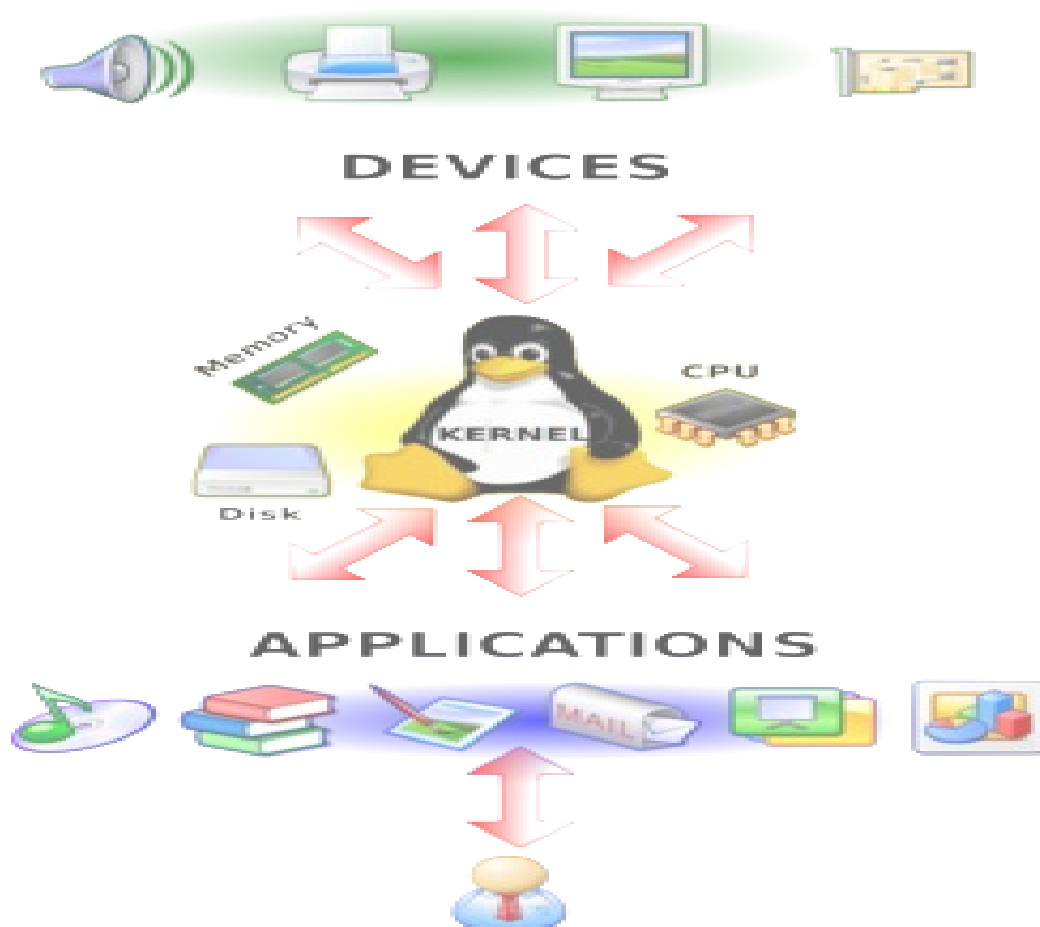


GeeksZine

Open Source is Fun



Issue : August 2010

Website : www.richnusgeeks.com

Publisher : RichNusGeeks Consulting

Table of Content

Just two words 3

Is that so? 4

Banishing a ghost 7

Surprising the master 11

Learning the hard way 12

The last rap 17

Just two words, *an editorial*

Hello everybody once again! First of all, apologies for not publishing the July edition of the zine due to some unforeseen reasons. The last two issue of the zine recorded more than 2000 views and a big thanks to all of you for the same. The zine has got one more contributor this month. Please welcome Sumit Rai on board. The most of the content in this issue was prepared by Ankur Aggarwal and kudos to him for his herculean efforts.

You may have noticed, the front page of the zine has a colorful image this month and this is not a coincidence. It is just a hint for the future directions of the zine. I'm very sure you all could guess about this indication and Ankur Agarwal has already started a pilot project this month to turn the zine into an eye candy as well. This issue of the zine contains some cool tricks with few interesting shell commands and tool to extract more fun and profit from your GNU/Linux box. There are contributed articles on GNU/Linux myths and Pitivi video editor. We have also covered under the hood mysteries of the GNU/Linux boot activities.

This zine is from geeks, for the geeks and of the geeks. So if anyone of you think of any weird section and stuff to add to zine or want to provide feedbacks then please drop us a line. We would be more than indebted to add your section and stuff to the zine and work upon your feedbacks. So all the geekheads, we hope to see you every month with more improved GeeksZine and add more fun to this world with open source software.

Keep hacking

Ankur Kumar
RichNusGeeks Consulting

geekszine@yahoo.in
geekszine@gmail.com

Is that so?, for the first timers to floss

There is always a fear in change. Since most of us start as Windows users so people are afraid of moving to GNU/Linux from Windows. Believe me, I have seen this is at my college at a very large scale. People wanna play a safe game :-) well I think that this fear came in their minds because of myths/rumors prevailing in the real world. I have tried to answer those myths in my way. Hope you will get your answers from it.

Myth 1 - Free means free of cost like free beer

Reality - Free word has many meanings and in the world of GNU/Linux its freedom not free of cost. Freedom to modify your software according to your needs by changing the available source code, freedom to distribute it among others. There is no mention anywhere that “Free” software is free of cost. You can charge the software you developed but the condition is that you have to make the source code available. In the world of GNU/Linux knowledge comes first and then money like “Free Speech Not Free Beer”. You can even make a good living by providing the support for the free softwares.

Myth 2 - GNU/Linux is command line environment, bad at graphical interface

Reality - You will start working and enjoying the command line as soon as you start working on the GNU/Linux. Those who thinks that it is bad at graphical interface then try out Compiz guys. You will see amazing graphics work. A screenshot is shown below to show the GNU/Linux graphical capabilities :



Myth 3 - There are only few GNU/Linux users around

Reality - GNU/Linux community is growing very fast. Ubuntu distribution now holds above 8 million users and this is the statistics only for one distribution. There are many distributions of GNU/Linux like RedHat, Gentoo, OpenSUSE, Fedora etc. and hundreds of more and GNU/Linux community is very active on web. Check this by pasting any forum on the web and you will be surprised to see the number of people want to help you.

Myth 4 - I am a Microsoft Windows user, what about the softwares and all I am using on Microsoft Windows?

Reality - I guarantee you people that there exist all the alternative of the Windows software in GNU/Linux :

Microsoft Office - Open Office

Matlab - GNU Octave, Scilab

PhotoShop - GIMP

Window Media Player – Vlc, Tottem Movie player

GCC compiler is one of the most powerful compilers in the world that supports a plethora of programming languages,

and the list goes on. You will find minimum one software alternative in GNU/Linux, however more than one are highly possible. Moreover, popular softwares like skype, vlc, gns3, dropbox, java-sdk also run on the GNU/Linux.

Myth 5 - It is not worth bothering to learn GNU/Linux because most companies use Microsoft Windows and thus a knowledge of Windows is desired for most jobs

Reality - I think it is the worst myth. People also use GNU/Linux at a large scale and 60% of the companies in the world use LAMP (GNU/Linux apache mysql php/perl/python) servers for their operations. Movies like Titanic, Avatar and thousands others were processed on these servers only. What stops the companies from adopting the GNU/Linux in their work culture is lack of GNU/Linux Professionals. People with GNU/Linux skills

are typically get paid substantially more than people with Windows skills. Developers always prefer GNU/Linux. So guys don't ever think that you can't earn through GNU/Linux.

Ankur Aggarwal

coolankur2006@gmail.com

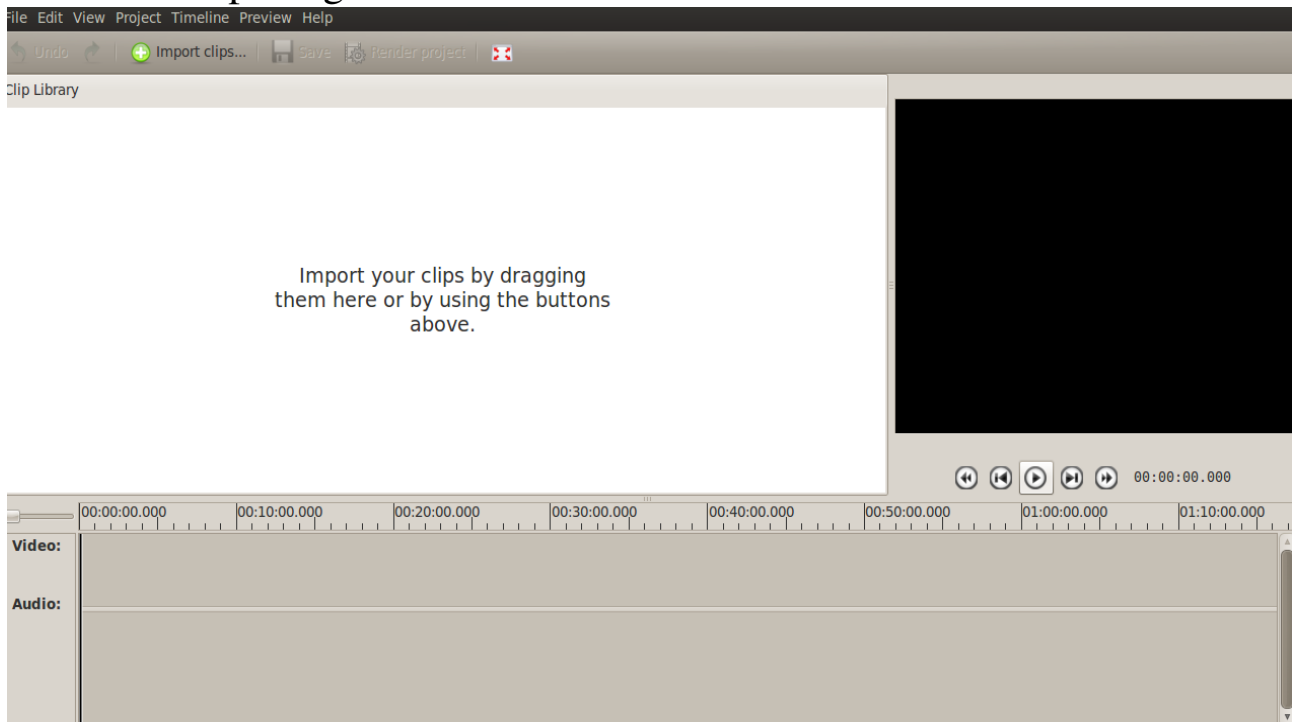
www.twitter.com/ankurtwi

Banishing a ghost, *transition to floss*

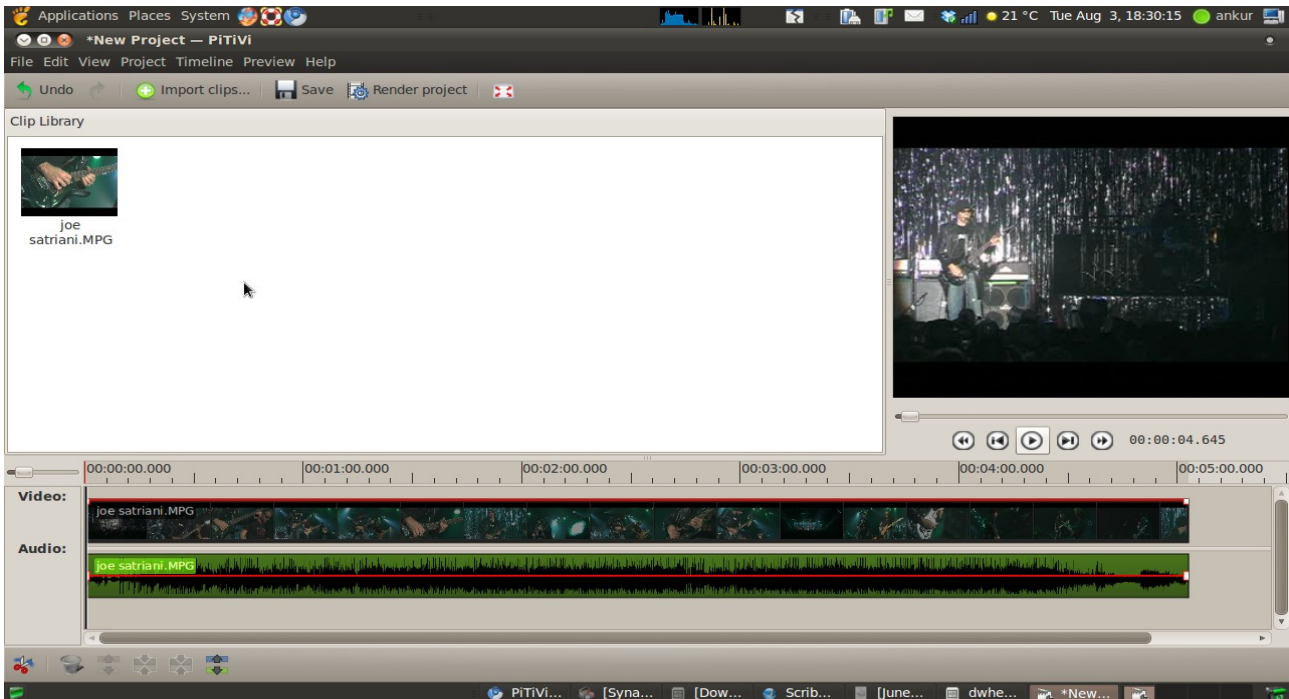
Everyone likes to hangout with videos. Whether it is a movie, tech documentary or just a fun videos, we all love to watch those. In this article I am gonna talk about video editing, one of my timepasses. There are many open source video editors are available like Open Shot video Editor, Avidemux, Pitivi. I am gonna talk About Pitivi video Editor. It comes pre-installed with ubuntu 10.04 and you can also manually install it by typing **sudo apt-get install pitivi** for ubuntu.

The latest version of Pitivi Video Editor was 0.13.4 at the time of writing. Its programmed in Python (Gtk+), you could refer to www.pitivi.org for more details about it. Pitivi is a open source and featureful video editor licensed under LGPL. It is Supported by the gstreamer multimedia framework so it can handle the most of the available formats. You can also convert your Videos to HD using this and also can do the conversion from one format to another. We will come to that part too so let's get started with it.

You can open Pitivi video either either by typing **pitivi** in your terminal or by going to Applications -> Sound and video -> Pitivi in Ubuntu. Well I will suggest you guys to try out side by side :-). You should see a screen like this on opening it :



Next step is to import a clip. Click the import button below your menu bar. A selection window will pop up as soon as you click on it. Select the video you want to edit. Then drag your video to the blank screen you will see in the right part of it and to the video & audio part too. You will see something like this there :



Video part consist of only the video part without audio and below video is the audio part (all your audio comes from there only). Now click on the play button behind the screen on the right side of the application and you will notice that seek bar movies along the time. At the bottom of the application window lies the most important tools which we are gonna use for video editing:

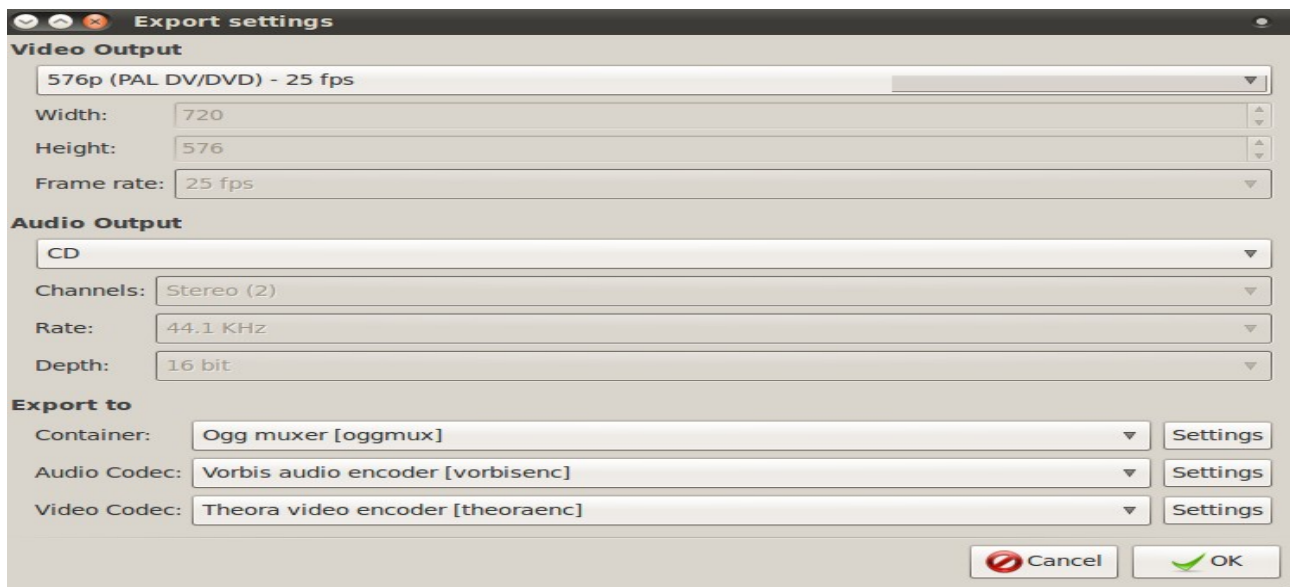


A scissor icon is used to cut. When you click on it the cut operation starts. Than make two clicks on the video part, first - start from where you wanna cut and second at the end of cut part.

Then comes the deleted section. Suppose you wanna delete the cut part, all you have to do is to pick up the cut part and put it into this section.

Then comes the groups and ungroup clips icons.

Groups clips will group the audio and video part together. When you move the video part audio part will move together. Ungroup clips will Separate them. Now all will move individually. This is used when you want to change the background sound of the video. You just ungroup it and remove the audio. Then Import an Audio clip by dragging it to audio part and then group them. Isn't it cool?



You can watch what will happen to your clipping by clicking the play button as discussed above. Now it is the turn to make a final product of your editing. You will see a button render project near the import button on the tool bar. It will ask you to choose file if you press it. It shows you selection window as soon as you click it. Give a name to your video and location to it and press Ok. Then click on modify button and you will see something like shown above in the screenshots. Give the video output to it. You will have a number of options over there like 720p, 1080p etc. It lets you too make HD video too. Then choose your audio output and then move to the format.

As soon as you scroll down the container part you will be shocked to see a large number of formats it supported. Select the appropriate audio and video codecs and press render. It will take some time and by the end of it you will get your edited video.

You can also convert your low quality video into HD by just importing the video and render the project into HD video output. This article should

get started new GNU/Linux users who wanna try their hands on video editing. Pitivi is good for professionals as well.

Ankur Aggarwal

coolankur2006@gmail.com

www.twitter.com/ankurtwi

Surprising the master, *cool tricks*

- You know that we can create a separate swap partition but you can create a swap file on any existing partition by using this method and then add it to swap file.

```
# dd if=/dev/zero of=/path/to/swapfile bs=1048576  
count=200
```

I just created a 200 MB file through the above shown command that is empty, /dev/zero is a virtual device. The command creates the file and writes zeros over it (in binary). Now it is the time to format this file as a swap (don't worry! it won't format your partition just the file) through the command **# mkswap /path/to/swapfile**. Now activate the swapfile using **# swapon /path/to/swapfile**. You can check the status using either **# swapon -s** or **# free -m** (if already have let's say 1024 Mb of swap space, then 200 MBs will add to it). If you want to make it permanent add the following entry in **/etc/fstab** file **/path/to/swapfile swap swap defaults 0 0**.

- Let's say there is a command that you have to type over and over again e.g. **ls -lh**, **ps aux | less** etc. You can save time by defining an alias **alias aliasname='command'**. Now you can use **aliasname** to execute the command. Add these entries in your **~/.bashrc** file to make the changes permanent.
- You can check power status of your computer battery, whether ac is plugged in or not etc. through **acpi** command. You could install it through command **sudo apt-get install acpi** on Ubuntu systems. The usage of the command is like **acpi -t** shows the temperature of your system, **acpi -b** shows the battery status etc. **acpi --help** will display help. This command does not have many options so it should not take much time to explore it.

Learning the hard way, *cool floss software tools*

We all love to work on GNU/Linux but have you ever thought how it boots up? Well we are gonna look at it deeply in this article. Basic input/output system (BIOS) does power on self test (POST) when you turn on your computer. After POST is complete, BIOS looks for bootable devices attached to your system. A boot device can be your hard disk, cd-rom or a network device. If your primary boot device is hard drive, then BIOS loads and executes boot code stored in MBR (Master Boot Record) of your hard drive. MBR is the first sector of your hard disk and its size is 512 byte. It stores your partition table and your boot loader. The first 446 bytes contain boot loader code and the next 64 bytes contain your partition table. The last two bytes stores the magic number. Magic number serves as a validation check for MBR.

You can backup your MBR using the command **# dd if=/dev/sda of=mbr bs=512 count=1** (use /dev/hda instead of /dev/sda if you have a IDE drive). Its recommended to save the backup file on external storage. The dd command reads the first 512 byte from /dev/sda and write them to mbr file. You can use any filename you wish. Use bs=446, if you only want to backup boot loader code not partition table and magic number. You should also backup your partition table using

```
# fdisk -l > partition_table.txt .
```

If your MBR gets corrupted you can boot from live cd and restore it using **# dd if=/path/to/mbr of=/dev/sda bs=512 count=1**, where mbr is the file you backed up previously. Once the first-stage boot loader stored in MBR is executed, the next step is to load second-stage boot loader. The first-stage boot loader now looks in the partition table for any active partition. When an active partition is found, the boot record of that partition is executed. Now the stage 2 boot loader is loaded in the memory. The stage 2 boot loader displays a menu of possible boot options. Now it loads the kernel and initial RAM disk. Initial RAM disk act as a temporary root filesystem. Kernel and initial RAM disk are located in /boot directory. Kernel usually goes by the name vmlinuz or vmlinz with kernel version number as suffix.

Lilo (GNU/Linux Loader) and GNU GRUB (GNU Grand Unified

Bootloader) are the two most popular boot loaders. We will only discuss GRUB here. The latest version of GRUB is referred to as GRUB 2 (version 1.97 or later). GRUB legacy (version 0.9x) is still widely used. To find which version you are using type the command **# grub-install -v**.

When GRUB is loaded, it provides menu of different boot options. If you don't see any menu press escape or shift key. The default GRUB configuration file is `/boot/grub/grub.conf` for GRUB Legacy, you can edit this file to change default settings or add custom boot entries. Similarly GRUB 2 has configuration file `/boot/grub/grub.cfg`, but you are not to edit this file because `grub.cfg` is overwritten every time kernel is added or removed or when user runs **update-grub** command. So all of your custom configuration will be lost. Instead you can safely edit `/etc/grub.d/40_custom` file to add custom boot entries and you can edit `/etc/default/grub` to change menu display settings.

In GRUB 2, everytime you change a configuration file you must run `update-grub` command to apply the changes. Now lets take a look at sample boot entry in `grub.conf` file :

```
title Fedora #Name of OS (will appear in boot menu)
root (hdX,Y) # Root partition or boot partition if you have one
kernel /boot/vmlinuz ro root=/dev/sdZY #Location of Kernel
initrd /boot/initrd.img #Loads initial RAM disk
```

The text written after # is considered a comment. The entry `root (hdX,Y)` denotes the partition where kernel and initial RAM disk files are stored. If you dont have a separate boot partition, it will just refer to your root partition. X in `root(hdX,Y)` is your hard drive and Y is partition on that drive. GRUB Legacy counts hard drive and partitions from 0, so if I wanted to refer to third partition (`/dev/sda3`) on my first hard drive I will use `root (0, 2)`. The line starting with `kernel` loads the kernel, `ro` just means initially mount root as read only. This is necessary to run filesystem checks on your root partition during startup. The entry `root=/dev/sdZY` tells the kernel your root partition, if your root partiton is `/dev/sda3` use `root=/dev/sda3`.

The next line starting with `initrd` just loads the initial RAM disk. Look up in `/boot` directory to find name of kernel and initial RAM disk. If you

are not sure which is your root partition, the three commands **fdisk -l**, **blkid** and **df -h** may be useful. Type these as root user. GRUB 2 count hard drives from 0 but partitions from 1, so /dev/sda4 will be root(0, 4) instead of root(hd0, 3) as in GRUB Legacy. A sample boot entry for GRUB 2 will look like :

```
# Custom boot options, you can add them to /etc/grub.d/40_custom
# file
#Ubuntu is installed on /dev/sda4
menuentry "Ubunutu"{
set root=(hd0,4)
GNU/Linux /boot/vmlinuz-2.6.32-21-generic ro root=/dev/sda4
initrd /boot/initrd.img-2.6.32-21-generic
}
```

Now you can run update-grub command to apply the changes. You can edit the boot options at startup by pressing e when menu entries are displayed. GRUB also provides command line interface. If your grub.conf or grub.cfg file is corrupted for some reason, grub will display a prompt like **grub>** or **grub-rescue>**. You can also press c to enter command line when boot menu is displayed. GRUB command line also has tab auto-completion just like bash. At prompt you can press tab to display a list of available commands. Here is how you boot manually from command line (here root partition is /dev/sda4, your root partition may be different) :

grub> root(hd0,3) or **grub> set root(hd0,4)** for GRUB2

You can press tab twice after typing "root(hd0)" to display a list of available partitions. In GRUB 2 you simply type "ls" command to display a list of drives and partitions known to grub. You can also use ls to display contents of any folder eg. **grub>ls(hd0,4)/boot/grub**. It will display contents of /boot/grub folder in /dev/sda4.

```
grub> kernel /boot/vmlinuz ro root=/dev/sda4 or for GRUB 2,
grub> linux /boot/vmlinuz ro root=/dev/sda4
grub> initrd /boot/initrd.img
```

grub> boot

If you have forgotten your password and want to reset it, in the second step you can type **grub> kernel /boot/vmlinuz ro root=/dev/sda4 init=/bin/bash.**

It will present you with a bash prompt and you can simply reset your password using **passwd** command. GRUB prompt also provides **cat** command so that you can look at text files. Also there is **grub> set pager 1** (equivalent to “l less” in bash the output will only scroll down after you hit enter). If your grub is wiped out you can reinstall it by booting into live cd and type the following commands as root :

```
# mount /dev/sdaX /mnt
```

If you have a separate boot partition mount it using (if you don't then skip this step) **# mount /dev/sdaY /mnt/boot.** Now reinstall grub using **# grub-install --root-directory=/mnt /dev/sda.**

Now after boot loader loads the kernel, kernel detects your hardware and initializes all the device drivers. After that the kernel runs the first user-level program **/sbin/init**. Init spawns all other processes and has its PID as 1. You can run the command **ps tree** to display all the processes in a tree like structure. Init looks up for its configuration file **/etc/inittab** and boots you into default runlevel. The following entry in **inittab** defines your default runlevel :

```
id:5:initdefault: (Here the default runlevel is 5).
```

Ubuntu and fedora use scripts stored in **/etc/init/** directory to configure init. Runlevel 0 means shutdown, runlevel 6 is for reboot and 1 is single user mode. The rest of runlevels vary from distribution to distribution. Most commonly runlevel 5 is used for graphical desktop. To find out your current level you can use the commands **# runlevel** or **# who -r**. Let's say you are now in runlevel 5, and you want to switch to runlevel 3, just type **# init 3**. Similarly you can shutdown your system using **init 0** or reboot using **init 6**. While entering a runlevel init runs or kills services depending upon configuration.

To find which services are run by init while entering a particular

runlevel look for the corresponding `/etc/rc?.d` or `/etc/rc.d/rc?.d` directory eg. for runlevel 3 look into `/etc/rc3.d/` or `/etc/rc.d/rc3.d` directory. The directory will contain scripts to start or stop a particular service. You will see files with the names of type `KnnService-name` or `SnnService-name` where `nn` is a number from 00 to 99 eg. `S10network`, `K36mysqld` etc. The scripts are executed in numeric order i.e. scripts with lower value of `nn` are executed before scripts with higher value of `nn`. The scripts with names starting with `K` are executed first to kill any existing service and the files starting with `S` are used to start services. If you want a particular service to start executing at a particular runlevel just mark the script in corresponding `rc?.d` unexecutable eg. **`# chmod -x /etc/rc5.d/S10network`**.

Now the networking service won't start at runlevel 5. Scripts in `rc?.d` directories are nothing but symbolic links to scripts stored in `/etc/init.d` or `/etc/rc.d/init.d` directories depending upon your distribution. So you can manually start a service by using scripts in `/etc/init.d` directory eg.

`# /etc/init.d/network restart` will restart the networking service. So keep digging deep and exploring GNU/Linux.

Sumit Rai

sumit.tux@gmail.com

The last rap, *an epilogue*

All the thoughts and the information presented in this zine are based upon the various freely and openly available resources on the internet and the personal experiences. So we don't guarantee the fitment of the opinions and the software mentioned for some particular purposes. Please try the information provided in the zine on your risk only and we are not responsible for any damage and loss caused by that.

We are putting this work in Public Domain and you are free to use and distribute the information anyway you like, with or without any attribution. If you like the work then we encourage you to share it more and more in various forms.

This entire document was produced with the FLOSS using OpenOffice.org 3.1 on Ubuntu 9.10 64-bit Desktop edition.