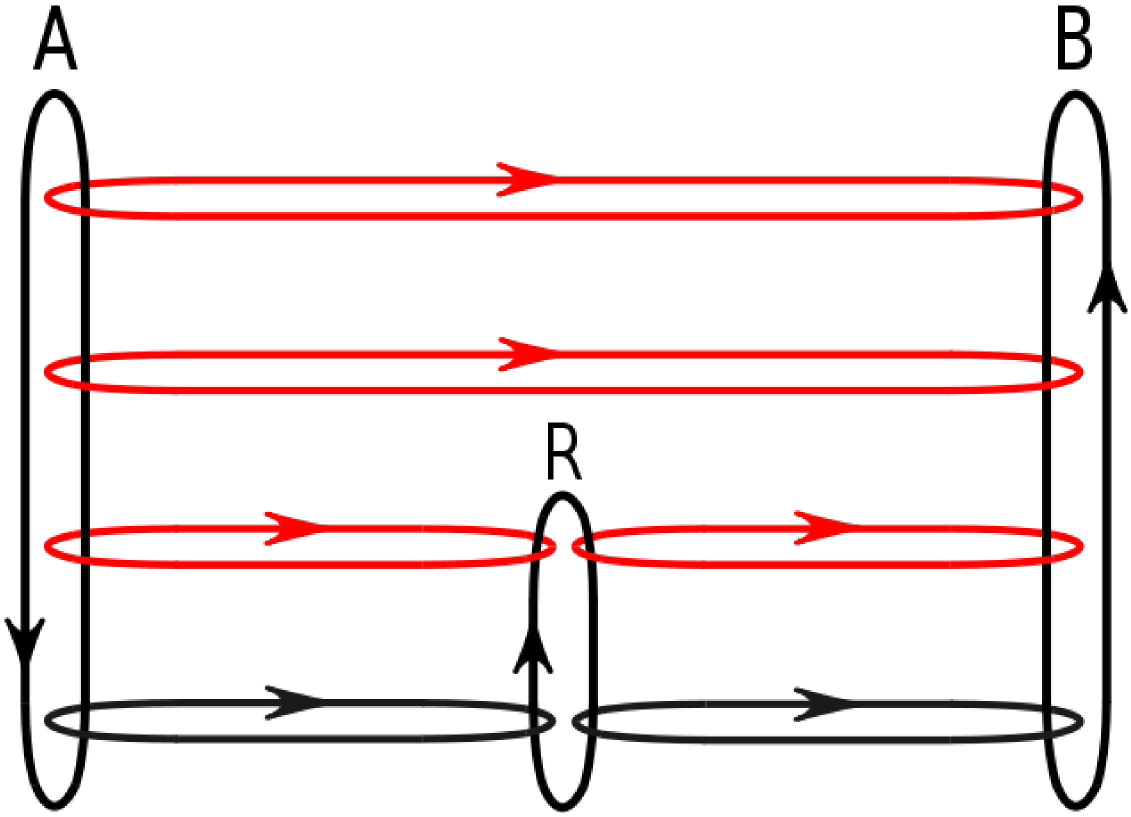


Figure 4. Message flows (A-B) in the presence of a router (R), red flows are effective communication paths, black paths are the actual paths.

A communications protocol is a set of rules for exchanging information over network links. In a protocol stack(also see the OSI model), each protocol leverages the services of the protocol below it. An important example of a protocol stack is HTTP (the World Wide Web protocol) running over TCP over IP (the Internet protocols) over IEEE 802.11 (the Wi-Fi protocol). This stack is used between the wireless router and the home user's personal computer when the user is surfing the web.

Whilst the use of protocol layering is today ubiquitous across the field of computer networking, it has been historically criticized by many researchers for two principal reasons. Firstly, abstracting the protocol stack in this way may cause a higher layer to duplicate functionality of a lower layer, a prime example being error recovery on both a per-link basis and an end-to-end basis. Secondly, it is common that a protocol implementation at one layer may require data, state or addressing information that is only present at another layer, thus defeating the point of separating the layers in the first place. For example, TCP uses the ECN field in the IPv4 header as an indication of congestion; IP is a network layer protocol whereas TCP is a transport layer protocol.

Communication protocols have various characteristics. They may be connection-oriented or connectionless, they may use circuit mode or packet switching, and they may use hierarchical addressing or flat addressing.

There are many communication protocols, a few of which are described below.

IEEE 802

The complete IEEE 802 protocol suite provides a diverse set of networking capabilities. The protocols have a flat addressing scheme. They operate mostly at levels 1 and 2 of the OSI model.

For example, MAC bridging (IEEE 802.1D) deals with the routing of Ethernet packets using a Spanning Tree Protocol. IEEE 802.1Q describes VLANs, and IEEE 802.1X defines a port-based Network Access Control protocol, which forms the basis for the authentication mechanisms used in VLANs (but it is also found in WLANs)—it is what the home user sees when the user has to enter a “wireless access key.”

Ethernet

Ethernet, sometimes simply called *LAN*, is a family of protocols used in wired LANs, described by a set of standards together called IEEE 802.3 published by the Institute of Electrical and Electronics Engineers.

Wireless LAN

Wireless LAN, also widely known as WLAN or WiFi, is probably the most well-known member of the IEEE 802 protocol family for home users today. It is standarized by IEEE 802.11 and shares many properties with wired Ethernet.

Internet Protocol Suite

The Internet Protocol Suite, also called TCP/IP, is the foundation of all modern networking. It offers connection-less as well as connection-oriented services over an inherently unreliable network traversed by data-gram transmission at the Internet protocol (IP) level. At its core, the protocol suite defines the addressing, identification, and routing specifications for Internet Protocol Version 4 (IPv4) and for IPv6, the next generation of the protocol with a much enlarged addressing capability.

SONET/SDH

Synchronous optical networking (SONET) and Synchronous Digital Hierarchy (SDH) are standardized multiplexing protocols that transfer multiple digital bit streams over optical fiber using lasers. They were originally designed to transport circuit mode communications from a variety of different sources, primarily to support real-time, uncompressed, circuit-switched voice encoded in PCM (Pulse-Code Modulation) format. However, due to its protocol neutrality and transport-oriented features, SONET/SDH also was the obvious choice for transporting Asynchronous Transfer Mode (ATM) frames.

Asynchronous Transfer Mode

Asynchronous Transfer Mode (ATM) is a switching technique for telecommunication networks. It uses asynchronous time-division multiplexing and encodes data into small, fixed-sized cells. This differs from other protocols such as the Internet Protocol Suite or Ethernet that use variable sized packets or frames. ATM has similarity with both circuit and packet switched networking. This makes it a good choice for a network that must handle both traditional high-throughput data traffic, and real-time, low-latency content such as voice and video. ATM uses a connection-oriented model in which a virtual circuitmust be established between two endpoints before the actual data exchange begins.

While the role of ATM is diminishing in favor of next-generation networks, it still plays a role in the last mile, which is the connection between an Internet service provider and the home user.

Geographic scale

A network can be characterized by its physical capacity or its organizational purpose. Use of the network, including user authorization and access rights, differ accordingly.

A nanoscale communication network has key components implemented at the nanoscale including message carriers and leverages physical principles that differ from macroscale communication mechanisms. Nanoscale communication extends communication to very small sensors and actuators such as those found in biological systems and also tends to operate in environments that would be too harsh for classical communication.

A personal area network (PAN) is a computer network used for communication among computer and different information technological devices close to one person. Some examples of devices that are used in a PAN are personal computers, printers, fax machines, telephones, PDAs, scanners, and even video game consoles. A PAN may include wired and wireless devices. The reach of a PAN typically extends to 10 meters. A wired PAN is usually constructed with USB and FireWire connections while technologies such as Bluetooth and infrared communication typically form a wireless PAN.

A local area network (LAN) is a network that connects computers and devices in a limited geographical area such as a home, school, office building, or closely positioned group of buildings. Each computer or device on the network is a node. Wired LANs are most likely based on Ethernet technology. Newer standards such as ITU-T G.hn also provide a way to create a wired LAN using existing wiring, such as coaxial cables, telephone lines, and power lines.

All interconnected devices use the network layer (layer 3) to handle multiple subnets (represented by different colors). Those inside the library have 10/100 Mbit/s Ethernet connections to the user device and a Gigabit Ethernet connection to the central router. They could be called *Layer 3 switches*, because they only have Ethernet interfaces and support the Internet Protocol. It might be more correct to call them access routers, where the router at the top is a distribution router that connects to the Internet and to the academic networks' customer access routers.

The defining characteristics of a LAN, in contrast to a wide area network (WAN), include higher data transfer rates, limited geographic range, and lack of reliance on leased lines to provide connectivity. Current Ethernet or other IEEE 802.3 LAN technologies operate at data transfer rates up to 10 Gbit/s. The IEEE investigates the standardization of 40 and 100 Gbit/s rates. A LAN can be connected to a WAN using a router.

A home area network (HAN) is a residential LAN used for communication between digital devices typically deployed in the home, usually a small number of personal computers and accessories, such as printers and mobile computing devices. An important function is the sharing of Internet access, often a broadband service through a cable TV or digital subscriber line (DSL) provider.

A storage area network (SAN) is a dedicated network that provides access to consolidated, block level data storage. SANs are primarily used to make storage devices, such as disk arrays, tape libraries, and optical jukeboxes, accessible to servers so that the devices appear like locally attached devices to the operating system. A SAN typically has its own network of storage devices that are generally not accessible through the local area network by other devices. The cost and complexity of SANs dropped in the early 2000s to levels allowing wider adoption across both enterprise and small to medium-sized business environments.

A campus area network (CAN) is made up of an interconnection of LANs within a limited geographical area. The networking equipment (switches, routers) and transmission media (optical fiber, copper plant, Cat5 cabling, etc.) are almost entirely owned by the campus tenant/owner (an enterprise, university, government, etc.).

For example, a university campus network is likely to link a variety of campus buildings to connect academic colleges or departments, the library, and student residence halls.

A backbone network is part of a computer network infrastructure that provides a path for the exchange of information between different LANs or sub-networks. A backbone can tie together diverse networks within the same building, across different buildings, or over a wide area.

For example, a large company might implement a backbone network to connect departments that are located around the world. The equipment that ties together the departmental networks constitutes the network backbone. When designing a network backbone, network performance and network congestionare critical factors to take into account. Normally, the backbone network's capacity is greater than that of the individual networks connected to it.

Another example of a backbone network is the Internet backbone, which is the set of wide area networks (WANs) and core routers that tie together all networks connected to the Internet.

A Metropolitan area network (MAN) is a large computer network that usually spans a city or a large campus.

A wide area network (WAN) is a computer network that covers a large geographic area such as a city, country, or spans even intercontinental distances. A WAN uses a communications channel that combines many types of media such as telephone lines, cables, and air waves. A WAN often makes use of transmission facilities provided by common carriers, such as telephone companies. WAN technologies generally function at the lower three layers of the OSI reference model: the physical layer, the data link layer, and the network layer.

An enterprise private network is a network that a single organization builds to interconnect its office locations (e.g., production sites, head offices, remote offices, shops) so they can share computer resources.

A virtual private network (VPN) is an overlay network in which some of the links between nodes are carried by open connections or virtual circuits in some larger network (e.g., the Internet) instead of by physical wires. The data link layer protocols of the virtual network are said to be tunneled through the larger network when this is the case. One common application is secure communications through the public Internet, but a VPN need not have explicit security features, such as authentication or content encryption. VPNs, for example, can be used to separate the traffic of different user communities over an underlying network with strong security features.

VPN may have best-effort performance, or may have a defined service level agreement (SLA) between the VPN customer and the VPN service provider. Generally, a VPN has a topology more complex than point-to-point.

A global area network (GAN) is a network used for supporting mobile across an arbitrary number of wireless LANs, satellite coverage areas, etc. The key challenge in mobile communications is handing off user communications from one local coverage area to the next. In IEEE Project 802, this involves a succession of terrestrial wireless LANs.

Organizational scope

Networks are typically managed by the organizations that own them. Private enterprise networks may use a combination of intranets and extranets. They may also provide network access to the Internet, which has no single owner and permits virtually unlimited global connectivity.

Intranets

An intranet is a set of networks that are under the control of a single administrative entity. The intranet uses the IP protocol and IP-based tools such as web browsers and file transfer applications. The administrative entity limits use of the intranet to its authorized users. Most commonly, an intranet is the internal LAN of an organization. A large intranet typically has at least one web server to provide users with organizational information. An intranet is also anything behind the router on a local area network.

Extranet

An extranet is a network that is also under the administrative control of a single organization, but supports a limited connection to a specific external network. For example, an organization may provide access to some aspects of its intranet to share data with its business partners or customers. These other entities are not necessarily trusted from a security standpoint. Network connection to an extranet is often, but not always, implemented via WAN technology.

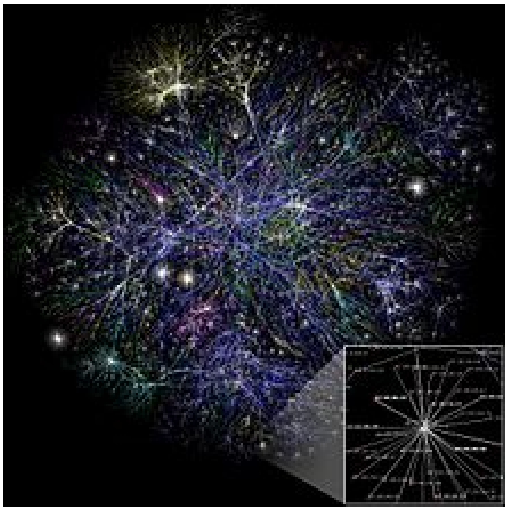
Internetwork

An internetwork is the connection of multiple computer networks via a common routing technology using routers.

Internet

The Internet is the largest example of an internetwork. It is a global system of interconnected governmental, academic, corporate, public, and private computer networks. It is based on the networking technologies of theInternet Protocol Suite. It is the successor of the Advanced Research Projects Agency Network (ARPANET) developed by DARPA of the United States Department of Defense. The Internet is also the communications backbone underlying the World Wide Web (WWW).

Participants in the Internet use a diverse array of methods of several hundred documented, and often standardized, protocols compatible with the Internet Protocol Suite and an addressing system (IP addresses) administered by the Internet Assigned Numbers Authority and address registries. Service providers and large enterprises exchange information about the reachability of their address spaces through the Border Gateway Protocol (BGP), forming a redundant worldwide mesh of transmission paths.



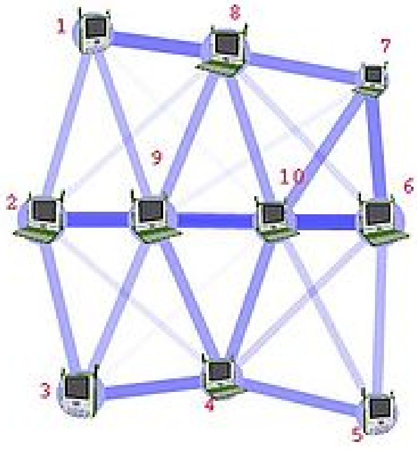
Partial map of the Internet based on the January 15, 2005 data found on opte.org. Each line is drawn between two nodes, representing two IP addresses. The length of the lines are indicative of the delay between those two nodes. This graph represents less than 30% of the Class C networks reachable.

Darknet

A Darknet is an overlay network, typically running on the internet, that is only accessible through specialized software. A darknet is an anonymizing network where connections are made only between trusted peers — sometimes called “friends” (F2F) — using non-standard protocols and ports.

Darknets are distinct from other distributed peer-to-peer networks as sharing is anonymous (that is, IP addresses are not publicly shared), and therefore users can communicate with little fear of governmental or corporate interference.

Routing



Routing is the process of selecting network paths to carry network traffic. Routing is performed for many kinds of networks, including circuit switching networks and packet switched networks.

Routing calculates good paths through a network for information to take. For example from node 1 to node 6 the best routes are likely to be 1-8-7-6 or 1-8-10-6, as this has the thickest routes.

In packet switched networks, routing directs packet forwarding (the transit of logically addressed network packets from their source toward their ultimate destination) through intermediate nodes. Intermediate nodes are typically network hardware devices such as routers, bridges, gateways, firewalls, or switches. General-purpose computers can also forward packets and perform routing, though they are not specialized hardware and may suffer from limited performance. The routing process usually directs forwarding on the basis of routing tables, which maintain a record of the routes to various network destinations. Thus, constructing routing tables, which are held in the router's memory, is very important for efficient routing. Most routing algorithms use only one network path at a time. Multipath routing techniques enable the use of multiple alternative paths.

There are usually multiple routes that can be taken, and to choose between them, different elements can be considered to decide which routes get installed into the routing table, such as (sorted by priority):

1. *Prefix-Length*: where longer subnet masks are preferred (independent if it is within a routing protocol or over different routing protocol)
2. *Metric*: where a lower metric/cost is preferred (only valid within one and the same routing protocol)
3. *Administrative distance*: where a lower distance is preferred (only valid between different routing protocols)

Routing, in a more narrow sense of the term, is often contrasted with bridging in its assumption that network addresses are structured and that similar addresses imply proximity within the network. Structured addresses allow a single routing table entry to represent the route to a group of devices. In large networks, structured addressing (routing, in the narrow sense) outperforms unstructured addressing (bridging). Routing has become the dominant form of addressing on the Internet. Bridging is still widely used within localized environments.

Network service

Network services are applications hosted by servers on a computer network, to provide some functionality for members or users of the network, or to help the network itself to operate.

The World Wide Web, E-mail, printing and network file sharing are examples of well-known network services. Network services such as DNS (Domain Name System) give names for IP and MAC addresses (people remember names like "nm.lan" better than numbers like "210.121.67.18"), and DHCP to ensure that the equipment on the network has a valid IP address.

Services are usually based on a service protocol that defines the format and sequencing of messages between clients and servers of that network service.

Network performance

Quality of service

Depending on the installation requirements, network performance is usually measured by the quality of service of a telecommunications product. The parameters that affect this typically can include throughput, jitter, bit error rate and latency.

The following list gives examples of network performance measures for a circuit-switched network and one type of packet-switched network, viz. ATM:

- Circuit-switched networks: In circuit switched networks, network performance is synonymous with the grade of service. The number of rejected calls is a measure of how well the network is performing under heavy traffic loads. Other types of performance measures can include the level of noise and echo.

- ATM: In an Asynchronous Transfer Mode (ATM) network, performance can be measured by line rate, quality of service (QoS), data throughput, connect time, stability, technology, modulation technique and modem enhancements.

There are many ways to measure the performance of a network, as each network is different in nature and design. Performance can also be modeled instead of measured. For example, state transition diagrams are often used to model queuing performance in a circuit-switched network. The network planner uses these diagrams to analyze how the network performs in each state, ensuring that the network is optimally designed.

Network congestion

Network congestion occurs when a link or node is carrying so much data that its quality of service deteriorates. Typical effects include queueing delay, packet loss or the blocking of new connections. A consequence of these latter two is that incremental increases in offered load lead either only to small increase in network throughput, or to an actual reduction in network throughput.

Network protocols that use aggressive retransmissions to compensate for packet loss tend to keep systems in a state of network congestion—even after the initial load is reduced to a level that would not normally induce network congestion. Thus, networks using these protocols can exhibit two stable states under the same level of load. The stable state with low throughput is known as *congestive collapse*.

Modern networks use congestion control and congestion avoidance techniques to try to avoid congestion collapse. These include: exponential backoff in protocols such as 802.11's CSMA/CA and the original Ethernet, window reduction in TCP, and fair queueing in devices such as routers. Another method to avoid the negative effects of network congestion is implementing priority schemes, so that some packets are transmitted with higher priority than others. Priority schemes do not solve network congestion by themselves, but they help to alleviate the effects of congestion for some services. An example of this is 802.1p. A third method to avoid network congestion is the explicit allocation of network resources to specific flows. One example of this is the use of Contention-Free Transmission Opportunities (CFTXOPs) in the ITU-T G.hn standard, which provides high-speed (up to 1 Gbit/s) Local area networking over existing home wires (power lines, phone lines and coaxial cables).

For the Internet RFC 2914 addresses the subject of congestion control in detail.

Network resilience

Network resilience is "the ability to provide and maintain an acceptable level of service in the face of faults and challenges to normal operation."

Security

Network security

Network security consists of provisions and policies adopted by the network administrator to prevent and monitor unauthorized access, misuse, modification, or denial of the computer network and its network-accessible resources. Network security is the authorization of access to data in a network, which is controlled by the network administrator. Users are assigned an ID and password that allows them access to information and programs within their authority. Network security is used on a variety of computer networks, both public and private, to secure daily transactions and communications among businesses, government agencies and individuals.

Network surveillance

Network surveillance is the monitoring of data being transferred over computer networks such as the Internet. The monitoring is often done surreptitiously and may be done by or at the behest of governments, by corporations, criminal organizations, or individuals. It may or may not be legal and may or may not require authorization from a court or other independent agency.

Computer and network surveillance programs are widespread today, and almost all Internet traffic is or could potentially be monitored for clues to illegal activity.

Surveillance is very useful to governments and law enforcement to maintain social control, recognize and monitor threats, and prevent/investigate criminal activity. With the advent of programs such as the Total Information Awareness program, technologies such as high speed surveillance computers and biometrics software, and laws such as the Communications Assistance For Law Enforcement Act, governments now possess an unprecedented ability to monitor the activities of citizens.

However, many civil rights and privacy groups—such as Reporters Without Borders, the Electronic Frontier Foundation, and the American Civil Liberties Union—have expressed concern that increasing surveillance of citizens may lead to a mass surveillance society, with limited political and personal freedoms. Fears such as this have led to numerous lawsuits such as *Hepting v. AT&T*. The hacktivist group Anonymous has hacked into government websites in protest of what it considers “draconian surveillance.”

End to end encryption

End-to-end encryption (E2EE) is a digital communications paradigm of uninterrupted protection of data traveling between two communicating parties. It involves the originating party encrypting data so only the intended recipient can decrypt it, with no dependency on third parties. End-to-end encryption prevents intermediaries, such as Internet providers or application service providers, from discovering or tampering with communications. End-to-end encryption generally protects both confidentiality and integrity.

Examples of end-to-end encryption include PGP for email, OTR for instant messaging, ZRTP for telephony, and TETRA for radio.

Typical server-based communications systems do not include end-to-end encryption. These systems can only guarantee protection of communications between clients and servers, not between the communicating parties themselves. Examples of non-E2EE systems are Google Talk, Yahoo Messenger,Facebook, and Dropbox. Some such systems, for example LavaBit and SecretInk, have even described themselves as offering “end-to-end” encryption when they do not. Some systems that normally offer end-to-end encryption have turned out to contain a back door that subverts negotiation of theencryption key between the communicating parties, for example Skype.

The end-to-end encryption paradigm does not directly address risks at the communications endpoints themselves, such as the technical exploitation of clients, poor quality random number generators, or key escrow. E2EE also does not address traffic analysis, which relates to things such as the identities of the end points and the times and quantities of messages that are sent.

Views of networks

Users and network administrators typically have different views of their networks. Users can share printers and some servers from a workgroup, which usually means they are in the same geographic location and are on the same LAN, whereas a Network Administrator is responsible to keep that network up and running. A community of interest has less of a connection of being in a local area, and should be thought of as a set of arbitrarily located users who share a set of servers, and possibly also communicate via peer-to-peer technologies.

Network administrators can see networks from both physical and logical perspectives. The physical perspective involves geographic locations, physical cabling, and the network elements (e.g., routers, bridges and application layer gateways) that interconnect via the transmission media. Logical networks, called, in the TCP/IP architecture, subnets, map onto one or more transmission media. For example, a common practice in a campus of buildings is to make a set of LAN cables in each building appear to be a common subnet, using virtual LAN (VLAN) technology.

Both users and administrators are aware, to varying extents, of the trust and scope characteristics of a network. Again using TCP/IP architectural terminology, an intranet is a community of interest under private administration usually by an enterprise, and is only accessible by authorized users (e.g. employees). Intranets do not have to be connected to the Internet, but generally have a limited connection. An extranet is an extension of an intranet that allows secure communications to users outside of the intranet (e.g. business partners, customers).

Unofficially, the Internet is the set of users, enterprises, and content providers that are interconnected by Internet Service Providers (ISP). From an engineering viewpoint, the Internet is the set of subnets, and aggregates of subnets, which share the registered IP address space and exchange information about the reachability of those IP addresses using the Border Gateway Protocol. Typically, the human-readable names of servers are translated to IP addresses, transparently to users, via the directory function of the Domain Name System (DNS).

Over the Internet, there can be business-to-business (B2B), business-to-consumer (B2C) and consumer-to-consumer (C2C) communications. When money or sensitive information is exchanged, the communications are apt to be protected by some form of communications security mechanism. Intranets and extranets can be securely superimposed onto the Internet, without any access by general Internet users and administrators, using secure Virtual Private Network (VPN) technology.

Licensing & Attributions

CC licensed content, Shared previously

- Computer Network. **Provided by:** Wikipedia. **Located at:** https://en.wikipedia.org/wiki/Computer_network. **License:** *CC BY-SA: Attribution-ShareAlike*
- Untitled . **Authored by:** Jared Tarbell. **Located at:** https://www.flickr.com/photos/generated/291537716/. **License:** *CC BY: Attribution*
- Access Denied. **Authored by:** Chris Smart. **Located at:** https://www.flickr.com/photos/sigma/10021615/. **License:** *CC BY-NC: Attribution-NonCommercial*

READING: TELECOMMUNICATIONS NETWORK

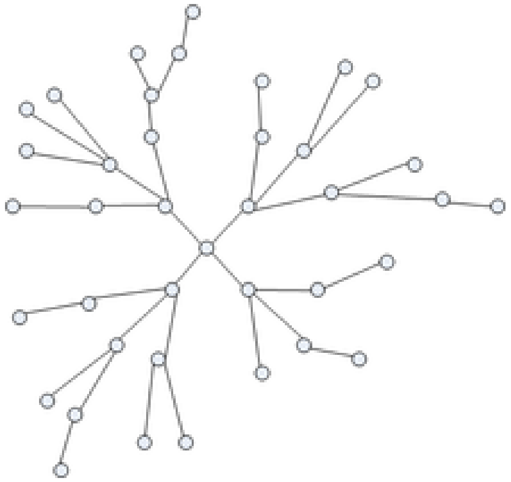
A **telecommunications network** is a collection of terminal nodes, links and any intermediate nodes which are connected so as to enable telecommunication between the terminals.

The transmission links connect the nodes together. The nodes use circuit switching, message switching or packet switching to pass the signal through the correct links and nodes to reach the correct destination terminal.

Each terminal in the network usually has a unique address so messages or connections can be routed to the correct recipients. The collection of addresses in the network is called the address space.

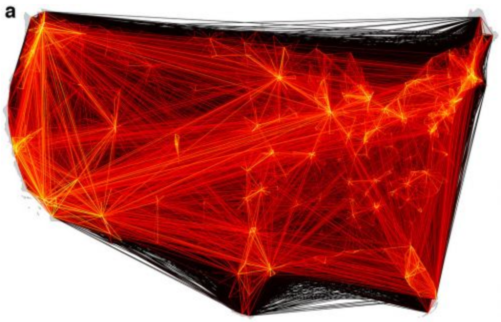
Examples of telecommunications networks are:

- computer networks
- the Internet
- the telephone network
- the global Telex network
- the aeronautical ACARS network



Example of how nodes may be interconnected with links to form a telecommunications network. This example is tree-like but many networks have loops.

Benefits of Telecommunications and Networking



Telecommunications can greatly increase and expand resources to all types of people. For example, businesses need a greater telecommunications network if they plan to expand their company. With Internet, computer, and telephone networks, businesses can allocate their resources efficiently. These core types of networks will be discussed below:

Computer Network: A computer network consists of computers and devices connected to one another. Information can be transferred from one device to the next. For example, an office filled with computers can share files together on each separate device. Computer networks can range from a local network area to a wide area network. The difference between the types of networks is the size. These types of computer networks work at certain speeds, also known as broadband. The Internet network can connect computer worldwide.

Internet Network: Access to the network allows users to use many resources. Over time the Internet network will replace books. This will enable users to discover information almost instantly and apply concepts to different situations. The Internet can be used for recreational, governmental, educational, and other purposes. Businesses in particular use the Internet network for research or to service customers and clients.

Telephone Network: The telephone network connects people to one another. This network can be used in a variety of ways. Many businesses use the telephone network to route calls and/or service their customers. Some businesses use a telephone network on a greater scale through a private branch exchange. It is a system where a specific business focuses on routing and servicing calls for another business. Majority of the time, the telephone network is used around the world for recreational purposes.

Network structure

In general, every telecommunications network conceptually consists of three parts, or planes (so called because they can be thought of as being, and often are, separate overlay networks):

- The control plane carries control information (also known as signalling).
- The data plane or user plane or bearer plane carries the network's users traffic.
- The management plane carries the operations and administration traffic required for network management.

Example: the TCP/IP data network

The data network is used extensively throughout the world to connect individuals and organizations. Data networks can be connected to allow users seamless access to resources that are hosted outside of the particular provider they are connected to. The Internet is the best example of many data networks from different organizations all operating under a single address space.

Terminals attached to TCP/IP networks are addressed using IP addresses. There are different types of IP address, but the most common is IP Version 4. Each unique address consists of 4 integers between 0 and 255, usually separated by dots when written down, e.g. 82.131.34.56.

TCP/IP are the fundamental protocols that provide the control and routing of messages across the data network. There are many different network structures that TCP/IP can be used across to efficiently route messages, for example:

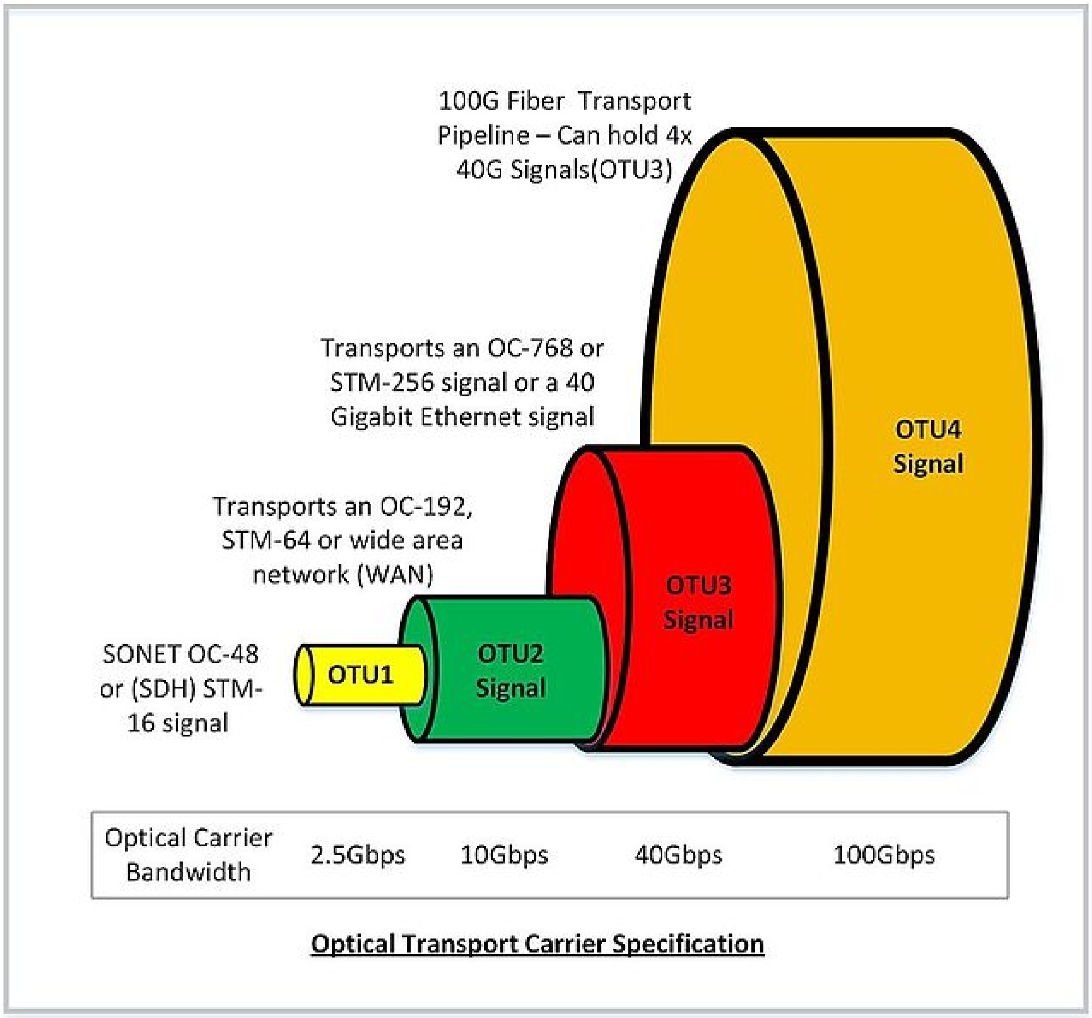
- wide area networks (WAN)
- metropolitan area networks (MAN)
- local area networks (LAN)
- Internet area networks (IAN)
- campus area networks (CAN)

- virtual private networks (VPN)

There are three features that differentiate MANs from LANs or WANs:

1. The area of the network size is between LANs and WANs. The MAN will have a physical area between 5 and 50 km in diameter.
2. MANs do not generally belong to a single organization. The equipment that interconnects the network, the links, and the MAN itself are often owned by an association or a network provider that provides or leases the service to others.
3. A MAN is a means for sharing resources at high speeds within the network. It often provide connections to WAN networks for access to resources outside the scope of the MAN.

Optical Transport Network (OTN)


100G Fiber Transport Pipeline – Can hold 4x 40G Signals(OTU3)
Transports an OC-768 or STM-256 signal or a 40 Gigabit Ethernet signal
Transports an OC-192, STM-64 or wide area network (WAN)
SONET OC-48 or (SDH) STM-16 signal
OTU1
OTU2 Signal
OTU3 Signal
OTU4 Signal
Optical Carrier Bandwidth
2.5Gbps
10Gbps
40Gbps
100Gbps
Optical Transport Carrier Specification

Optical Transport Network (OTN) Specs

Optical Transport Network (OTN) is a large complex network of server hubs at different locations on ground, connected by Optical fiber cable or optical network carrier, to transport data across different nodes. The server hubs are also known as head-ends, nodes or simply, sites. OTNs are the backbone of Internet Service Providers and are often daisy chained and cross connected to provide network redundancy. Such a setup facilitates uninterrupted services and fail-over capabilities during maintenance windows, equipment failure or in case of accidents.

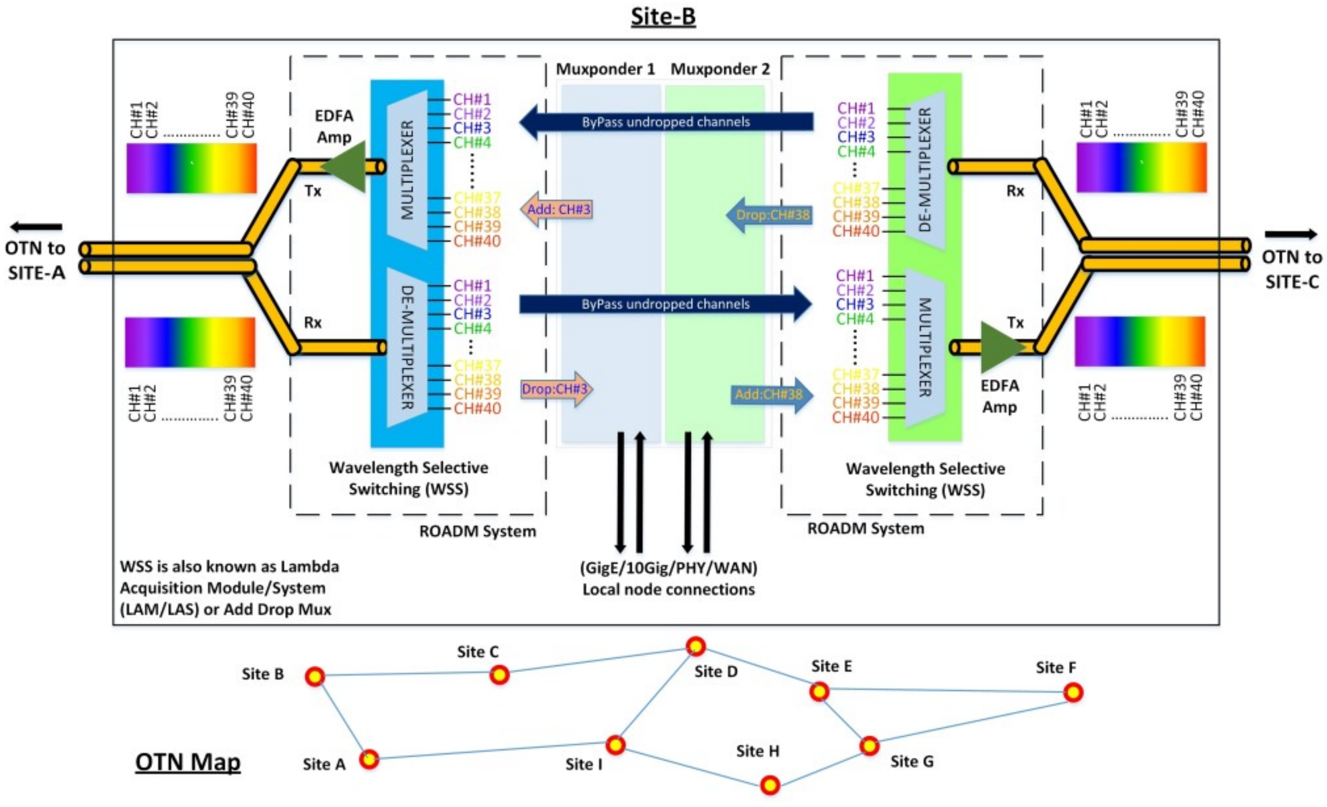
The devices used to transport data are known as network transport equipment. Some of the widely used equipment are manufactured by

- Alcatel Lucent – AL7510, AL7750
- Nortel Networks Corp. (acquired by Ciena Corp.) – Optera Metro series – OM4500, OM6500
- Fujitsu Ltd. – FlashWave series FW4500, FW7500, FW9500

The capacity of a network is mainly dependent on the type of signaling scheme employed on transmitting and receiving end. In the earlier days, a single wavelength light beam was used to transmit data, which limited the bandwidth to the maximum operating frequency of the transmitting and receiving end equipment. With the application of wavelength division multiplexing (WDM), the bandwidth of OTN has risen up to 100Gbit/s (OTU4 Signal), by emitting light beams of different wavelengths. Lately, AT&T, Verizon, and Rogers Communication have been able to employ these 100G “pipes” in their metro network. Large field areas are mostly serviced by 40G pipes (OC192/STM-64).

A 40G pipe can carry 40 different channels as a result of Dense Wave Division Multiplexing (DWDM) transmission. Each node in the network is able to access different channels, but is mostly tuned to a few channels. The data from a channel can be dropped to the node or new data can be added to the node using Re-configurable Optic Add Drop Mux (ROADM) that uses Wavelength Selective Switching (WSS) to extract and infuse a configured frequency. This eliminates the need to convert all the channels to electric signals, extract the required channels, and convert the rest back to optical into the OTN. Thus ROADM systems are fast, less expensive and can be configured to access any channel in the OTN pipe.

The extracted channels at a site are connected to local devices through muxponder or tranponder cards that can split or combine 40G channels to 4x 10G channels or 8x 2.5G channels.



Re-configurable Optical Add Drop Multiplexer (ROADM). Click for a larger view of the image.

Licensing & Attributions

CC licensed content, Shared previously

- Telecommunications network. **Provided by:** Wikipedia. **Located at:** https://en.wikipedia.org/wiki/Telecommunications_network. **License:** *CC BY-SA: Attribution-ShareAlike*
- Human mobility network derived from bank note fluxes via Christian Thiemann et al. **Authored by:** Duncan Hill. **Located at:** https://www.flickr.com/photos/dullhunk/5221572088/. **License:** *CC BY: Attribution*

READING: SECURITY AND SOCIAL ISSUES

Security Within the Organization

Security risks within an organization include the processing of fraudulent transactions, unauthorized access to data and program files, and the physical theft or damage of equipment.

Fraud

Computer fraud is increasing at an alarming rate. Fraud can be defined as the manipulation of the records of an organization to conceal an illegal act (normally the theft of funds or other assets). Computers can make it easy for employees in particular to defraud the organization, in particular when the level of security and internal control is lax. In manual systems, a common control to limit fraud is to involve two or more people in a process, each one effectively controlling the activities of the others. We call this control process separation of duties. For example in a payroll system one might give an individual the authority to approve increases, another the task of updating the computer and the third the responsibility to distribute funds to employees. Without collusion between them, it would be difficult for any one of these individuals to steal funds from the payroll system and hide his tracks. Unfortunately, in many computer systems, too many separate functions have been computerized and often there is a single clerk responsible for running the entire payroll process. In these situations, anyone who has access to the application can take the opportunity to commit fraud. The most common fraud tactics are:

- Entering fictitious transactions. Most frauds are committed by employees using the system in the normal way to enter fictitious transactions. No special technical knowledge is required and the employee relies on the fact that management supervision of the process in weak.
- Modification of computer files. Normally requires a little more technical expertise as this would involve, for example, the increase or reduction of amounts held on the master file, which cannot be changed within the application without an appropriate transaction (such as a payment).
- Unauthorized changes to programs. This type of fraud is usually limited to staff with programming expertise. A common example is the skimming or salami technique. In a payroll system this would entail deducting a small amount from each individual salary cheque and adding the total to a select individual's payment. The secret behind this technique is that employees are unlikely to notice a change in their salary (PAYE and other deductions often cause regular variation in the total) and the total payroll will balance (the total amount being paid is the same.)

How does an organization limit fraud? Experts suggest a three-pronged attack. Firstly the organization must stress the need for honesty and ethical behavior in all business activities. Managers must lead by example, new employees must be screened and staff training must support this theme. The second concern is the level of opportunity in the organization to commit fraud. There must be strong internal controls, separation of duties, restricted access to sensitive applications and constant management supervision. Audit trails are used to record the origin of every transaction, and sequential numbering ensures that records cannot be deleted or reports destroyed. Finally, where a case of fraud is discovered, action must be taken against the offender. Many organizations prefer not to prosecute employees suspected of fraudulent behavior because of the negative publicity they will receive in the press. This in itself encourages criminals to repeat the activity in their new working environment knowing the likelihood of punishment is remote.

Unauthorized Data Access

Password protection is the most common method of protecting corporate data. Nevertheless, fraudulent transactions are often carried out by unauthorized users who manage to gain access to the corporate network by using the login details of another user. One way of achieving this is through a *terminal spoof*—a simple yet effective approach to finding other user's passwords. A terminal spoof is a program that runs on a machine and looks like the normal login screen. Once a user has given his or her user-ID and password, the terminal spoof will record both on the local disk or server, give what looks like an authentic error message (such as invalid password—please re-enter) and then passes control to the real login program. The criminal will pick up the passwords later to gain access to the system masquerading as the unfortunate victim.

Other criminals simply make use of an unattended computer that has been left on by a user who has logged in to the network and then left the office. Time-out or screen-saver programs with password protection provide a simple barrier to this approach In addition, locked doors are a traditional means of excluding undesirable visitors.

Other dangers of which managers should be aware include the *Trojan horse*, in which code is added to a program, which will activate under certain conditions. For example, a computer consultant in Johannesburg had a client in Durban. He placed a Trojan horse in the payroll program so that it would malfunction while processing the June payroll. They would fly him down, all expenses paid, to fix the problem and stay for the Durban July horse race. Once this had happened for the third time, another consultant was used who uncovered the offending code.

Another risk is the *Back-door technique*. When programmers are building systems, they may try to bypass all the access security procedures to speed up the development time. In some cases, these "back doors" have not been removed and the programmer can gain illegal entry into the production system.

Sabotage and Theft

When computers were the size of small houses and hidden in secure computer installations, then theft of computer hardware was rare. Today, PCs are on most desks and in many cases they have to be physically bolted to the table to prevent their disappearance. One famous case of theft involved a laptop computer stolen from the back seat of a car in the USA in early 1991. On the hard disk was the master plan for Desert Storm, the details of how the United States and her allies would attack Iraq.

Mobile computing devices are especially vulnerable to theft, and limiting of physical access to equipment is the most effective first line of defense. Restrictions to entry can be based on electronic locks, activated by means of magnetic disks or swipe cards, or on advanced biometric devices that identify the individual based on characteristics such as fingerprints or the pattern of the retina. (In each case, the security mechanism would obviously be linked to a database containing details of authorized users.)

Another form of theft relates to the copying of programs and data resources in an organization. Obtaining customer lists together with the details of the amount and type of business can obviously assist companies to encourage customers away from their competition. Theft of software is a major problem in the PC world where users often make illegal copies of the programs rather than purchase the package themselves – this practice is known as software piracy. This type of theft is more difficult to identify, since the original product has not physically disappeared as with the theft of computer hardware. Where software piracy is discovered, the owner of the computer on which the software resides (often the employer) is held to be legally responsible for the presence of pirated software.

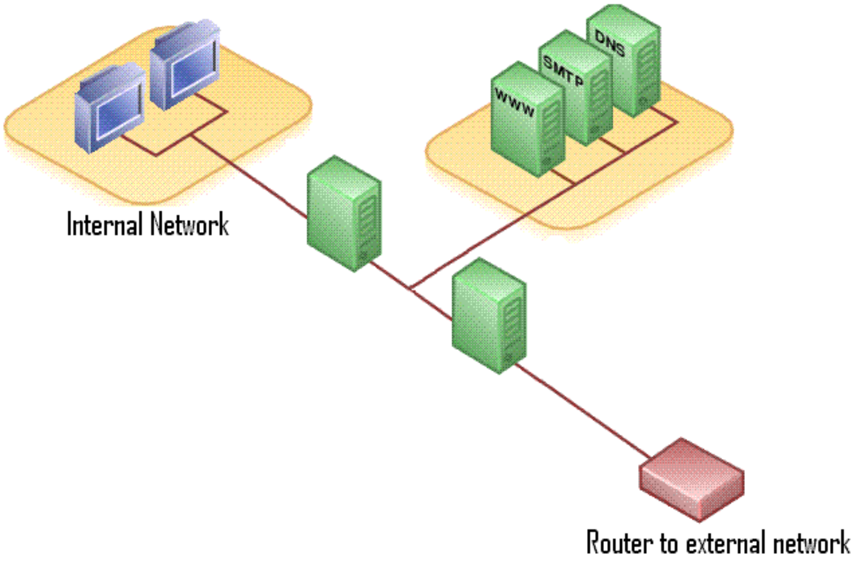
The last category of computer theft covers the illegal use of computer time. In the past computer operators were often caught processing work for third parties or users were doing their own work at the office. Computer hackers spend their time searching for networks to which they can gain access. Having breached the security controls, they often browse around the databases in the installation but may not do any damage. In these instances, the only crime they can be charged with is the theft of computer time.

Security Beyond the Organization

Hackers and Firewalls

Hackers are users from outside the organization, who penetrate a computer system usually through its communication lines. Although some hackers are content merely to demonstrate that they have bypassed network security, there is a high risk of malicious damage to data, stealing of sensitive information (such as customers' credit card numbers) or entry of fraudulent transactions by the hacker. Hackers may also instigate a denial-of-service attack, in which a targeted web site is inundated with requests for information initiated by the hacker, rendering it inaccessible for genuine business customers.

A firewall is an additional system that enforces access control policy between two networks, especially between a corporate network and the Internet. The firewall monitors all external communications, checks user authorization and maintains a log of all attempts to access the network. They can also be used to check for the presence of viruses, for the downloading of unauthorized software, and to guard against denial-of-service attacks.



Click on the image for a larger view.

Data which is in the process of being communicated is also vulnerable to eavesdropping. Encryption, which scrambles data into an unreadable form, can be used to improve data privacy and prevent any unauthorized changes to the message, as well as protecting the confidentiality of data within the organization.

Viruses

A computer virus is a program that invades a computer system, normally by residing in corrupt files. The virus has the ability to replicate itself and so spread to other files and computer systems. Some viruses are benign and merely advertise their presence, but others corrupt the files they infect and even destroy entire databases.

There are three main types of viruses. The original viruses were mainly systems viruses. They resided in the boot sector of a disk (the first place the computer looks when loading the operating system) or in the operating system utilities. These viruses were usually easy to find and clean. One problem was that they loaded into memory as soon as an infected machine was switched on and could often hide themselves from the anti-viral software.

The next generation of viruses attached themselves to executable files. When an infected program is run, the virus resides in memory and infects all new programs run until the system is switched off. These viruses are difficult to counter, especially on a network as common files are often infected. Even when the file server is cleared of infected programs, users have copies on their personal hard drives, or a infected copy of the program in memory when the virus check is performed on the server.

One area most users with which felt safe was the accessing of non-executable files such as word processing documents. Unfortunately there are now a number of macro viruses that can attach themselves to documents and spreadsheets. Even receiving a simple letter as an e-mail attachment can infect your machine. Some of the more well known viruses include:

- Michelangelo. This virus infected many machines in the early months of 1992. The virus was primed to activate on Michelango's birthday (6th March) and had the capability of destroying all files on the hard disk of infected PC's. News of the virus made headlines and many businessmen and home users rushed out to purchase programs to check and clean viruses from their installations. On the day, some infections did occur but the main winners were the vendors of virus detection software.
- Stoned. A very common virus in the late 1980s it came in many forms, some harmless while others corrupted files by attacking the directories and allocation tables. The main theme of the virus was to legalize the smoking of cannabis and normally the message "Your PC is now Stoned" would appear on the screen.
- Jerusalem. Deletes all programs that are run on Friday 13th.
- Concept. A macro virus, attached to word processing documents.
- EXEBUG. A nasty little bug that may corrupt your hard drive. This is a systems virus and can infect the CMOS of your PC. Very difficult to find and eradicate as it uses "stealth" technology to hide from virus checkers.

With increasing globalization and interorganizational communications, viruses are able to spread faster and further than ever before. Recent examples include Melissa, ILoveYou, and the Nimda virus, which make use of multiple methods of transmission. The real mystery about viruses is why they exist at all. Some experts suggest that computer hackers are motivated by the challenge of "beating the system" by writing programs that can bypass virus protection systems and take control of each individual machine. A more likely motive is to induce computer users to purchase legal copies of software. The only real winners in the new world of computer viruses are the companies selling computer software.

In the early 1980's the illegal copying of PC software, or software piracy was rife as there was no real business advantage to purchasing the software (except a copy of the official reference manual). Today it is estimated that about 50% of the PC software used in the USA has been illegally copied and the situation in South Africa is likely to be worse. One of the major motivations for buying software rather than copying from a friend is that the shrink-wrapped product is guaranteed to be virus free.

The best line of defense against viruses is the regular use of up-to-date anti-virus software, which will scan files for viruses and remove them if found.

Licensing & Attributions

CC licensed content, Shared previously

- Discovering Information Systems Sections 10.1-10.3. **Authored by:** Jean-Paul Van Belle, Mike Eccles, and Jane Nash. **Located at:** https://archive.org/details/ost-computer-science-discoveringinformationsystems. **License:** *CC BY-NC-ND: Attribution-NonCommercial-NoDerivatives*
- DMZ Network Diagram Two Firewalls. **Authored by:** Jasonlfunk. **Located at:** https://en.wikipedia.org/wiki/File:DMZ_network_diagram_2_firewalls.png. **License:** *CC0: No Rights Reserved*
- Vulnerable. **Authored by:** Jannis Blume. **Located at:** https://www.flickr.com/photos/iblume/6230865185/. **License:** *CC BY-SA: Attribution-ShareAlike*
- I, the Hacker. **Authored by:** Charis Tsevis. **Located at:** https://www.flickr.com/photos/tsevis/93150384/. **License:** *CC BY-NC-ND: Attribution-NonCommercial-NoDerivatives*

READING: COMPUTER SECURITY

https://learn.saylor.org/course/view.php?id=94&sectionid=974

Licensing & Attributions

Public domain content

- License: *Public Domain: No Known Copyright*

READING: INFORMATION SECURITY - SECURITY INTRODUCTION

https://www.oercommons.org/courses/information-security-06-01-introduction

Licensing & Attributions

Public domain content

- License: *Public Domain: No Known Copyright*

READING: INFORMATION SECURITY - PROTECTING YOUR DATA

https://www.oercommons.org/courses/information-security-06-04-protecting-your-data

Licensing & Attributions

Public domain content

- License: *Public Domain: No Known Copyright*

READING: INFORMATION SECURITY - INTERNET SECURITY

https://www.oercommons.org/courses/information-security-06-06-internet-security

Licensing & Attributions

Public domain content

- License: *Public Domain: No Known Copyright*

READING: INFORMATION SECURITY - PHYSICAL SECURITY

https://www.oercommons.org/courses/information-security-06-02-physical-security

READING: INFORMATION SECURITY - SOCIAL ENGINEERING

https://www.oercommons.org/courses/information-security-06-08-social-engineering

Licensing & Attributions

Public domain content

- License: *Public Domain: No Known Copyright*

READING: INFORMATION SECURITY - MALWARE

https://www.oercommons.org/courses/information-security-06-07-malware

Licensing & Attributions

Public domain content

- License: *Public Domain: No Known Copyright*

READING: INFORMATION SECURITY - NETWORK SECURITY

https://www.oercommons.org/course/information-security-06-05-network-security

Licensing & Attributions

Public domain content

- License: *Public Domain: No Known Copyright*

READING: INTERNET SECURITY

*This reading was included previously in the course but is repeated here as a refresher, as the content is relevant to our discussion of networks and security.

Internet security is a tree branch of computer security specifically related to the Internet, often involving browser security but also network security on a more general level as it applies to other applications or operating systems on a whole. Its objective is to establish rules and measures to use against attacks over the Internet. The Internet represents an insecure channel for exchanging information leading to a high risk of intrusion or fraud, such as phishing. Different methods have been used to protect the transfer of data, including encryption.

Types of security

Network layer security

TCP/IP which stands for Transmission Control Protocol (TCP) and Internet Protocol (IP) aka Internet protocol suite can be made secure with the help of cryptographic methods and protocols. These protocols include Secure Sockets Layer (SSL), succeeded by Transport Layer Security (TLS) for web traffic, Pretty Good Privacy (PGP) for email, and IPsec for the network layer security.

Internet Protocol Security (IPsec)

This protocol is designed to protect communication in a secure manner using TCP/IP aka Internet protocol suite. It is a set of security extensions developed by the Internet Task force IETF, and it provides security and authentication at the IP layer by transforming data using encryption. Two main types of transformation that form the basis of IPsec: the Authentication Header (AH) and ESP. These two protocols provide data integrity, data origin authentication, and anti-replay service. These protocols can be used alone or in combination to provide the desired set of security services for the Internet Protocol (IP) layer.

The basic components of the IPsec security architecture are described in terms of the following functionalities:

- Security protocols for AH and ESP
- Security association for policy management and traffic processing
- Manual and automatic key management for the Internet key exchange (IKE)
- Algorithms for authentication and encryption

The set of security services provided at the IP layer includes access control, data origin integrity, protection against replays, and confidentiality. The algorithm allows these sets to work independently without affecting other parts of the implementation. The IPsec implementation is operated in a host or security gateway environment giving protection to IP traffic.

Security token

Some online sites offer customers the ability to use a six-digit code which randomly changes every 30–60 seconds on a security token. The keys on the security token have built in mathematical computations and manipulate numbers based on the current time built into the device. This means that every thirty seconds there is only a certain array of numbers possible which would be correct to validate access to the online account. The website that the user is logging into would be made aware of that devices' serial number and would know the computation and correct time built into the device to verify that the number given is indeed one of the handful of six-digit numbers that works in that given 30-60 second cycle. After 30–60 seconds the device will present a new random six-digit number which can log into the website.

Electronic mail security (E-mail)

Background

Email messages are composed, delivered, and stored in a multiple step process, which starts with the message's composition. When the user finishes composing the message and sends it, the message is transformed into a standard format: an RFC 2822 formatted message. Afterwards, the message can be transmitted. Using a network connection, the mail client, referred to as a mail user agent (MUA), connects to a mail transfer agent (MTA) operating on the mail server. The mail client then provides the sender's identity to the server. Next, using the mail server commands, the client sends the recipient list to the mail server. The client then supplies the message. Once the mail server receives and processes the message, several events occur: recipient server identification, connection establishment, and message transmission. Using Domain Name System (DNS) services, the sender's mail server determines the mail server(s) for the recipient(s). Then, the server opens up a connection(s) to the recipient mail server(s) and sends the message employing a process similar to that used by the originating client, delivering the message to the recipient(s).

Pretty Good Privacy (PGP)

Pretty Good Privacy provides confidentiality by encrypting messages to be transmitted or data files to be stored using an encryption algorithm such Triple DES or CAST-128. Email messages can be protected by using cryptography in various ways, such as the following:

- Signing an email message to ensure its integrity and confirm the identity of its sender.
- Encrypting the body of an email message to ensure its confidentiality.
- Encrypting the communications between mail servers to protect the confidentiality of both message body and message header.

The first two methods, message signing and message body encryption, are often used together; however, encrypting the transmissions between mail servers is typically used only when two organizations want to protect emails regularly sent between each other. For example, the organizations could establish a virtual private network (VPN) to encrypt the communications between their mail servers over the Internet. Unlike methods that can only encrypt a message body, a VPN can encrypt entire messages, including email header information such as senders, recipients, and subjects. In some cases, organizations may need to protect header information. However, a VPN solution alone cannot provide a message signing mechanism, nor can it provide protection for email messages along the entire route from sender to recipient.

Multipurpose Internet Mail Extensions (MIME)

MIME transforms non-ASCII data at the sender's site to Network Virtual Terminal (NVT) ASCII data and delivers it to client's Simple Mail Transfer Protocol (SMTP) to be sent through the Internet. The server SMTP at the receiver's side receives the NVT ASCII data and delivers it to MIME to be transformed back to the original non-ASCII data.

Message Authentication Code

A Message authentication code (MAC) is a cryptography method that uses a secret key to encrypt a message. This method outputs a MAC value that can be decrypted by the receiver, using the same secret key used by the sender. The Message Authentication Code protects both a message's data integrity as well as its authenticity.

Firewalls

A computer firewall controls access between networks. It generally consists of gateways and filters which vary from one firewall to another. Firewalls also screen network traffic and are able to block traffic that is dangerous. Firewalls act as the intermediate server between SMTP and Hypertext Transfer Protocol (HTTP) connections.

Role of firewalls in web security

Firewalls impose restrictions on incoming and outgoing Network packets to and from private networks. Incoming or outgoing traffic must pass through the firewall; only authorized traffic is allowed to pass through it. Firewalls create checkpoints between an internal private network and the public Internet, also known as *choke points*(borrowed from the identical military term of a combat limiting geographical feature). Firewalls can create choke points based on IP source and TCP port number. They can also serve as the platform for IPsec. Using tunnel mode capability, firewall can be used to implement VPNs. Firewalls can also limit network exposure by hiding the internal network system and information from the public Internet.

Types of firewall

Packet filter

A packet filter is a first generation firewall that processes network traffic on a packet-by-packet basis. Its main job is to filter traffic from a remote IP host, so a router is needed to connect the internal network to the Internet. The router is known as a screening router, which screens packets leaving and entering the network.

Stateful packet inspection

In a stateful firewall the circuit-level gateway is a proxy server that operates at the network level of an Open Systems Interconnection (OSI) model and statically defines what traffic will be allowed. Circuit proxies will forward Network packets (formatted unit of data) containing a given port number, if the port is permitted by the algorithm. The main advantage of a proxy server is its ability to provide Network Address Translation (NAT), which can hide the user's IP address from the Internet, effectively protecting all internal information from the Internet.

Application-level gateway

An application-level firewall is a third generation firewall where a proxy server operates at the very top of the OSI model, the IP suite application level. A network packet is forwarded only if a connection is established using a known protocol. Application-level gateways are notable for analyzing entire messages rather than individual packets of data when the data are being sent or received.

Malicious software

A computer user can be tricked or forced into downloading software onto a computer that is of malicious intent. Such software comes in many forms, such as viruses, Trojan horses, spyware, and worms.

- Malware, short for malicious software, is any software used to disrupt computer operation, gather sensitive information, or gain access to private computer systems. Malware is defined by its malicious intent, acting against the requirements of the computer user, and does not include software that causes unintentional harm due to some deficiency. The term badware is sometimes used, and applied to both true (malicious) malware and unintentionally harmful software.
- A botnet is a network of zombie computers that have been taken over by a robot or bot that performs large-scale malicious acts for the creator of the botnet.
- Computer Viruses are programs that can replicate their structures or effects by infecting other files or structures on a computer. The common use of a virus is to take over a computer to steal data.
- Computer worms are programs that can replicate themselves throughout a computer network, performing malicious tasks throughout.
- Ransomware is a type of malware which restricts access to the computer system that it infects, and demands a ransom paid to the creator(s) of the malware in order for the restriction to be removed.
- Scareware is scam software with malicious payloads, usually of limited or no benefit, that are sold to consumers via certain unethical marketing practices. The selling approach uses social engineering to cause shock, anxiety, or the perception of a threat, generally directed at an unsuspecting user.
- Spyware refers to programs that surreptitiously monitor activity on a computer system and report that information to others without the user's consent.
- A Trojan horse, commonly known as a *Trojan*, is a general term for malicious software that pretends to be harmless, so that a user willingly allows it to be downloaded onto the computer.

Denial-of-service attack

A denial-of-service attack (DoS attack) or distributed denial-of-service attack (DDoS attack) is an attempt to make a computer resource unavailable to its intended users. Although the means to carry out, motives for, and targets of a DoS attack may vary, it generally consists of the concerted efforts to prevent an Internet site or service from functioning efficiently or at all, temporarily or indefinitely. According to businesses who participated in an international business security survey, 25% of respondents experienced a DoS attack in 2007 and 16.8% experienced one in 2010.

Phishing

Phishing is another common threat to the Internet. “SA, the Security Division of EMC, today announced the findings of its January 2013 Fraud Report, estimating the global losses from Phishing at $1.5 Billion in 2012.”. Filter evasion, website forgery, phone phishing, Covert Redirect are some well known phishing techniques.

Hackers use a variety of tools to conduct phishing attacks. They create forged websites that pretend to be other websites in order for users to leave their personal information. These hackers usually host these sites on legitimate hosting services using stolen credit cards while the last trend is to use a mailing system and finding a mailing list of people which they can try and fraud.

Browser choice

Web browser statistics tend to affect the amount a Web browser is exploited. For example, Internet Explorer 6, which used to own a majority of the Web browser market share, is considered extremely insecure because vulnerabilities were exploited due to its former popularity. Since browser choice is more evenly distributed (Internet Explorer at 28.5%, Firefox at 18.4%, Google Chrome at 40.8%, and so on) and vulnerabilities are exploited in many different browsers.

Application vulnerabilities

Applications used to access Internet resources may contain security vulnerabilities such as memory safety bugs or flawed authentication checks. The most severe of these bugs can give network attackers full control over the computer. Most security applications and suites are incapable of adequate defense against these kinds of attacks.

Internet security products

Antivirus

Antivirus and Internet security programs can protect a programmable device from malware by detecting and eliminating viruses; Antivirus software was mainly shareware in the early years of the Internet, but there are now several free security applications on the Internet to choose from for all platforms.

Security suites

So called “security suites” were first offered for sale in 2003 (McAfee) and contain a suite of firewalls, anti-virus, anti-spyware and more.[15] They may now offer theft protection, portable storage device safety check, private Internet browsing, cloud anti-spam, a file shredder or make security-related decisions (answering popup windows) and several were free of charge as of at least 2012.

Licensing & Attributions

CC licensed content, Shared previously

- Network Security. **Provided by:** Wikipedia. **Located at:** https://en.wikipedia.org/wiki/Network_security. **License:** *CC BY-SA: Attribution-ShareAlike*

TUTORIALS: COMPUTER SECURITY

View the following Atomic Learning Tutorials at Atomic Learning (TOOLS menu–ATOMIC LEARNING – search on COMPUTER SECURITY – select PC Security & Maintenance)

B. MALWARE and OTHER THREATS

- Defining Malware
- Preventing Malware

C. PC PROTECTION

- Protecting your Windows 7 PC with Microsoft PC Security Essentials
- Protecting your Windows 8 System

D. Firewalls – ALL

G. Home Network Security – ALL

To access these trainings, follow the instructions from NVCC.

Licensing & Attributions

CC licensed content, Original

- Introduction to Computer Applications and Concepts Computer Security Tutorials. **Authored by:** Jim Shannon. **Provided by:** Extended Learning Institute (ELI), Northern Virginia Community College (NOVA). **Located at:** http://eli.nvcc.edu/. **License:** *CC BY: Attribution*

Public domain content

- Image of blue i. **Authored by:** OpenClipArtVectors. **Located at:** https://pixabay.com/en/information-circle-icon-blue-160885/. **License:** *CC0: No Rights Reserved*

MODULE 8: MICROSOFT WORD

READING: MS OFFICE 2013

https://oercommons.org/courses-microsoft-office-2013

Licensing & Attributions

CC licensed content, Shared previously

- License: *Public Domain: No Known Copyright*

READING: COMPUTER SKILLS FOR SUCCESS - A STEP BY STEP GUIDE

https://www.oercommons.org/authoring3779-computer-skills-for-success-a-step-by-step-guide

Licensing & Attributions

CC licensed content, Shared previously

- License: *Public Domain: No Known Copyright*

READING: HOW TO RESET MS 2013 - INTERFACE - MOST EXCELLENT WORD TIPS

https://www.oercommons.org/
search?f.search=Most+Excellent+Word+Tips=&f.general_subject=&f.sublevel=&f.alignment_standard=

Licensing & Attributions

CC licensed content, Shared previously

- License: *Public Domain: No Known Copyright*

READING: BUSINESS LETTER EVALUATION

https://www.oercommons.org/authoring/11959-business-letter-evaluation

Licensing & Attributions

CC licensed content, Shared previously

- License: *Public Domain: No Known Copyright*

READING: CREATING A RESUME

https://www.oercommons.org/courses/creating-a-resume

Licensing & Attributions

CC licensed content, Shared previously

- License: *Public Domain: No Known Copyright*

SUPPLEMENTAL TRAININGS: MICROSOFT OFFICE

For additional MS Office Tutorials, click on the links below:

- 2013 MS Office Tutorials
- 2010 Atomic Learning MS Word Tutorials
- 2010 Office Video Tutorials

Licensing & Attributions

CC licensed content, Original

- Introduction to Computer Applications and Concepts Word Tutorials. **Authored by:** Jim Shannon. **Provided by:** Extended Learning Institute of Northern Virginia Community College. **Located at:** http://www.nvcc.edu/eli/index.html. **License:** *CC BY: Attribution*

Public domain content

- Image of file folder and magnifying glass. **Authored by:** ClkrFreeVectorImages. **Located at:** https://pixabay.com/en/folders-glass-zoom-lens-search-25133/. **License:** *CC0: No Rights Reserved*

TUTORIALS: WORD BASICS

Read, view, and complete the following tutorials at GCFLearnFree.org.

Word Basics

1. #4 – GETTING TO KNOW WORD
2. #5 – CREATING AND OPENING DOCUMENTS
3. #6 – SAVING AND SHARING DOCUMENTS
4. #7 – TEXT BASICS
5. #8 – FORMATTING TEXT
6. #9 – PAGE LAYOUT
7. #10- PRINTING DOCUMENTS

Optional Additional Tutorials are available from GCF Learnfree.org. You can find them by scrolling down to the bottom of the page and looking under *Doing More with Word* and *Extras*.

Licensing & Attributions

CC licensed content, Original

- Introduction to Computer Applications and Concepts Word Tutorials. **Authored by:** Jim Shannon. **Provided by:** Extended Learning Institute (ELI), Northern Virginia Community College (NOVA). **Located at:** http://eli.nvcc.edu/. **License:** *CC BY: Attribution*

Public domain content

- Image of file folder and magnifying glass. **Authored by:** ClkrFreeVectorImages. **Located at:** https://pixabay.com/en/folders-glass-zoom-lens-search-25133/. **License:** *CC0: No Rights Reserved*

TUTORIALS: WORKING WITH TEXT

Read, view, and complete the following tutorials at GCFLearnFree.org.

Working with Text

1. #11- Indents and tabs
2. #12- Line and paragraph spacing
3. #13- Lists
4. #14- Hyperlinks
5. #15- Breaks
6. #16- Columns
7. #17- Headers, footers, and page numbers

Optional Additional Tutorials are available from GCF Learnfree.org. You can find them by scrolling down to the bottom of the page and looking under *Doing More with Word* and *Extras*.

Licensing & Attributions

CC licensed content, Original

- Introduction to Computer Applications and Concepts, Introduction to Windows Tutorials. **Authored by:** Jim Shannon. **Provided by:** Extended Learning Institute (ELI), Northern Virginia Community College (NOVA). **Located at:** http://eli.nvcc.edu/. **License:** *CC BY: Attribution*

Public domain content

- Image of file folder and magnifying glass. **Authored by:** ClkrFreeVectorImages. **Located at:** https://pixabay.com/en/folders-glass-zoom-lens-search-25133/. **License:** *CC0: No Rights Reserved*

TUTORIALS: WORKING WITH OBJECTS

Required Trainings

Read, view, and complete the following tutorials at GCFLearnFree.org.

Working with Objects

1. #18- Pictures and text wrapping
2. #19- Formatting pictures
3. #20- Shapes
4. #21- Text boxes and word art
5. #22- Arranging objects
6. #23- Tables
7. #24- Charts

Optional Additional Tutorials are available from GCF Learnfree.org. You can find them by scrolling down to the bottom of the page and they are under *Doing More with Word* and *Extras*.

Licensing & Attributions

CC licensed content, Original

- Introduction to Computer Applications and Concepts Word Trainings Part II. **Authored by:** Jim Shannon. **Provided by:** Extended Learning Institute of Northern Virginia Community College. **Located at:** http://www.nvcc.edu/eli/index.html. **License:** *CC BY: Attribution*

Public domain content

- Image of file folder and magnifying glass. **Authored by:** ClkrFreeVectorImages. **Located at:** https://pixabay.com/en/folders-glass-zoom-lens-search-25133/. **License:** *CC0: No Rights Reserved*

TUTORIALS: REVIEWING DOCUMENTS AND COLLABORATION

Required Trainings

Read, view, and complete the following tutorials at GCFLearnFree.org.

Reviewing Documents and Collaboration

1. #25- checking spelling and grammar
2. #26- track changes and comments
3. #27- finalizing and protecting documents

Optional Additional Tutorials are available from GCF Learnfree.org. You can find them by scrolling down to the bottom of the page and they are under *Doing More with Word* and *Extras*.

Licensing & Attributions

CC licensed content, Original

- Introduction to Computer Applications and Concepts. **Authored by:** Jim Shannon. **Provided by:** Extended Learning Institute (ELI), Northern Virginia Community College (NOVA). **Located at:** http://eli.nvcc.edu/. **License:** *CC BY: Attribution*

Public domain content

- Image of file folder and magnifying glass. **Authored by:** ClkrFreeVectorImages. **Located at:** https://pixabay.com/en/folders-glass-zoom-lens-search-25133/. **License:** *CC0: No Rights Reserved*

MODULE 9: HTML

READING: HTML TRAINING

Read the following sections in the HTML Tutorial.

- Starting
- Creating
- Naming
- Viewing
- Revising
- Validation
- FTP
- Linking
- Images
- Lists

Licensing & Attributions

CC licensed content, Original

- HTML. **Authored by:** Judy Jacobs Miller. **Provided by:** ITE 115 at Northern Virginia Community College. **Located at:** http://jumiller@nvcc.edu. **License:** *CC BY-NC-ND: Attribution-NonCommercial-NoDerivatives*

READING: CREATING YOUR FIRST WEB PAGE

Getting Started

So you want to make a web site. It's easy. You don't need any fancy software. Just a computer and you're set. The software you will use is already on most computers that are sold today. So what do we need?

1. A text editor
2. A browser

A text editor is a program that let's you write text. Actually, ASCII text. What's ASCII text? Basically, the characters on your keyboard. On Windows computers the text editor program is called Notepad. You open it up by selecting: Start > Programs > Accessories > Notepad. Don't confuse a text editor with a word processor. Using a word processor, like Word Pad or Word, will mess you up. Be sure you use Notepad.

A browser is software that let's you view a web page. It takes what's written on Notepad and interprets it. The browser shows you all the formatting; it's what makes the page look pretty, or ugly, or whatever. There are lots of browsers. Two popular browsers are Internet Explorer and Firefox. To open a browser, either double click on the browser icon on your desktop, or select Start > Programs > Internet Explorer (or Firefox).

Let's do a little experiment. Open a web page, say http://www.amazon.com. Now that you have Amazon open, right-click on the web page and choose "View Source" or "View Page Source". What do you see? Lots of HTML code. Don't worry if it looks a little complicated. It's really pretty easy. You'll make a web page in less than an hour. Just keep reading.

So what have we learned? You create a web page using Notepad. You view a web page using a browser. Lets make a web page!

Creating Your First Web Page

The best way to make a web page is to just dive right in. Open Notepad. To open notepad in Windows 7 and before: Start > Programs > Accessories > Notepad. To open notepad in Windows 8: Move your mouse over the start button and then into the bottom left corner. This brings up the new short menu .Click the Apps button at the bottom and you get a list of all installed apps in alphabetical order. At the bottom is notepad.

Once notepad is open, start typing (or if you're lazy, like me, just copy and paste):

```
<html>

<head>
<title>The title goes here</title>
</head>

<body>

<h1>This is a level one heading</h1>

<p>My first lovely paragraph. Wow. This is fun.</p>

<p>My second lovely paragraph. We can make paragraphs forever.</p>

<h2>This is a level two heading</h2>

<p>This is my third paragraph. Notice how the paragraph tag sets off the text. Also notice that every
beginning tag has an ending tag.</p>

</body>

</html>
```

What does all this mean? The stuff between the < and the > symbols are "tags." Tags (mostly) come in pairs, a beginning tag and an ending tag. The ending tag repeats the beginning tag, with a forward slash in it. For example, the web page starts with <html> and ends with </html>.

Each web page has two parts:

1. The head, which begins with <head> and ends with </head>, and
2. The body, which begins with <body> and ends with </body>

Beginning to get the idea? It's easy.

So what goes in the head? The title. The title is what appears at the top of the browser, in the blue “title bar.” Everything else goes in the body, that is, between the body tags.

Most of what you have to say goes in paragraphs, just like in real life. A paragraph starts with <p> and ends with </p>. You can also put stuff between heading tags, which come in various sizes, from <h1>, <h2>, … through <h6>. Headings are usually used for, well, headings. The <h1> heading will make text bold and large. <h2> is a little smaller, and so forth. <h6> headings are quite small. Now, it may seem like a heading tag would go in the head, but it doesn’t. Headings go in the body, just like everything else (except the title).

After you create the web page, you must name it and save it. Naming a web page the right way is critical, so we devote a whole section to naming. Read on.

Licensing & Attributions

CC licensed content, Shared previously

- Judy's HTML5 Tutorial. **Authored by:** Judy Jacobs Miller. **Provided by:** Extended Learning Institute (ELI), Northern Virginia Community College (NOVA). **Located at:** http://www.nvcc.edu/home/jumiller/html_tutorial/creating.html. **License:** *CC BY-NC-ND: Attribution-NonCommercial-NoDerivatives*

READING: NAMING AND VIEWING YOUR WEB PAGE

Naming Your Page

Before you name your web page, create a folder to hold all of your web pages. The folder can be anywhere (a: drive, c: drive, flash drive), it doesn’t mater. Just be sure you know where it is so you can find it again. Name the folder “website”. Now you can save the text file that you just created using Notepad in the folder called “website.” As you save the text file, name it “first.html”.

Naming a web page correctly is easy if you follow these three rules:

1. No spaces
2. No capitals
3. The file extension is .html (that’s dot html)

If you want to know the why behind these rules, read on.

Spaces in filenames can confuse browsers and email programs. Browsers usually substitute “%20” for a space. Email programs sometimes cut off URLs after a space. If you want to use a space for clarity, use an underscore instead. For example, “onion recipe.html” becomes “onion_recipe.html”

Capitals matter. You’re probably creating your first web page in a Windows operating system environment. If so, Windows is pretty lax about file names. But once you’re done creating your page, and you upload it to a server (so that Aunt Tilly in Iowa can see it), the server may be using an entirely different operating system, like Unix or Linux. Unix and Linux care about capitals. MyFile.html is a totally different file than myfile.html in a Unix or Linux environment. You can use capitals in your file names, but when you create your links (which we’ll get to real soon), they won’t work if your link file name doesn’t match your actual file name. If you use all lowercase, you don’t have to try to remember “Did I use a capital or not?”

File extensions. Extensions matter. Your web page extension must be either .htm or .html. It doesn’t matter which, but you’ve got to be consistent. If you create a link to myfile.html and the page is named myfile.htm, the link won’t work.

Ok, now that we’ve saved our first web page with the name “first.html” let’s take a look at it in the browser.

Viewing Your Web Page in a Browser

Now that we've created our web page with Notepad, and saved it as "first.html", we want to see what it looks like in the browser. The easiest way to open a webpage in a browser is to double-click on the filename in "Windows Explorer" or "My Computer." You can also right-click the name of the file, then choose either "Firefox" or "Internet Explorer."

There, you've done it. You've made your first web page, and viewed it in the browser. Don't worry, we'll soon learn how to jazz it up.

Licensing & Attributions

CC licensed content, Shared previously

- Judy's HTML5 Tutorial. **Authored by:** Judy Jacobs Miller. **Provided by:** Extended Learning Institute (ELI), Northern Virginia Community College (NOVA). **Located at:** http://www.nvcc.edu/home/jumiller/html_tutorial/naming.html. **License:** *CC BY-NC-ND: Attribution-NonCommercial-NoDerivatives*

READING: REVISING YOUR WEB PAGE

Ok, your page is a bit boring. For example, the title bar says "The title goes here." Let's put in a better title. Go back to Notepad. You've still got your web page open in Notepad, don't you? If not, don't worry. Just open the webpage in notepad to see the html code. The easiest way to open the webpage in notepad is to right-click on the filename in "Windows Explorer" or "My Computer," then choose Open With > Notepad.

Now change the title to something good, like "Marvin's awesome first page", or whatever. If you're not Marvin you might want to change the name. Also, change the level one heading (that's the stuff between the beginning <h1> tag and the ending </h1> tag) to something better. Maybe "Marvin Rocks" or whatever.

Now you've made the changes in Notepad. But you don't see "Marvin Rocks" in your browser. There are two things you have to do to make the changes in Notepad show up in the browser.

1. Do a "File > Save" in Notepad
2. Hit the "refresh" or "reload" button on your browser

Awesome. You did it. You made your first web page.

Licensing & Attributions

CC licensed content, Shared previously

- Judy's HTML5 Tutorial. **Authored by:** Judy Jacobs Miller. **Provided by:** Extended Learning Institute (ELI), Northern Virginia Community College (NOVA). **Located at:** http://www.nvcc.edu/home/jumiller/html_tutorial/revising.html. **License:** *CC BY-NC-ND: Attribution-NonCommercial-NoDerivatives*

READING: VALIDATION

You may think that if your web page shows up in the browser, the code is correct. Not so. Browser software is designed to attempt to correct coding errors. Incorrect code may appear the way you intended in one browser, may appear odd in another browser, and may not appear at all in a third browser.

There's an easy way to find out if your code is correct. Validate! The World Wide Web Consortium offers a nifty validator. It's at http://validator.w3.org/. Now, there's one little catch. You have to tell the validator what version of html you're using, so it knows what to validate. That's easy, just add a DOCTYPE at the top of your web page. You also have to add language information to the html tag, as well as a meta tag which indicates the character set. Well, all this technical stuff is beyond me, so I just copy and paste.

```
<!DOCTYPE html>
<html lang=”en”>
<head>
<title>The title goes here</title>
  <meta charset=”utf-8″>
</head>

<body>

<h1>This is a level one heading</h1>

<p>My first lovely paragraph. Wow. This is fun.</p>

<p>My second lovely paragraph. We can make paragraphs forever.</p>

<h2>This is a level two heading</h2>

<p>Thi is my third paragraph. Notice how the paragraph tag sets off the text. Also notice that every beginning tag has
an ending tag.</p>

</body>

</html>
```

Do not try to type this. Just copy and paste it at the top of the web page you're making in Notepad. Be sure to remove the old <html> tag that you had, so you don't have two html tags. Don't forget to save the file again.

The DOCTYPE tells the validator what version of HTML you are using. There are several different versions of html. We're learning HTML5.

Now you're ready to validate. Here's the link again: http://validator.w3.org/. When you get to the W3C validation page, click the tab that says “Validate by File Upload.” Browse to the correct file, then click the “Check” button.

If the code validates, you'll see “This document was successfully checked as HTML5!” on a green background. Green is good. If the code does not validate, you'll see ” Error found while checking this document as HTML5! ” on a red background. Red is bad. If your code does not validate, the validator will tell you what is wrong. Read the feedback from the validator, change your code in Notepad, save, then validate again.

The validator generates error messages that are usually pretty helpful, but because of the way it parses documents, occasionally its feedback is cryptic. You'll get used to this, but here are some tips to help you get started:

- Validation errors cascade. Don't get discouraged if you see you have 36 errors in your document! Fix the first one, then validate again. This will often fix many of the following errors, especially those that pertain to the structure of the document.
- You can have only one html statement per document. If you copy and paste <html lang=”en”> into your document, take out the old <html>.
- You can only have one body tag in your document. Later, when we get to styling the body tag, be sure you don't have two body tags.
- The validator may object to code which is correct, but is in the wrong location. For example

```
<!DOCTYPE html>
<html lang=”en”>
<head>
<title>The title goes here</title>
<meta charset=”utf-8″>
</head>

<body>

<h1>This is a level one heading</h1>

<p>This is BAD CODE EXAMPLE. Don’t copy it</p>

<p>My second lovely paragraph. </p>

</body>

<h2>This is a level two heading</h2>

<p>Thi is my third paragraph. </p>

</html>
```

The above code generates the following validation error:

✖ *Line 19, Column 4*: Stray start tag h2.

```
<h2>This is a level two heading</h2>
```

There is nothing wrong with the code for the h2 statement; the validator is objecting to its location. The previous statement, the </body> statement, is in the wrong place. The closing body tag must come after all statements and right before the </html> tag.

Licensing & Attributions

CC licensed content, Shared previously

- Judy's HTML5 Tutorial. **Authored by:** Judy Jacobs Miller for ITE 115 at Northern Virginia Community College. **Provided by:** Extended Learning Institute (ELI), Northern Virginia Community College (NOVA). **Located at:** http://www.nvcc.edu/home/jumiller/html_tutorial/validation.html. **License:** *CC BY-NC-ND: Attribution-NonCommercial-NoDerivatives*

READING: FTP

You made a web page. Great! You probably created the web page on the C: drive or on a flash drive. It looks good and it validates. Now you want your Aunt Tilly in Iowa to see the page. How are you going to do this? Aunt Tilly cannot see your C: drive or your flash drive. In order for Aunt Tilly to see the web page, you must put it on a web server. A web server is a computer that’s hooked up to the Internet (hopefully with a reliable, fast connection) and that has special server software on it. The server software and fast Internet connection allow this computer to act as a web server. So it can serve up web pages, just like a waiter serves up your food. Well, sort of. The web pages don’t taste as good.

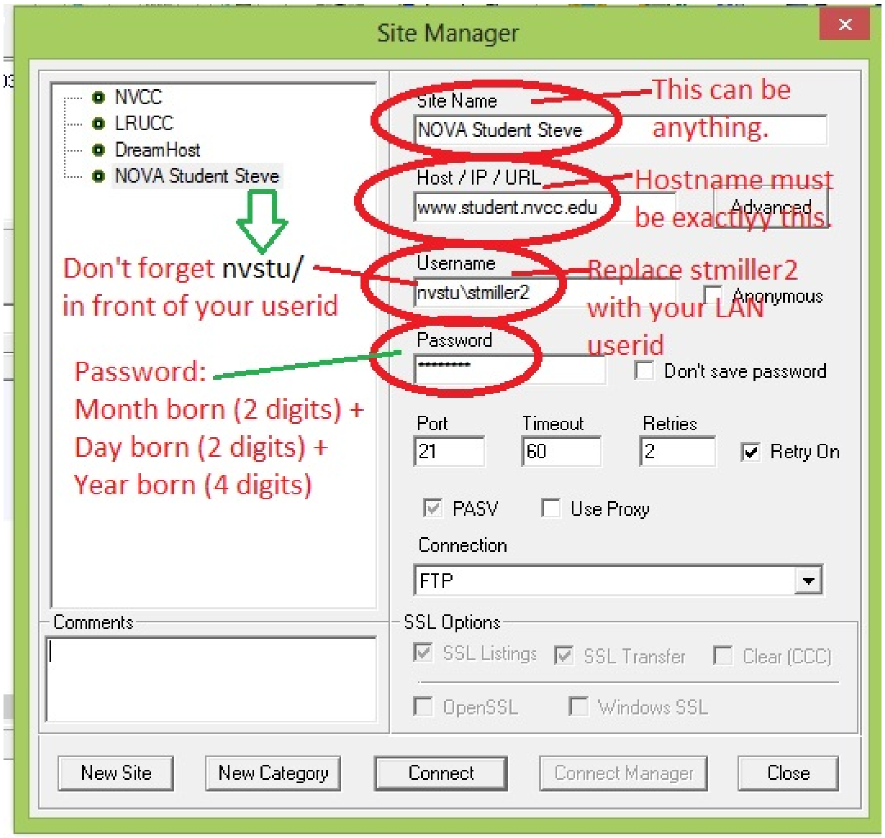
So how do you get your web page to the server? There are several ways. One way is to use an FTP program. FTP stands for “file transfer protocol,” which is one of the protocols, or set of rules, used to transfer files on the Internet.

At Northern Virginia Community College we use an FTP program called Core FTP LIte. Why? Because it’s cheap and it’s good. Core FTP is already loaded on the computers in the NVCC open labs. If you want to use the program at home, you can download it from www.coreftp.com. (Click on the “Download” tab, then download Core FTP LE, which is the free version.)

All NVCC students have 10 MB of server space. It’s free, and it will exist as long as you are enrolled in classes. In order to send your web page to your NVCC server space, you need to know three things:

1. Hostname: www.student.nvcc.edu
   - This is the name of NVCC’s student web server.
2. NVCC LAN ID : look it up here: https://www.nvcc.edu/stu_id/
   - In front of your LAN ID goes nvstu/
   - So if your LAN ID is stmiller2, your userid for FTP is: nvstu/stmiller2
3. Password: 8 digits long
   - The first 2 digits are the month you were born
   - The next 2 digits are the day you were born
   - The next 4 digits are the year you were born

Let’s get started. Open the Core FTP Lite program by double-clicking on the icon on the desktop. If the Site Manager Window is not open, open it by clicking on ” File > Connect.” In the Site Manager window, click the New Site button in the lower left corner. Fill in the following information:



Site Name: NOVA Student Steve

Host / IP / URL: www.student.nvcc.edu

Username: nvstu\stmiller2 □ Anonymous

Password: ******** □ Don't save password

Port: 21 Timeout: 60 Retries: 2 ☑ Retry On

☑ PASV □ Use Proxy

Connection: FTP

SSL Options: ☑ SSL Listings ☑ SSL Transfer □ Clear (CCC) □ OpenSSL □ Windows SSL

This can be anything.

Hostname must be exactly this.

Don't forget nvstu/ in front of your userid

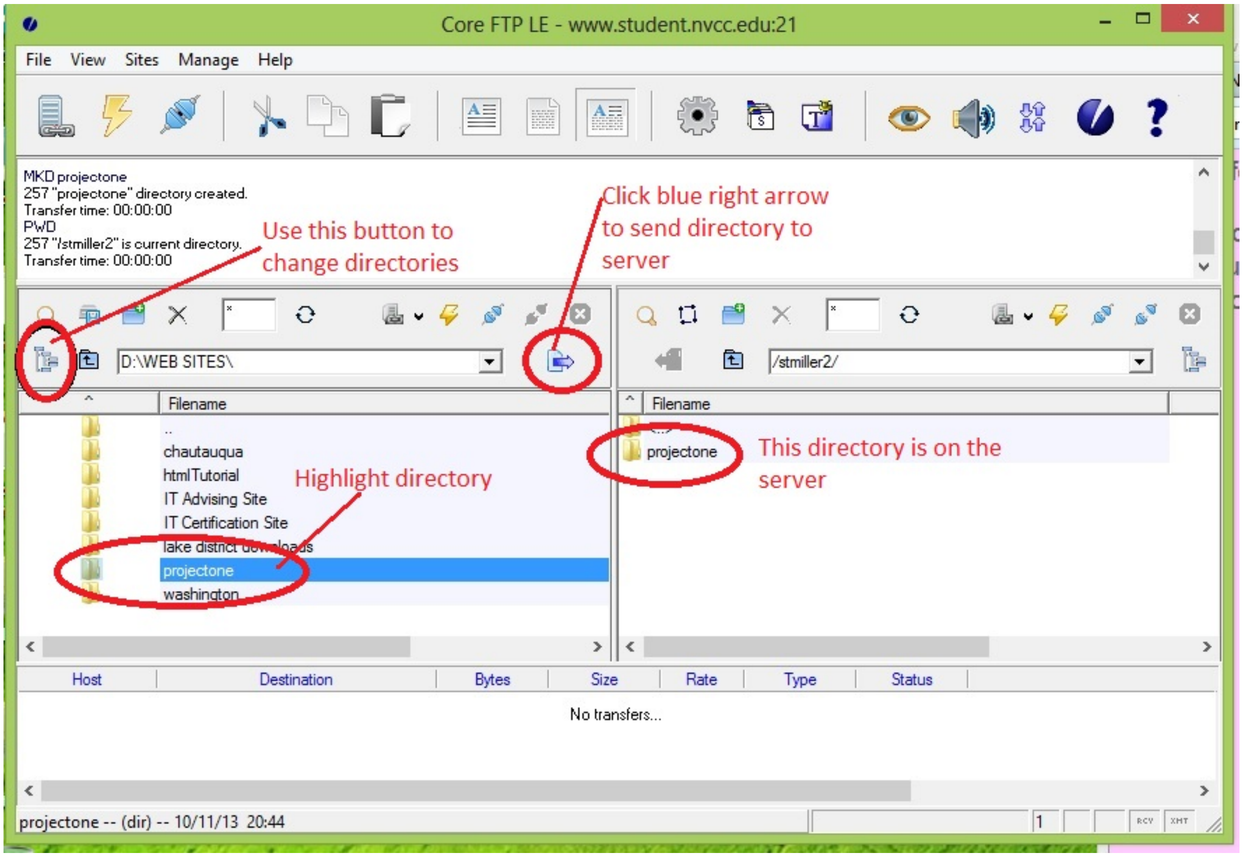
Replace stmiller2 with your LAN userid

Password: Month born (2 digits) + Day born (2 digits) + Year born (4 digits)

Click the Connect button. The files on the left are the the files on your local computer. The files on the right are the files on the server. You’ll know you connected successfully if you see your LAN id on the right. Each student has a folder, the name of which is his or her LAN id, on the NVCC student server. All of the student folders are contained in a folder called “home.” If you were unable to connect successfully, call (not email) the IT help desk at 703.426.4141.

You navigate around your files in the CORE program much as you would navigate around Windows Explorer. Double click on a folder to open it. Double click on the two dots at the top of the folder listing to see the parent folder. The only tricky thing is when you want to see another drive, such as your A: drive or your flash drive. There is a little icon on the left, called Directory Tree. I think it’s supposed to look like a folder tree in Windows Explorer. If you click it you will see all of your drives.

To upload your web pages to the NVCC student server, first highlight the local file or folder on the left side of the CORE window, and click the blue right arrow. The file or folder will be transferred to the server on the right. After an upload, you should see the file or folder name listed on the right.



```
MKD projectone
257 "projectone" directory created.
Transfer time: 00:00:00
PWD
257 "/stmiller2" is current directory.
Transfer time: 00:00:00
```

Use this button to change directories

Click blue right arrow to send directory to server

Highlight directory

This directory is on the server

Once you've uploaded your web page, Aunt Tilly in Iowa, or anyone in the world connected to the Internet, can see it. How? By typing the URL in the web browser address bar:

http://www.student.nvcc.edu/home/YourUserID/projectone/

Of course, replace YourUserID with your actual userid, and replace the directory namel with the actual name of the directory you are using. Your files need not be in a directory if you are only creating one website.

Always open your browser and check out your web page as it appears on the server. Sometimes things look great on your local computer, but look different on the server.

Licensing & Attributions

CC licensed content, Shared previously

- Judy's HTML5 Tutorial. **Authored by:** Judy Jacobs Miller. **Provided by:** Extended Learning Institute (ELI), Northern Virginia Community College (NOVA). **Located at:** http://www.nvcc.edu/home/jumiller/html_tutorial/ftp.html. **License:** *CC BY-NC-ND: Attribution-NonCommercial-NoDerivatives*

READING: LINKING

Linking two pages together requires the following code:

<a href="filename.html">words for the user to click on</a>

Where do you put this code? Where you want the link to appear. Make sure it’s between the body tags, just as all other content on you web page should be.

The filename.html gets replaced by the name of the file you are linking to. The filename in the link must match exactly the actual name of the file. If you’ve named your file as I’ve suggested, it has no capitals, no spaces, and ends with .html. That makes it easier when you’re linking.

The “words for the user to click on” should be something that describes where the link will take the user. So “Click Here!” would not be very helpful to the user, whereas “Learn about zebras” would give the user a clue about where the link leads.

For example, if you are linking to a file which contains a description of your favorite movies, the link code might look like this:

```
Learn about Judy’s <a href=”fav_movies.html”> favorite movies</a>.
```

This code will work only when the file to which you are linking is in the same folder as the file from which you are linking. If the files are in different folders, you must do some relative pathing, which is beyond the scope of this tutorial.

Once you’ve created the link, test it out, twice. First, test it on your local computer. Then, test it on the server. Don’t forget to send both files to the server: the file you are linking from and the file you are linking to. If your link doesn’t work, check the following.

- Is the filename in the link code exactly the same as the actual filename?
    - Beware: sometimes you don’t see the entire filename on your computer. The Windows operating system typically hides file extensions. To see your file extensions, open Windows Explorer and select: Tools > Folder Options > View, or Organize > Folder and Search Options > View. Then uncheck “Hide extensions for known file types.”
- Are both the file you are linking from and the file you are linking to in the same folder?
    - Don’t forget to send both files to the server.
- Never start the filename in your link code with c:/ or e:/ or any other drive.
    - This may work on your local computer, but when a user on another computer tries to access the link, it will fail. The server doesn’t have a clue what is on your computer.

If you want to link to another web site, the link code should include the entire URL, as follows:

```
For a great place to buy books, visit <a href=”http://www.amazon.com”>Amazon.</a>
```

Links within your site will also work if you start them with http:// … etc. For example, you could link to a page within your site as follows:

```
<a href=” http://www.student.nvcc.edu/home/YourUserID/fav_movies.html”>Link to My Favorite Movies</a>
```

The link would work, but this is a bad way to link to pages within your own site, for two reasons: 1) link access time is increased and 2) when you want to move your web site to another server, you have to change all your links.

Licensing & Attributions

CC licensed content, Shared previously

- Judy's HTML5 Tutorial. **Authored by:** Judy Jacobs Miller. **Provided by:** Extended Learning Institute (ELI), Northern Virginia Community College (NOVA). **Located at:** http://www.nvcc.edu/home/jumiller/html_tutorial/linking.html. **License:** *CC BY-NC-ND: Attribution-NonCommercial-NoDerivatives*

READING: IMAGES

Adding images can really spice up a web page. Images which appear on web pages are typically .jpg, .gif, or .png files. These three popular compression formats are universally recognized by browsers. Do not use another format, such as .bmp.

Please make sure you either own the image or have permission from the owner to use the image. Here are a few web sites that have some free images available for your use:

- Corbis
- Digital Blasphemy
- GettyImages
- Irfanview
- MorgueFile
- PhotoshopSupport

To download an image, place your cursor over the image, right-click, and select “Save Picture As” or “Save Image As.” Save the image in the same folder as the web page on which you want the image to appear. The following code will add an image to your web page:

```
<img src=”imagename.jpg” alt=”alternative text” height=”100″ width=”200″>
```

Imagename.jpg gets replaced by the name of the image file. The alt attribute provides alternative text for those users who cannot or prefer not to view images. The height and width attributes should reflect the actual height and width of the image, and are used by the browser as it renders the page. Height and width should not be used to resize the image; use image editing software (such as Paint, Fireworks or Photoshop) to resize.

For example, if you want an image of a turtle to appear on your web page, the code might look like this

```
<img src=”turtle.jpg” alt=”Pretty Box Turtle” height=”100″ width=”200″>
```

The image code will work only when the image file and the html file are in the same folder. If the files are in different directories, you must do some relative pathing, which is beyond the scope of this tutorial.

If the image does not appear on your web page, check the following:

- Is the filename in the image code exactly the same as the actual filename?
    - Beware: sometimes you don’t see the entire filename on your computer. The Windows operating system typically hides file extensions. To see your file extensions, open Windows Explorer and select: Tools > Folder Options > View, or Organize > Folder and Search Options > View. Then uncheck “Hide extensions for known file types.”
- Are both the html file and the image file in the same folder?
    - Don’t forget to send the image file to the server, along with the html file.
- Never start the image filename with c:/ or e:/ or any other drive.
    - This may work on your local computer, but when a user on another computer tries to view the web page, the image will not appear. The server doesn’t have a clue what is on your computer.

Licensing & Attributions

CC licensed content, Shared previously

- Judy's HTML5 Tutorial. **Authored by:** Judy Jacobs Miller for ITE 115 at Northern Virginia Community College. **Provided by:** Extended Learning Institute (ELI), Northern Virginia Community College (NOVA). **Located at:** http://www.nvcc.edu/home/jumiller/html_tutorial/images.html. **License:** *CC BY-NC-ND: Attribution-NonCommercial-NoDerivatives*

READING: LISTS

Let’s make some lists. First we’ll make an ordered list; this is a list that has numbers. Here’s how:

```
<!DOCTYPE html>
<html lang=”en”>
<head>
<title>List Example</title>
<meta charset=”utf-8″>
</head>
```

```
<body>

<h1>Judy’s Shopping List </h1>

<ol>
<li>Milk</li>
<li>Sugar</li>
<li>Candy</li>
<li>Bread</li>
</ol>

</body>

</html>
```

Give it a try. Copy the code into Notepad, save it with the name “shop.html”, then open it in the browser and see how it looks.

You might have figured out that the “ol” in the code stands for “ordered list.” The “li” in the code stands for “list item.” Notice, every beginning tag has an ending tag.

Now try an unordered list: replace the <ol> tag with <ul>, and the </ol> tag with </ul>. Yes, “ul” stands for “unordered list.”

Licensing & Attributions

CC licensed content, Shared previously

- Judy's HTML5 Tutorial. **Authored by:** Judy Jacobs Miller. **Provided by:** Extended Learning Institute (ELI), Northern Virginia Community College (NOVA). **Located at:** http://www.nvcc.edu/home/jumiller/html_tutorial/lists.html. **License:** *CC BY-NC-ND: Attribution-NonCommercial-NoDerivatives*

READING: TABLES

Here’s how to add a simple table to your web page. This table has two rows and three cells in each row:

```
<!DOCTYPE html>
<html lang=”en”>
<head>
<title>The title goes here</title>
<meta charset=”utf-8″>
<title>Tables are Fun </title>
<style type=”text/css”>

table
{
border-style: solid;
}

td
{
border-style: solid;
border-color: #FF66FF;
padding: 10px;
}
</style>
</head>
```

```
<body>
<h1>My Lovely Table </h1>
<table>
<tr>
<td>1st Cell</td>
<td>2nd Cell</td>
<td>3rd Cell</td>
</tr>
<tr>
<td>4th Cell</td>
<td>5th Cell</td>
<td>6th Cell</td>
</tr>
</table>
</body>
</html>
```

Let's analyze this code. The <table> tag starts the table, the <tr> tag starts the row, and the <td> tag starts the cell. (I think it stands for "table data.") Every opening tag has a corresponding ending tag. I've used embedded CSS to style the <table> and <td> tags. The table itself has a solid border. Each cell appears with a solid pink border, and each cell has 10 pixels of padding. Cell padding is the space between the contents of the cell and the cell border.

Licensing & Attributions

CC licensed content, Shared previously

- Judy's HTML5 Tutorial. **Authored by:** Judy Jacobs Miller. **Provided by:** Extended Learning Institute (ELI), Northern Virginia Community College (NOVA). **Located at:** http://www.nvcc.edu/home/jumiller/html_tutorial/tables.html. **License:** *CC BY-NC-ND: Attribution-NonCommercial-NoDerivatives*

READING: STYLING

Cascading Style Sheets add style to your web pages. There are three different kinds of styles:

1. inline
2. embedded (also called "internal")
3. external

External styling is by far the best way to style, for reasons we'll discuss in just a bit. Occasionally, however, you may want to do a bit of inline or embedded styling, so we'll cover them briefly.

Inline CSSWith inline styles, you change the appearance of one tag. For example, if you want to make the body have a pink background color, you change the body tag, like this:

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Styling Example</title>
<meta charset="utf-8″>
</head><body style="background-color: #FFCCFF"><h1>This is a level one heading</h1>

<p>My first lovely paragraph. Wow. This is fun.</p>

</body>

</html>
```

If you want to make your page have navy blue font, you do it like this:

```
<body style=”color: #000066″>
```

And if you want to make your page have a pink background and navy blue font, it’s like this:

```
<body style=”background-color: #FFCCFF; color: #000066″>
```

What are those funny numbers that change the color? They’re the hexadecimal representation of the amount of red, green and blue that make up the color. You can find the hexadecimal representation of colors at http://www.visibone.com/colorlab/.

You can take any tag and apply styling. If you want to make a paragraph have a red Arial font that is 18 pixels high, it’s like this:

```
<p style=”color: red; font-family: Arial; font-size: 18px”>Hello</p>
```

If you want to make a level two heading have a red background, white font 24 pixels high, and be aligned to the center it’s like this:

```
<h2 style=”text-align: center; color: #FFFFFF; background-color: #FF0000; font-size: 24px”>Awesome</h2>
```

Notice you can use the name of a color, like red or white, rather than the hexadecimal value, for the more common colors. But you’ll have a greater selection of colors and more predictability by using hexadecimal values.

Do you like to style? You can find many more style properties and values at http://www.w3schools.com/cssref/default.asp. Here are a few of the more common properties and values:

Property	Possible Values
background-color	hex value
border-color	hex value
border-style	solid, double, dotted
color	hex value
font-family	arial, courier, etc
font-size	# of pixels
letter-spacing	# of pixels
line-height	# of pixels
text-align	right, left, center

Embedded CSS

Above we learned that Cascading Style Sheets come in three types:

- Inline (we learned this already)
- Embedded (also called “internal”)
- External (we’ll cover this in just a bit)

We have already learned how to do inline styles: the style code is placed within each tag, and affects one tag and one tag only. With embedded CSS, we specify the style for a particular tag, and every time that tag is used on a page, the style is applied. Embedded CSS is placed in the head of the document, right after the ending title tag, like this:

```
<!DOCTYPE html>
<html lang=”en”>
<head>
<title>The title goes here</title>
<meta charset=”utf-8″>
<style type=”text/css”>h2
{
color: #FF0000;
}p
{
color: #0000FF;
font-family: Verdana;
}

</style>
</head>

<body>

<h2>Cardinals</h2>

<p>The male cardinal is a beautiful bright red, whereas the female is a less colorful brown.</p>

<h2>Bluebirds</h2>

<p>Bluebirds are a beautiful shade of blue, but are difficult to attract to your garden. </p>

</body>
</html>
```

The above code says make every level two heading in the document appear in red text, and make every paragraph on the page appear in blue text with a Verdana style font.

Embedded styles are convenient if you want several tags on your page to appear the same. If you use inline and embedded styles for the same tag, the inline style will take precedence. For example, you could use embedded CSS to make every paragraph appear blue (as we did above), then use an inline style to make just one paragraph appear green.

External CSS

Just a word or two about external CSS. It is by far the most powerful of the three types of CSS, and is also the easiest to use when you’re styling a website with lots of pages. With external CSS, you don’t put the CSS styling in the html document; you pull it out and put it in another document that has a .css extention. Each html page gets “linked” up to the external css page. This way, you can create one CSS document which styles hundreds of pages. It’s also alot easier to maintain, because if you decide to change the color of all level three headings in your site, you can do it with one new line of code. If you’re interested in learning about external CSS, visit my CSS Tutorial

Licensing & Attributions

CC licensed content, Shared previously

- Judy's HTML 5 Tutorial. **Authored by:** Judy Jacobs Miller. **Provided by:** Extended Learning Institute (ELI), Northern Virginia Community College (NOVA). **Located at:** http://www.nvcc.edu/home/jumiller/html_tutorial/styling.html. **License:** *CC BY-NC-ND: Attribution-NonCommercial-NoDerivatives*

READING: HTML SUMMARY

We’ve learned how to create web pages with images, links, lists, tables, and styles. We’ve also learned how to ftp our pages to the server. In short, we’ve made a web site. Here’s a list of the HTML tags we’ve used:

- html
  - head
    - title
  - body
    - h1 through h6 (headings)
    - p (paragraphs)
    - ol (ordered list)
      - li (list item)
    - ul (unordered list)
      - li (list item)
    - img (image) – requires a src attribute to indicate the source file of the image
    - a (anchor – makes a link) – requires an ” href” attribute to indicate where the link goes
    - table
      - tr (table row)
      - td (table data – makes a cell)

Have fun building your web sites!

Licensing & Attributions

CC licensed content, Shared previously

- Judy's HTML5 Tutorial. **Authored by:** Judy Jacobs Miller. **Provided by:** Extended Learning Institute (ELI), Northern Virginia Community College (NOVA). **Located at:** http://www.nvcc.edu/home/jumiller/html_tutorial/summary.html. **License:** *CC BY-NC-ND: Attribution-NonCommercial-NoDerivatives*

MODULE 10: MICROSOFT EXCEL

READING: WHAT IS EXCEL?

https://www.oercommons.org/courses/what-is-excel

Licensing & Attributions

CC licensed content, Shared previously

- License: *Public Domain: No Known Copyright*

READING: INTRODUCTION TO SPREADSHEETS

https://www.oercommons.org/courses/introduction-to-spreadsheets-2

Licensing & Attributions

CC licensed content, Shared previously

- License: *Public Domain: No Known Copyright*

READING: INTRODUCTION TO MS EXCEL

https://www.oercommons.org/courses/introduction-to-microsoft-excel

Licensing & Attributions

CC licensed content, Shared previously

- License: *Public Domain: No Known Copyright*

READING: CAR DEPRECIATION OVER 5 YEARS

https://www.oercommons.org/search?f.search=Car+depreciation+over+5+years

Licensing & Attributions

CC licensed content, Shared previously

- License: *Public Domain: No Known Copyright*

READING: HOW TO USE EXCEL - CAREERS IN PRACTICE

https://oer.commons.org/search?f.search=Hot+to+use+MS+Excel%3A+Careers+in+Practice+Series

Licensing & Attributions

CC licensed content, Shared previously

- License: *Public Domain: No Known Copyright*

READING: HOW TO CREATE A CHART IN EXCEL

http://www.wikihow.com/Make-a-Chart-in-Excel

Licensing & Attributions

Public domain content

- License: *Public Domain: No Known Copyright*

READING: EXCEL HELP CHART

https://www.oercommons.org/courses/excel-help-chart

Licensing & Attributions

CC licensed content, Shared previously

- License: *Public Domain: No Known Copyright*

READING: USING IMPORTED DATA IN EXCEL

https://www.oercommons.org/courses/using-imported-data-in-excel-plot-atmospheric-temperature

Licensing & Attributions

CC licensed content, Shared previously

- License: *Public Domain: No Known Copyright*

MICROSOFT OFFICE TUTORIALS: EXCEL

https:www.oercommons.org/courses/microsoft-office-tutorials

Licensing & Attributions

CC licensed content, Shared previously

- License: *Public Domain: No Known Copyright*

TUTORIALS: EXCEL BASICS

https://www.oercommons.org/courses/microsoft-excel-basic-tutorial

Licensing & Attributions

CC licensed content, Shared previously

- License: *Public Domain: No Known Copyright*

TUTORIALS: EXCEL BASICS

Required Trainings

Read, view, and complete the following tutorials at GCF LearnFree.org.

Excel Basics

1. #4 – GETTING TO KNOW excel
2. #5 – CREATING AND OPENING workbooks
3. #6 – SAVING AND SHARING workbooks
4. #7 – cell basics
5. #8 – modifying columns, rows, and cells
6. #9 – formatting cells
7. #10- worksheet basics
8. #11- page layout
9. #12- printing workbooks

Optional Additional Tutorials are available from GCF Learnfree.org under DOING MORE WITH WORD and EXTRAS.

Licensing & Attributions

CC licensed content, Original

- Introduction to Computer Applications and Concepts Excel Training. **Authored by:** Jim Shannon. **Provided by:** Extended Learning Institute of Northern Virginia Community College. **Located at:** http://www.nvcc.edu/eli/index.html. **License:** *CC BY: Attribution*

Public domain content

- Image of file folder and magnifying glass. **Authored by:** ClkrFreeVectorImages. **Located at:** https://pixabay.com/en/folders-glass-zoom-lens-search-25133/. **License:** *CC0: No Rights Reserved*

TUTORIALS: WORKING WITH DATA

Required Trainings

Read, view, and complete the following tutorials at GCF LearnFree.org.

Working with Data

1. #17- freezing panes and views options
2. #18 – sorting data
3. #19 – filtering data
4. #20 – groups and subtotals
5. #21 – tables
6. #22 – charts
7. #23 – sparklines

Optional Additional Tutorials are available from GCF Learnfree.org under DOING MORE WITH WORD and EXTRAS.

Licensing & Attributions

CC licensed content, Original

- Introduction to Computer Applications and Concepts. **Authored by:** Jim Shannon. **Provided by:** Extended Learning Institute (ELI), Northern Virginia Community College (NOVA). **Located at:** http://eli.nvcc.edu/. **License:** *CC BY: Attribution*

Public domain content

- Image of file folder and magnifying glass. **Authored by:** ClkrFreeVectorImages. **Located at:** https://pixabay.com/en/folders-glass-zoom-lens-search-25133/. **License:** *CC0: No Rights Reserved*

TUTORIALS: FORMULAS AND FUNCTIONS

Required Trainings

Read, view, and complete the following tutorials at GCF LearnFree.org.

Formulas and Functions

1. #13- simple formulas
2. #14- complex formulas
3. #15- relative and absolute references
4. #16- functions

Optional Additional Tutorials are available from GCF Learnfree.org under DOING MORE WITH WORD and EXTRAS.

Licensing & Attributions

CC licensed content, Original

- Introduction to tutorials. **Authored by:** Jim Shannon. **Provided by:** Extended Learning Institute (ELI), Northern Virginia Community College (NOVA). **Located at:** http://eli.nvcc.edu/. **License:** *CC BY: Attribution*

Public domain content

- Image of file folder and magnifying glass. **Authored by:** ClkrFreeVectorImages. **Located at:** https://pixabay.com/en/folders-glass-zoom-lens-search-25133/. **License:** *CC0: No Rights Reserved*

TUTORIALS: EXCEL GRAPHING TUTORIAL

https://oercommons.org/courses/resources-graphing-tutorial-graphing-with-excel

Licensing & Attributions

CC licensed content, Shared previously

- License: *Public Domain: No Known Copyright*

READING: EXCEL – CREATING A HOUSEHOLD BUDGET

https://www.oercommons.org/courses/creating-a-household-budget

Licensing & Attributions

CC licensed content, Shared previously

- License: *Public Domain: No Known Copyright*

MODULE 11: MICROSOFT ACCESS

READING: DATA AND DATABASES

https://learn.saylor.org/course/view.php?id=41&sectionid=431

Licensing & Attributions

CC licensed content, Shared previously

- License: *Public Domain: No Known Copyright*

READING: WHAT IS A DATABASE?

https://www.oercommons.org/courses/what-is-a-database

Licensing & Attributions

CC licensed content, Shared previously

- License: *Public Domain: No Known Copyright*

READING: DATABASE FUNDAMENTALS

https://www.oercommons.org/searchf.search=Database+Fundamentals+Mr.+Ford

Licensing & Attributions

CC licensed content, Shared previously

- License: *Public Domain: No Known Copyright*

READING: DATABASE MANAGEMENT SYSTEMS

https://www.oercommons.org/courses/database-management-systems

Licensing & Attributions

CC licensed content, Shared previously

- License: *Public Domain: No Known Copyright*

READING: ACCESS - SOME FINAL BITS

https://oercommons.org/courses/database-08-04-some-final-bits

Licensing & Attributions

CC licensed content, Shared previously

- License: *Public Domain: No Known Copyright*

TUTORIALS: MICROSOFT ACCESS

https://www.oercommons.org/courses/microsoft-access-tutorials

Licensing & Attributions

CC licensed content, Shared previously

- License: *Public Domain: No Known Copyright*

TUTORIALS: ACCESS BASICS

Required Trainings

Read, view, and complete the following tutorials at GCF LearnFree.org.

Access Basics

1. #3 – INTRODUCTION TO DATABASES
2. #4 – INTRODUCTION TO OBJECTS
3. #5 – GETTING STARTED WITH ACCESS
4. #6 – MANAGING DATABASES AND OBJECTS

Supplemental Trainings

You can find the following supplemental training by clicking the MS Office Tutorials link in the course menu.

- Atomic Learning—Use these videos to supplement your GCF Learn Free Tutorials as necessary.
    - To access these trainings, follow the instructions from NVCC.
- MS Office Tutorials—Watch the Access tutorials in MS Office 2013 Video Tutorials link or the MS Office 2010 Tutorials link.
- Access Help Menu – Additional learning materials are also available from the Access HELP menu. Click on the BLUE QUESTION MARK in the upper right hand corner of your application HOME page. This will take you to the Access Help page where you will find tutorials and a searchable index of topics.

Licensing & Attributions

CC licensed content, Original

- Introduction to Computer Applications and Concepts. **Authored by:** Jim Shannon. **Provided by:** Extended Learning Institute (ELI), Northern Virginia Community College (NOVA). **Located at:** http://eli.nvcc.edu/. **License:** *CC BY: Attribution*

Public domain content

- Image of file folder and magnifying glass. **Authored by:** ClkrFreeVectorImages. **Located at:** https://pixabay.com/en/folders-glass-zoom-lens-search-25133/. **License:** *CC0: No Rights Reserved*

TUTORIALS: WORKING WITH DATA

Required Trainings

Read, view, and complete the following tutorials at GCF LearnFree.org, (http://www.gcflearnfree.org/access2013).

Working with Data

1. #7 – WORKING WITH TABLES
2. #8 – WORKING WITH FORMS
3. #9 – SORTING AND FILTERING RECORDS

Supplemental Trainings

You can find the following supplemental training by clicking the MS Office Tutorials link in the course menu.

- Atomic Learning – Use these videos to supplement your GCF Learn Free Tutorials as necessary.
  - To access these trainings, follow the instructions from NVCC.
- MS Office Tutorials – Watch the Access tutorials in MS Office 2013 Video Tutorials link or the MS Office 2010 Tutorials link.
- Access Help Menu – Additional learning materials are also available from the Access HELP menu. Click on the BLUE QUESTION MARK in the upper right hand corner of your application HOME page. This will take you to the Access Help page where you will find tutorials and a searchable index of topics.

Licensing & Attributions

CC licensed content, Original

- Introduction to Computer Applications and Concepts. **Authored by:** Jim Shannon. **Provided by:** Extended Learning Institute (ELI), Northern Virginia Community College (NOVA). **Located at:** http://eli.nvcc.edu/. **License:** *CC BY: Attribution*

Public domain content

- Image of file folder and magnifying glass. **Authored by:** ClkrFreeVectorImages. **Located at:** https://pixabay.com/en/folders-glass-zoom-lens-search-25133/. **License:** *Public Domain: No Known Copyright*

TUTORIALS: DATABASE DESIGN TIPS

Required Trainings

Read, view, and complete the following tutorials at GCF LearnFree.org.

Database Design Tips

1. #15- MODIFYING TABLES
2. #16- CREATING FORMS
3. #17- FORMATTING FORMS
4. #18 – DESIGNING YOUR OWN DATABASE

Optional Additional Tutorials are available from GCF Learnfree.org under EXTRAS.

Supplemental Trainings

You can find the following supplemental training by clicking the MS Office Tutorials link in the course menu.

- Atomic Learning—Use these videos to supplement your GCF Learn Free Tutorials as necessary.
  - To access these trainings, follow the instructions from NVCC.
- MS Office Tutorials—Watch the Access tutorials in MS Office 2013 Video Tutorials link or the MS Office 2010 Tutorials link.
- Access Help Menu—Additional learning materials are also available from the Access HELP menu. Click on the BLUE QUESTION MARK in the upper right hand corner of your application HOME page. This will take you to the Access Help page where you will find tutorials and a searchable index of topics.

Licensing & Attributions

CC licensed content, Original

- Introduction to Computer Applications and Concepts. **Authored by:** Jim Shannon. **Provided by:** Extended Learning Institute (ELI), Northern Virginia Community College (NOVA). **Located at:** http://eli.nvcc.edu/. **License:** *CC BY: Attribution*

Public domain content

- Image of file folder and magnifying glass. **Authored by:** ClkrFreeVectorImages. **Located at:** https://pixabay.com/en/folders-glass-zoom-lens-search-25133/. **License:** *CC0: No Rights Reserved*

TUTORIALS: MORE ACCESS TASKS

Required Trainings

Read, view, and complete the following tutorials at GCF LearnFree.org.

More Access Tasks

1. #19 – CREATING CALCULATED FIELDS AND TOTALS ROWS
2. #20 – CREATING A PARAMETER QUERY
3. #21 – CREATING A FIND DUPLICATES QUERY

Optional Additional Tutorials are available from GCF Learnfree.org under EXTRAS.

Supplemental Trainings

You can find the following supplemental training by clicking the MS Office Tutorials link in the course menu.

- Atomic Learning—Use these videos to supplement your GCF Learn Free Tutorials as necessary.
  - To access these trainings, follow the instructions from NVCC.
- MS Office Tutorials—Watch the Access tutorials in MS Office 2013 Video Tutorials link or the MS Office 2010 Tutorials link.
- Access Help Menu—Additional learning materials are also available from the Access HELP menu. Click on the BLUE QUESTION MARK in the upper right hand corner of your application HOME page. This will take you to the Access Help page where you will find tutorials and a searchable index of topics.

Licensing & Attributions

CC licensed content, Original

- Introduction to Computer Applications and Concepts. **Authored by:** Jim Shannon. **Provided by:** Extended Learning Institute (ELI), Northern Virginia Community College (NOVA). **Located at:** http://eli.nvcc.edu/. **License:** *CC BY: Attribution*

Public domain content

- Image of file folder and magnifying glass. **Authored by:** ClkrFreeVectorImages. **Located at:** https://pixabay.com/en/folders-glass-zoom-lens-search-25133/. **License:** *CC0: No Rights Reserved*

TUTORIALS: RUNNING QUERIES AND REPORTS

Required Trainings

Read, view, and complete the following tutorials at GCF LearnFree.org.

Running Queries and Reports

1. #10- DESIGNING A SIMPLE QUERY
2. #11- DESIGNING A MULTI-TABLE QUERY
3. #12- MORE QUERY DESIGN OPTIONS
4. #13- CREATING REPORTS
5. #14- ADVANCED REPORT OPTIONS

Database Design Tips

1. #15- MODIFYING TABLES
2. #16- CREATING FORMS
3. #17- FORMATTING FORMS
4. #18 – DESIGNING YOUR OWN DATABASE

More Access Tasks

1. #19 – CREATING CALCULATED FIELDS AND TOTALS ROWS
2. #20 – CREATING A PARAMETER QUERY
3. #21 – CREATING A FIND DUPLICATES QUERY

Optional Additional Tutorials are available from GCF Learnfree.org under EXTRAS.

Supplemental Trainings

You can find the following supplemental training by clicking the MS Office Tutorials link in the course menu.

- Atomic Learning—Use these videos to supplement your GCF Learn Free Tutorials as necessary.
    - To access these trainings, follow the instructions from NVCC.
- MS Office Tutorials—Watch the Access tutorials in MS Office 2013 Video Tutorials link or the MS Office 2010 Tutorials link.
- Access Help Menu—Additional learning materials are also available from the Access HELP menu. Click on the BLUE QUESTION MARK in the upper right hand corner of your application HOME page. This will take you to the Access Help page where you will find tutorials and a searchable index of topics.

Licensing & Attributions

CC licensed content, Original

- Introduction to Computer Applications and Concepts. **Authored by:** Jim Shannon. **Provided by:** Extended Learning Institute (ELI), Northern Virginia Community College (NOVA). **Located at:** http://eli.nvcc.edu/. **License:** *CC BY: Attribution*

Public domain content

- Image of file folder and magnifying glass. **Authored by:** ClkrFreeVectorImages. **Located at:** https://pixabay.com/en/folders-glass-zoom-lens-search-25133/. **License:** *CC0: No Rights Reserved*

MODULE 12: MICROSOFT POWERPOINT

READING: BUSINESS COMMUNICATION - WRITTEN AND VERBAL PRESENTATION SKILLS

https://www.oercommons.org/courses/business-communication-written-verbal-presentation-skills

Licensing & Attributions

CC licensed content, Shared previously

- License: *Public Domain: No Known Copyright*

READING: LEARN POWERPOINT 01 NAVIGATING POWERPOINT

https://www.oercommons.org/courses/learn-powerpoint-01-navigating-powerpoint

Licensing & Attributions

CC licensed content, Shared previously

- License: *Public Domain: No Known Copyright*

READING: TITLE AND STATUS BAR FUNCTIONS IN POWERPOINT

https://www.oercommons.org/courses/at-the-bar-with-powerpoint-2013-most-excellent-powerpoint-tips

Licensing & Attributions

CC licensed content, Shared previously

- License: *Public Domain: No Known Copyright*

READING: 4 PANES OF POWERPOINT 2013 MOST EXCELLENT POWERPOINT TIPS

https://www.oercommons.org/course/the-4-panes-of-powerpoint-2013-most-excellent-powerpoint-tips

Licensing & Attributions

CC licensed content, Shared previously

- License: *Public Domain: No Known Copyright*

READING: ADD YOUTUBE VIDEOS TO YOUR PPT PRESENTATION

https://www.oercommons.org/course/add-youtube-videos-to-your-presentation-most-excellent-powerpoint-tips

Licensing & Attributions

CC licensed content, Shared previously

- License: *Public Domain: No Known Copyright*

TUTORIALS: POWERPOINT BASICS

Required Trainings

Read, view, and complete the following tutorials at GCF LearnFree.org.

PowerPoint Basics

1. #4 – GETTING TO KNOW POWERPOINT
2. #5 – CREATING AND OPENING PRESENTATIONS
3. #6 – SAVING AND SHARING
4. #7 – SLIDE BASICS
5. #8 – TEXT BASICS
6. #9 – APPLYING THEMES
7. #10- APPLYING TRANSITIONS
8. #11- MANAGING SLIDES
9. #12- PRINTING
10. #13- PRESENTING

Supplemental Trainings

You can find the following supplemental training by clicking the MS Office Tutorials link in the course menu.

- Atomic Learning—Use these videos to supplement your GCF Learn Free Tutorials as necessary.
    - To access these trainings, follow the instructions from NVCC.
- MS Office Tutorials—Watch the Access tutorials in MS Office 2013 Video Tutorials link or the MS Office 2010 Tutorials link.
- PowerPoint Help Menu—Additional learning materials are also available from the PowerPoint HELP menu. Click on the BLUE QUESTION MARK in the upper right hand corner of your application HOME page. This will take you to the PowerPoint Help page where you will find tutorials and a searchable index of topics.

Licensing & Attributions

CC licensed content, Original

- Introduction to Computer Applications and Concepts. **Authored by:** Jim Shannon. **Provided by:** Extended Learning Institute of Northern Virginia Community College. **Located at:** http://www.nvcc.edu/eli/index.html. **License:** *CC BY: Attribution*

Public domain content

- Image of file folder and magnifying glass. **Authored by:** ClkrFreeVectorImages. **Located at:** https://pixabay.com/en/folders-glass-zoom-lens-search-25133/. **License:** *CC0: No Rights Reserved*

TUTORIALS: TEXT AND OBJECTS

Required Trainings

Read, view, and complete the following tutorials at GCF LearnFree.org.

Text and Objects

1. #14- lists
2. #15- Indents and line spacing
3. #16- inserting pictures
4. #17- formatting pictures
5. #18 – shapes and word art
6. #19 – arranging objects
7. #20 – animating text and objects

More objects

1. #21 – inserting videos
2. #22 – inserting audio
3. #23 – tables
4. #24 – charts
5. #25 – smart art graphics

Optional Additional Tutorials are available from GCF Learnfree.org under EXTRAS.

Supplemental Trainings

You can find the following supplemental training by clicking the MS Office Tutorials link in the course menu.

- Atomic Learning—Use these videos to supplement your GCF Learn Free Tutorials as necessary.
    - To access these trainings, follow the instructions from NVCC.
- MS Office Tutorials—Watch the Access tutorials in MS Office 2013 Video Tutorials link or the MS Office 2010 Tutorials link.
- PowerPoint Help Menu—Additional learning materials are also available from the PowerPoint HELP menu. Click on the BLUE QUESTION MARK in the upper right hand corner of your application HOME

page. This will take you to the PowerPoint Help page where you will find tutorials and a searchable index of topics.

Licensing & Attributions

CC licensed content, Original

- Introduction to Computer Applications and Concepts. **Authored by:** Jim Shannon. **Provided by:** Extended Learning Institute (ELI), Northern Virginia Community College (NOVA). **Located at:** http://eli.nvcc.edu/. **License:** *CC BY: Attribution*

Public domain content

- Image of file folder and magnifying glass. **Authored by:** ClkrFreeVectorImages. **Located at:** https://pixabay.com/en/folders-glass-zoom-lens-search-25133/. **License:** *CC0: No Rights Reserved*

TUTORIALS: REVIEWING PRESENTATIONS AND COLLABORATION

Required Trainings

Read, view, and complete the following tutorials at GCF LearnFree.org.

Reviewing presentations and collaboration

1. #26 – CHECKING SPELLING AND GRAMMAR
2. #27 – REVIEWING PRESENTATIONS
3. #28 – FINALIZING AND PROTECTING PRESENTATIONS

Optional Additional Tutorials are available from GCF Learnfree.org under EXTRAS.

Supplemental Trainings

You can find the following supplemental training by clicking the MS Office Tutorials link in the course menu.

- Atomic Learning—Use these videos to supplement your GCF Learn Free Tutorials as necessary.
  - To access these trainings, follow the instructions from NVCC.
- MS Office Tutorials—Watch the Access tutorials in MS Office 2013 Video Tutorials link or the MS Office 2010 Tutorials link.
- PowerPoint Help Menu—Additional learning materials are also available from the PowerPoint HELP menu. Click on the BLUE QUESTION MARK in the upper right hand corner of your application HOME page. This will take you to the PowerPoint Help page where you will find tutorials and a searchable index of topics.

Licensing & Attributions

CC licensed content, Original

- Introduction to Computer Applications and Concepts. **Authored by:** Jim Shannon. **Provided by:** Extended Learning Institute (ELI), Northern Virginia Community College (NOVA). **Located at:** http://eli.nvcc.edu/. **License:** *CC BY: Attribution*

Public domain content

- Image of file folder and magnifying glass. **Authored by:** ClkrFreeVectorImages. **Located at:** https://pixabay.com/en/folders-glass-zoom-lens-search-25133/. **License:** *CC0: No Rights Reserved*

TUTORIALS: CUSTOMIZING YOUR PRESENTATION

Required Trainings

Read, view, and complete the following tutorials at GCF LearnFree.org.

Customizing Your Presentation

1. #29 – MODIFYING THEMES
2. #30 – SLIDE MASTER VIEW
3. #31 – HYPERLINKS AND ACTION BUTTONS
4. #32 – ADVANCED PRESENTATION OPTIONS

Optional Additional Tutorials are available from GCF Learnfree.org under EXTRAS.

Supplemental Trainings

You can find the following supplemental training by clicking the MS Office Tutorials link in the course menu.

- Atomic Learning—Use these videos to supplement your GCF Learn Free Tutorials as necessary.
    - To access these trainings, follow the instructions from NVCC.
- MS Office Tutorials—Watch the Access tutorials in MS Office 2013 Video Tutorials link or the MS Office 2010 Tutorials link.
- PowerPoint Help Menu—Additional learning materials are also available from the PowerPoint HELP menu. Click on the BLUE QUESTION MARK in the upper right hand corner of your application HOME page. This will take you to the PowerPoint Help page where you will find tutorials and a searchable index of topics.

Licensing & Attributions

CC licensed content, Shared previously

- Introduction to Computer Applications and Concepts. **Authored by:** Jim Shannon. **Provided by:** Extended Learning Institute (ELI), Northern Virginia Community College (NOVA). **Located at:** http://eli.nvcc.edu/. **License:** *CC BY: Attribution*

Public domain content

- Image of file folder and magnifying glass. **Authored by:** ClkrFreeVectorImages. **Located at:** https://pixabay.com/en/folders-glass-zoom-lens-search-25133/. **License:** *CC0: No Rights Reserved*

ADDITIONAL RESOURCES

SPECIAL TOPICS: DATA SECURITY AND PRIVACY: LEGAL, POLICY AND ENTERPRISE ISSUES

Click HERE to access the University of Michigan's course on Data Security and Privacy by Don Blumenthal.

TUTORIALS: WINDOWS MOVIE MAKER

Windows Essentials: This link will take you to an online site to help you use Movie Maker in Windows 7.

Windows Live Movie Maker Training: Or search for another version of the Windows Movie Maker Tutorials by using the "Applications" drop-down box HERE.

Download Windows Live Movie Maker HERE

Watch the Movie Maker Live Tutorial Here:

Watch this video online: https://youtu.be/vh82WqBQ_2c

Licensing & Attributions

CC licensed content, Original

- Introduction to Computer Applications and Concepts . **Authored by:** Melissa Stange. **Provided by:** Lord Fairfax Community College. **Located at:** http://www.lfcc.edu/. **License:** *CC BY: Attribution*

All rights reserved content

- Windows Live Movie Maker Quick Tutorial. **Provided by:** PCMech. **Located at:** https://www.youtube.com/watch?v=vh82WqBQ_2c&feature=youtu.be. **License:** *Other*. **License Terms:** Standard YouTube License

INTERACTIVE: WHO WANTS TO BE A PROTOCOLINAIRE?

Test your knowledge of Protocols and Computer Networks by playing the review game found HERE.

Licensing & Attributions

CC licensed content, Original

- Introduction to Computer Applications and Concepts, Protocol Review. **Authored by:** Melissa Stange. **Provided by:** Lord Fairfax Community College. **Located at:** http://www.lfcc.edu/,%20http://www2.rad.com/networks/2000/million/index.htm. **License:** *CC BY: Attribution*

READING: SOCIAL MEDIA

Click on the links to read and learn about various types of social media.

Licensing & Attributions

CC licensed content, Original

- Introduction to Computer Applications and Concepts Social Media links. **Authored by:** Melissa Stange. **Provided by:** Lord Fairfax Community College. **Located at:** http://www.lfcc.edu/. **License:** *CC BY: Attribution*