

# GeeksZine

*Free Open Source is Fun*

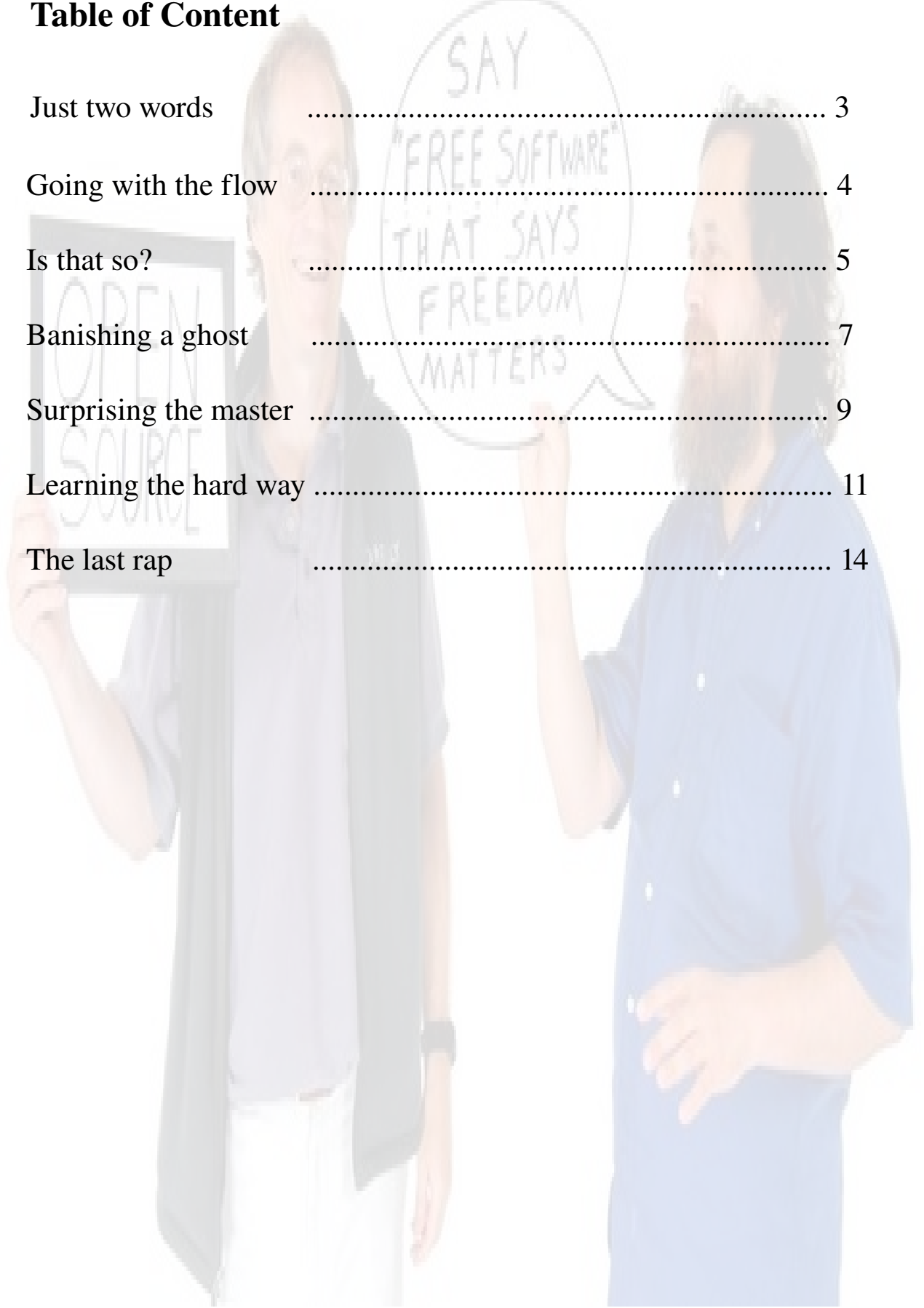


Issue : December 2010

Website : [www.richnusgeeks.com](http://www.richnusgeeks.com)

Publisher : RichNusGeeks Consulting

## Table of Content



Just two words .....	3
Going with the flow .....	4
Is that so? .....	5
Banishing a ghost .....	7
Surprising the master .....	9
Learning the hard way .....	11
The last rap .....	14

## **Just two words, *an editorial***

Hi FLOSS geeks and all, we have reached to the end of the year 2010. So this is the last issue of the year 2010 and 6<sup>th</sup> in total to get published. It's a pleasure to present the 6<sup>th</sup> issue of the GeeksZine and hope that we'll keep on improving and delivering the zine for many more years to come. We started our punch line with “Open Source is Fun” and now it has become “Free Open Source is Fun”. The addition of “Free” in the software world created an underground revolution years ago and now that revolution has finally emerged on the mainstream software scenario in 360 degrees.

Last few years have been remarkably historic for the acceptance of the free and open source software in the mainstream where everyone had to agree that you love or hate but the FLOSS is the way to go. We are sure that the growth of the FLOSS is gonna be multi fold in the coming years. The cover image of the zine captures the spirit of the freedom in the software world and who could indicate it better than Richard Stallman himself? This issue of the zine consists of Turning your old machines into development workhorse using PuppyLinux, Text processing with power tool Sed and a writeup on the current release of PyMakeMeLazyNDumb. Also there are regular News bytes and Tips n tricks sections for you.

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 free open source software.

Keep hacking

Ankur Kumar

RichNusGeeks Consulting

geekszine@yahoo.in

geekszine@gmail.com

## Going with the flow, *news bytes*

- Seattle-based Attachmate Corp. is buying Novell for \$2.2 billion the companies announced on November 22. At the same time, Novell announced the concurrent sale of certain intellectual property assets to CPTN Holdings LLC, a consortium of technology companies organized by Microsoft Corporation, for \$450 million in cash ( complete news at <http://bit.ly/c81zCA> ).
- In Asia, Android has just surpassed surpassed long-time leader Symbian for smartphone supremacy in Q3, according to market research firm GfK Asia. Interesting to note is the rise in overall smartphone sales, up from 1.27 million to over 4.7 million from the same time last year ( complete news at <http://bit.ly/gais2P> ).
- Wikipedia Founder, Jimmy Wales, in this annual fund-raising campaign, is asking for users for donations. In his appeal Wales notes that Wikipedia has ten years worth of content and 400 million Wikipedia users and that users through their donations can help keep the site free of ads and free to use ( complete news at <http://bit.ly/h7wKfv> ).
- Fedora 14 has gone gold. According to Adam Williamson, Red Hat Senior Quality Assurance Engineer, the Fedora 14 Final Release Go/No-Go Meeting resulted in the unanimous decision that RC1 should be declared Gold ( complete news at <http://bit.ly/9IWKzW> ).
- On its wiki, the Symbian Foundation has announced that its websites will shut down on December 17, making available the source code, kits, bug database and reference documentation available for developers upon request ( complete news at <http://bit.ly/gvKoCk> ).

## Is that so?, *for the first timers to floss*

We all computer users browse through a lot of data and store it on everyday basis. But this stored data becomes scattered over the time if not properly managed from time to time and it becomes difficult to track the stored items. The simplest solution to manage the data we store on our computers everyday is to store the data items in their proper *directories and subdirectories*. For example – you could store all your python sources in a subdirectory called Works and tutorials in a subdirectory Tutorials contained in a Python directory.

A proper *directory hierarchy* itself is the *simplest tracking system* in this world. Once you store the relevant pieces of your data then you could pinpoint the type of data stored from the name of the directories and subdirectories itself. You can create directories manually in case there are few directories and subdirectories but what about when you have to create an hierarchy of 10 directories and 2 subdirectories? Don't worry! There is a simple but effective python application known as *pycreatemyspace.py* to automatically accomplish the work just with the execution of a simple command.

Pycreatemyspace.py is a component of the PyMakeMeLazyNDumb automation tool described in the November 2010 issue of GeeksZine. It can take the directories and subdirectories arguments from command line or a configuration file. You could also specify the parent directory where your directory hierarchy is created. This application also takes care of not disturbing the existing directories or subdirectories and it only creates the non-existing directories and subdirectories even if provided with the list of the existing ones. The pycreatemyspace.py logs all the errors and warnings in a log file named as *activity.log* and it takes all the arguments from *config.conf* existing in the current location from where it is invoked, like all the components of PyMakeMeLazyNDumb.

Now we see the usages of the pycreatemyspace.py through some examples. First of all, like all other standard command line utilities you can see the usages of it through the following command :

```
python pycreatemyspace.py -h
```

If you want to create subdirectories named Personal, Development, Official in your home directory then the command to execute is :

```
python pycreatemyspace.py -d ~ -p 'Personal Development Official'
```

Now, if you want to create directories C, C++, Python, Php with every directory having subdirectories Works and Tutorials in the Development subdirectory of your home directory then the command to type is :

```
python pycreatemyspace.py -d ~/Development -p 'C C++ Python Php' \  
-s 'Works Tutorials'
```

If you are creating an hierarchy of many directories and subdirectories then typing at the command line is cumbersome. In that case you can specify all the directories and subdirectories in a configuration file config.conf created in the current working directory as shown below :

```
# example config file for PyMakeMeLazyNDumb  
[pycreatemyspace]
```

```
Directories = list of directories
```

```
Subdirectories = list of subdirectories
```

now finally, type the following command to create the directory hierarchy taken from the config.conf :

```
python pycreatemyspace.py -c
```

You can see yourself that how useful pycreatemyspace.py could be to create good directory hierarchies so that you could store and track your data properly to save your time and efforts in the hours of need. You can download the latest version of PyMakeMeLazyNDumb from the following sites :

<http://pymakemelazyndu.sourceforge.net/>

<http://code.google.com/p/pymakemelazyndumb/>

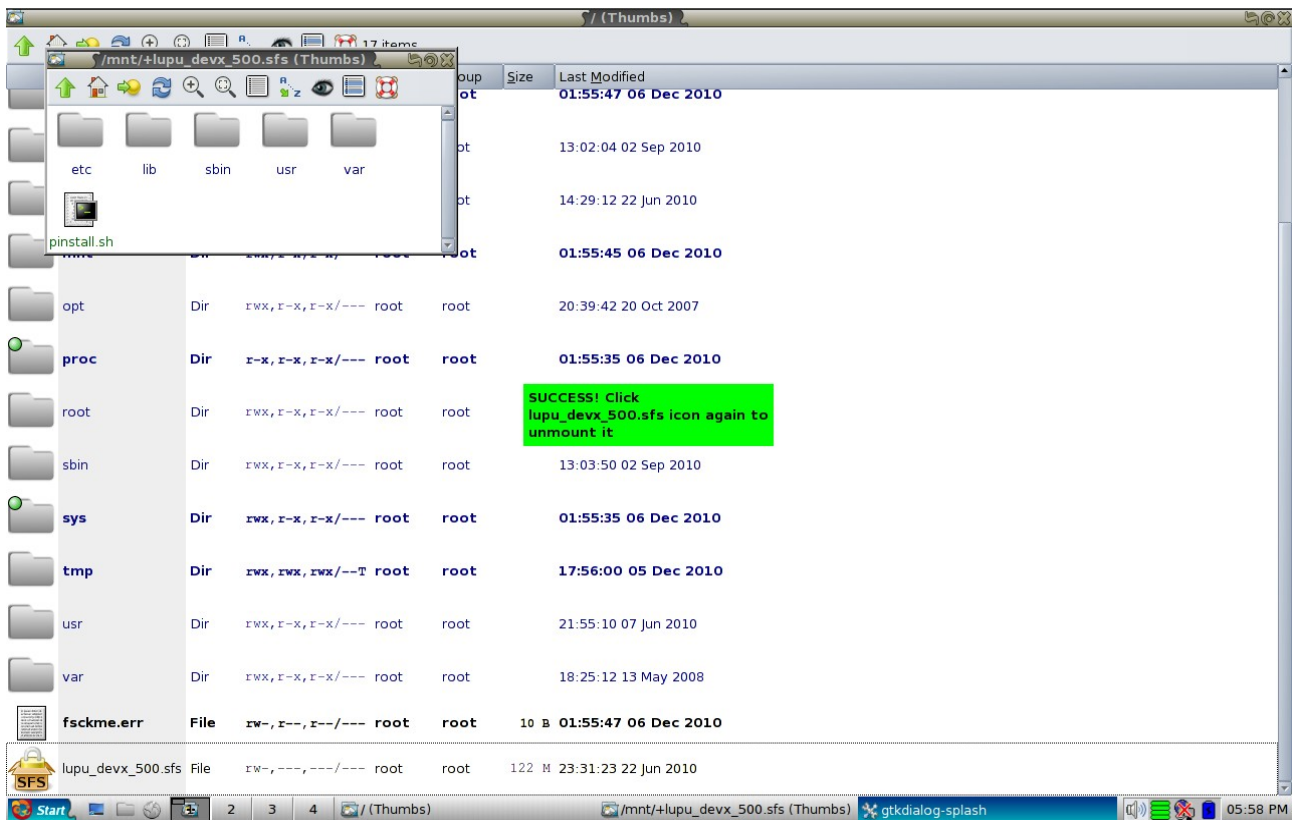
## Banishing a ghost, *transition to floss*

The different distributions of GNU/Linux operating system package a plethora of packages targeting from a first time user to the hardcore software developers. But most of the times people turn to the GNU/Linux because they have heard its reputation to run on any kind of hardware, either legacy or the modern ones. This ability of GNU/Linux to utilize any kind of hardware is of a special interest to the developers who want to get their hands dirty in various technologies quickly without the need of complicated and time consuming setup process on expensive or modern hardware.

We have explored a very light and very feature rich GNU/Linux distribution known as *PuppyLinux* in the May 2010 issue of the zine (you could get it from the website of GeeksZine). PuppyLinux not only provides a fulfilling computing experience to the new as well as seasoned users of FLOSS but is also a very excellent software development platform that fully cooperates with legacy to the modern hardware. Now we shall see how easily we can turn this cute looking PuppyLinux into a powerful *software development workhorse*?

First of all, we need to download a *sfs archive* of all the tools like gcc, g++, gdb, binutils, vala compiler etc. that are required to develop with the programming languages like C, C++, Vala/Genie etc. and to build other components like Kernel, Interpreters, Source tarballs etc. To do this, go to [www.puppylinux.com](http://www.puppylinux.com) and download the devx package that matches to your PuppyLinux version. For example – if you are running PuppyLinux v5.1.1 then you have to download devx\_511.sfs package. Once you download the appropriate devx package then go to the folder where that was saved and click on it. There should be a message like shown in the screenshot below. It means that the content of the devx package was mounted to a folder in */mnt* directory. Now go to */mnt* subdirectory and you should see a devx folder over there. We are very near to turn our PuppyLinux system into a development platform. Now open a rxvt terminal in the devx folder and type the following command into that :

```
cp -a --remove-destination ./*/ && sync
```

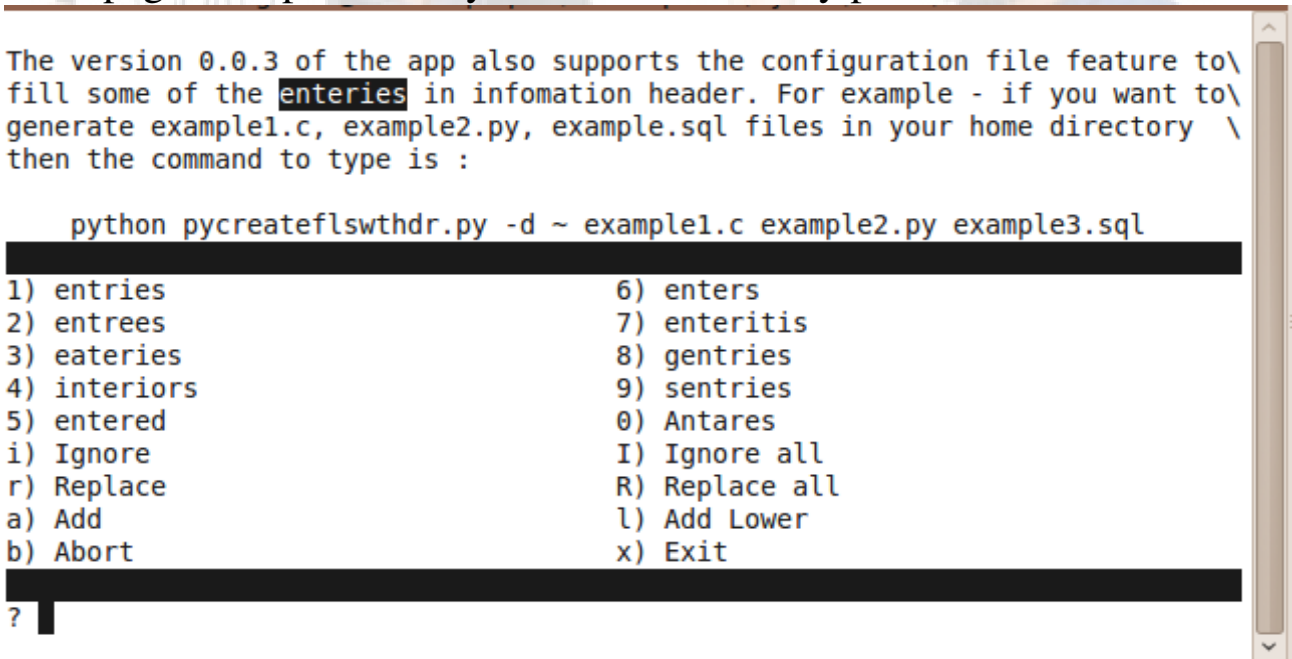


The command may take some time to complete depending upon your hardware configuration. Once the command returns then try to type `gcc -v`, `g++ -v`, `valac --version` and other binutils commands etc. in the console and you should see all the versions info there. It means now you're ready with your development setup. Now again click on your devx archive to unmount it and get ready to unleash the great development powers of your cute PuppyLinux :-)

## Surprising the master, *cool tricks*

- We all need to check our documents for the proper spellings etc. There is an GNU utility available known as *aspell* to accomplish this spell checking work efficiently and fast. You could install this utility on Ubuntu systems by issuing command `sudo apt-get install aspell` in a text console. On other platforms, you could build it from source code by downloading it from <http://aspell.net/> and following the instructions provided in README file in the source tarball.

The basic usage of *aspell* is pretty simple, type `aspell -c filename` to interactively check/correct the spellings in your file. The screenshot of this operation is shown below. You could also produce a list of suspicious spellings from your file through the command `cat filename | aspell list`. Explore more about *aspell* through its man page and spell check your files without any pain.



```
The version 0.0.3 of the app also supports the configuration file feature to\
fill some of the enteries in infomation header. For example - if you want to\
generate example1.c, example2.py, example.sql files in your home directory \
then the command to type is :
```

```
python pycreateflswthdr.py -d ~ example1.c example2.py example3.sql
```

1) entries	6) enters
2) entrees	7) enteritis
3) eateries	8) gentries
4) interiors	9) sentries
5) entered	0) Antares
i) Ignore	I) Ignore all
r) Replace	R) Replace all
a) Add	l) Add Lower
b) Abort	x) Exit

```
? |
```

- Htop is an interactive version of process viewer top. It's an ncurses based utility that enables you to move in both the dimensions in the process lists to select processes and see their command line. You could also setup the whole screen of htop by pressing F2 and selecting the proper options.

You can also change the nice values of the processes, sort those on basis of various criterion, kill process, arrange those in tree structure

etc. by using the function keys shown at the bottom of the screen. You can install htop on Ubuntu systems by running `sudo apt-get install htop` in a text console. Please browse the home page of htop at <http://htop.sourceforge.net/> for other distributions and to build it from the source tarball. A screenshot of htop running is shown below :

```

1  [|||||||||||||||||||||||||||||85.7%]   Tasks: 226 total, 3 running
2  [|||||                               12.7%]   Load average: 1.13 0.99 0.98
Mem[|||||||||||||||||||||2986/3964MB]   Uptime: 05:32:46
Swp[                                     0/7632MB]
Battery: 99.1% (Running on A/C)

  PID USER   PRI  NI  VIRT   RES   SHR  S  CPU% MEM%   TIME+  Command
1808 richnug  20   0 1723M  886M 50932 R  93.0 22.4 3h33:25 /usr/lib/firefo
4922 richnug  20   0 21436  3256  1028 R   1.0  0.1 0:12.05 htop
1362 root     20   0  857M  304M 15268 R   0.0  7.7 4:46.00 /usr/bin/X :0 -
2269 richnug  20   0 1723M  886M 50932 S   0.0 22.4 1:39.90 /usr/lib/firefo
2057 richnug  20   0  578M  105M 36432 S   0.0  2.7 0:19.64 /usr/bin/google
1778 richnug  20   0  441M 26824 16228 S   0.0  0.7 0:22.66 gnome-panel
5151 richnug  20   0  188M 12752  9508 S   0.0  0.3 0:00.19 gnome-screensho
1777 richnug  20   0  254M 35696  7680 S   0.0  0.9 0:36.63 /usr/bin/compiz
1813 richnug  20   0  303M 29692 15104 S   0.0  0.7 0:04.99 /usr/bin/python
1695 richnug  20   0  202M 13472  9436 S   0.0  0.3 0:01.83 /usr/lib/notify

F1Help F2Setup F3Search F4Invert F5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Qu

```

```

richnusgeeks@ankur-laptop:~/Development/Python/Works/Utilities$ lscpu
Architecture:          x86_64
CPU(s):                2
Thread(s) per core:   1
Core(s) per socket:   2
CPU socket(s):        1
NUMA node(s):         1
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 23
Stepping:              6
CPU MHz:               2101.000
Virtualization:       VT-x
L1d cache:             32K
L1i cache:             32K
L2 cache:              3072K
richnusgeeks@ankur-laptop:~/Development/Python/Works/Utilities$ █

```

- There comes a command with Ubuntu systems to dump the information about your CPU(s). This command is `lscpu` and it dumps the CPU info when executed in a text console. A screenshot of this command is shown above.

## Learning the hard way, *cool floss software tools*

Text related operations like searching, replacing, deleting, appending etc. for different words or patterns is a very general activities in the daily life of computer users. People use various editors, tools, programming libraries to complete these text processing tasks. All the operating systems derived from Unix culture comes packed with some of the traditional and time tested utilities to accomplish various kind of text processing goals catering from novice users to professional programmers. These text processing utilities are powerful enough to accomplish text processing in few lines with little efforts that otherwise would take a lot of efforts. We explored one of these classic utilities *grep* in the October 2010 issue of the zine. Now we'll explore another utility known as *sed* that is the grand daddy of *grep*.

*Sed* stands for stream editor and basically its features were derived from a line editor known as *ed*. In fact, the origin of *grep* is a *sed* command *g/re/p* that prints all the lines with instances of regular expression *re*. Its known as stream editor as it works upon text stream one line at a time according to the commands given to it. There are different flavors of *sed* but we'll refer to the most popular version of it that is GNU *sed*. The basic usage of *sed* is :

```
sed -e 'commands separated by ;' sourcetextfile
```

By default, *sed* writes the transformed text stream to the stdout (text screen by default) but there is a *-i* option that transforms the source file inplace. The output of the *sed* is usually redirected to a file so please take care to backup the source text file first if the output is redirected to the source file. If you are using multiple *sed* commands to process the text stream then it is recommended to put all those in a text file on different lines and use *-f* option to feed the commands from that file as shown below :

```
sed -f sedcommandsfile sourcetextfile > outputfile
```

By default, *sed* emits both input and processed lines and it may be

confusing and problematic in case we are looking for only the processed lines. So sed provides *-n* option for suppressing the input lines. Now we'll move to some of the most used commands of sed through examples. If we want to print all the methods in a python file through sed then the command is :

```
sed -ne '/def/p' source.py
```

here the regular expression to be matched is put in // pair and *p* is the command to print the matching lines. If you also want to print the line numbers for the python methods then the sed command to print current line is = the command to type is :

```
sed -ne '/def/p; /def/=' source.py
```

The sed command to substitute a pattern in a line is *s* and it only substitutes the first matching in the line if not provided with *g* address. The command to replace all instances of word perl by python in a file is :

```
sed -ne 's/perl/python/g' sourcetextfile
```

here the pattern to be replaced is put first and the replacement is put second separated by / symbol. On the same lines, If you want to delete the first instances of perl from a file then the command to type is :

```
sed -ne 's/perl//' sourcetextfile
```

The sed command to delete the matching lines is *d*. You can also use a range of lines with *d* (infact, with *p* too) to indicate the lines you want to remove in the output. The command to print only lines after 500 with line numbers is :

```
sed -ne '1, 500 d; p; =' sourcetextfile
```

There is a sed command *G* if you want to space your input text. If you

want to double space your file then the command to type is :

```
sed -ne 'G; G; p' sourcetextfile
```

the sed command to quit the processing is *q* if it is provided with a number before it then sed quits after processing those number of lines. The command to print first 10 lines of a file (like *head* command) is :

```
sed -e '10q' sourcetextfile
```

If you want to use *extended regular expressions* with sed then the option is *-r*. The examples presented in the article were some of the basic and common usages of the sed otherwise it is a very powerful utility having a lot of other features to accomplish very complicated text processing task. This article should be enough to get you started with sed and you could explore more about it through man pages and other freely available learning material.

## **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 fitments of the opinions and the softwares 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 releasing all the sections of the zine in Attribution-ShareAlike license <http://creativecommons.org/licenses/by-sa/3.0/> except News Bytes and Cool Tricks and you are free to use and distribute the information with proper attribution. If you like the work and find it useful then we encourage you to share and reuse the content 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.