LLP-LdV/IVT/2011/RO/188

Lifelong Learning Programme

Programme: Leonardo da Vinci, grup tinta IVT

root@user:~$ LET’S WIN LIN!

Partners: Alfatraining Bildungszentrum, Karlsruhe, Germany

Colegiul National Iasi, Romania.

Tutor: Slobodan Aleksic

Waltraud Koger

Anton Ioana Cătălina

Students:

1. Amariei Anca-Elena
2. Androniciuc Ana-Miruna
3. Crâșmăreanu Teodora
4. Codreanu Maria-Teodora
5. Diaconu Anca
6. Dima Andreea
7. Ferariu Tudor
8. Gheorghiu Teodora Diana
9. Harton Ioana
10. Lămătic Ioana
11. Nechita Mihai
12. Popa Teodor
13. Stoleru Ingrid Georgiana
14. Tătaru Mădălina Constantina
15. Ursu Cătălina Ioana

Summary

Chapter I – History of LINUX

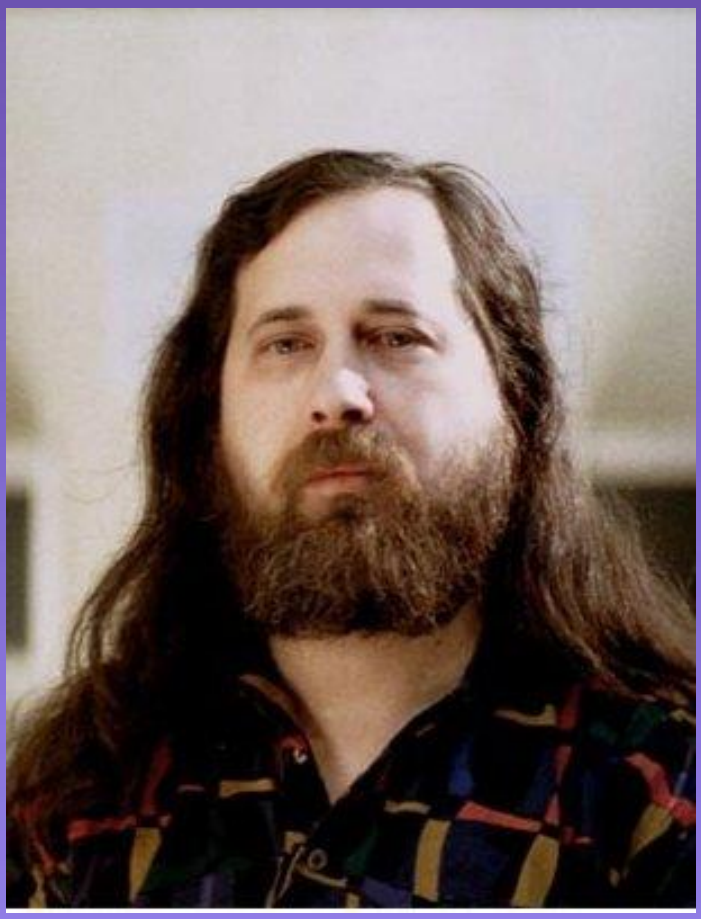
Whether you know or not what Linux is, or maybe you've even used it before, this will be your guide for discovering it, from what it is and the reason that stood behind its appearance to the possibility of changing and sharing it with your friends. Yes, we did say changing and sharing. Unlike other Operating Systems, Linux can be modified as you wish without breaking the law, but before we talk about this we should answer some basic questions, like…

What is this LINUX OS?

Chances are you've heard about Linux before reading this, but do you really know what it is? Linux isn't exactly an OS, it's just the kernel that makes the GNU/Linux OS functional.

Anyway, consider this GNU/Linux thing we're talking about a simple alternative to Windows. It was developed by hundreds of programmers who collaborated on the Internet. The thing is that behind this whole concept, there is an entire way of thinking based on a truly inspiring revolution for individual freedom.

This process began in the 1980's with the Free Software Movement and GNU project. And now it is most commonly associated with Linux and the Open Source Movement.



What is an OPERATING SYSTEM?

First we should clear up what an OS is. Basically, it is the main connection between us and the programs we wish to use. Sadly, an operating system is not human so it does not have free will, it only waits for the programs to ask for certain resources and then to connect them to the outside world. Its purpose is to step in and make it easy for us to write programs.

What exactly does OPEN SOURCE mean?

Open Source refers to the philosophy of free redistribution and permission to change the parts of the product (software). One of consequences of Open Source movement in the technology field is that the users of a certain software or programs are able not only to discuss and collaborate, but also to contribute to it. Sacrificing the intellectual property rights and letting the

whole world use the software and see the source code, the producers of the OS allowed everyone to be part of the evolution of the software.

Richard Stallman and the Free Software Movement

Before there was Open Source, there was the Free Software Movement. It is a social and political movement which grants the software users' four basic freedoms: the freedom to run their software, to study and change their software, and to redistribute copies with or without changes. These four freedoms are found in the The Free Software Definition, written by Richard Stallman.

Through his efforts to build the GNU Operating System, he created the legal, philosophical and technological basics for the Free Software Movement. It is highly important to say that freedom stands for FREE in "free speech" and not for FREE in "free beer", as Richard Stallman put it himself. Without these contributions, it's unlikely that Linux and Open Source would have evolved in to their current forms today.

You should also remember the following:

> The term "open source" software is used by some people to mean more or less the same category as free software. It is not exactly the same class of software: they accept some licenses that we consider too restrictive, and there are free software licenses they have not accepted. However, the differences in extension of the category are small: nearly all free software is open source, and nearly all open source software is free.
>
> — Free Software Foundation, *http://www.gnu.org/philosophy/categories.html*

While studying at MIT, Richard Stallman was faced with a dilemma: he was asked by a company to fix their computers but he couldn't do it."Why was that?", you may ask yourself…Well, the answer is quite simple: the operating system was proprietary and those who had developed its source code forbid the ability of editing the source code and sharing the OS. At that point he thought ""This is wrong! I am not going to live this way" And from experiences like this he developed a profound hostility to the idea of intellectual property and software.

When he realized that, as software developer, he could make a brand new alternative for the proprietary OSes he quit his job at MIT and started developing the GNU operating system . In this way he also managed to respect his values, as he later admitted : " Not only could I gave myself a way to keep using computers without betraying other people, but I'd give it to everybody else, too. Everybody would have a way out of that moral dilemma. And so I realized this was what I had to do with my life."

You probably realized by now that Free Software can't have a copyright, or at least a normal copyright. Wanting to make sure about

the well being and the profit of the Free Software and the users, Richard Stallman came with the idea of Copyleft.

"The idea of Copyleft is that it's "Copyright" flipped over. And what we do is, we say, this software is copyrighted and we, the authors give you permission to redistribute copies, we give you permission to change, we give you permission to add to it. But when you redistribute it, it has to be under these terms, no more and no less. So that whoever gets it from you also gets the freedom to cooperate with other people, if he wants to. And then, in this way everywhere the software goes, the freedom goes, too. And it becomes an inalienable right to cooperate with other people and form a community. **" says Richard Stallman.**

"GNU is not Unix"

GNU is a acronym standing for "GNU's not Unix" used for highlighting that despite the resemblances between the two OSes , they were different systems. An important distinction between them is the fact that Stallman and his team had to write Unix-like programs from scratch. By 1991 he had configured all the needed components of his newly born OS, except the kernel.

What is a kernel?

The kernel is the main component of most computer operating systems, and it could be call "the bridge" between the hardware and the software. It manages the system's resources, allocating the to all the other programs.

The Linux kernel

Unix consists of a large number of programs that communicated with each other, so the GNU project had to done program by program. It started with the basic tools: a C compiler, a debugger, a text-editor etc. The last piece that had to be done was developing a kernel to sit underneath the programs and be the center of the operating system. By about 1990 they had successfully developed that toolkit, and it was in wide use on great many variants of Unix. But there was still no free kernel, but then Linus Torvalds came along.

When Linus developed the kernel he wasn't doing it for the GNU project. He released it on his own and the people who did know about it decided to look for what else they could find to put together with it to obtain a whole system. Not very long after they had found all the pieces of the GNU system which was missing just the kernel, so when they put all that together really they "invented" the GNU/Linux OS.

“There's a symbiosis between Linux and the programs that the programs run on Linux and at the same time and they take the advantage of Linux as a platform, while Linux takes the advantage of the programs by just being able to use them. “

Richard Stallman

Other distributions of the LINUX kernel:

Fedora, CentOS (Red Hat)
Ubuntu (Canonical Ltd.)
Mandriva Linux (Mandriva)
Debian
Gentoo
There are Enteprise Linux distributions as well:
RedHat Enterprise Linux (RHEL)
SuSE Linux Enterprise Server (SLES)
Univention Corporate Server (UCS)

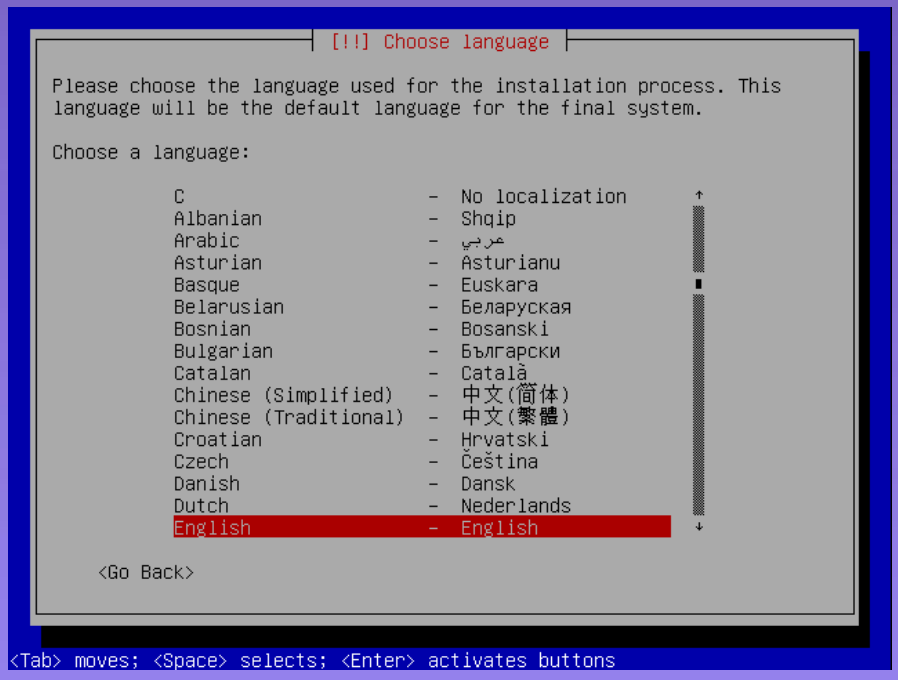
Chapter II – Installation of LINUX

Everyone uses at least one operating system. It sounds simple in the beginning because you think you know the basics of using a PC. You know how to use the keyboard and the mouse and you think it's enough before you copy an online installation guide and read the first page(Things aren't that simple as you thought, are they?)

You may laugh when you will the right picture but that was exactly how I felt when I was for the first time put face to face with Linux content of which I just had heard.

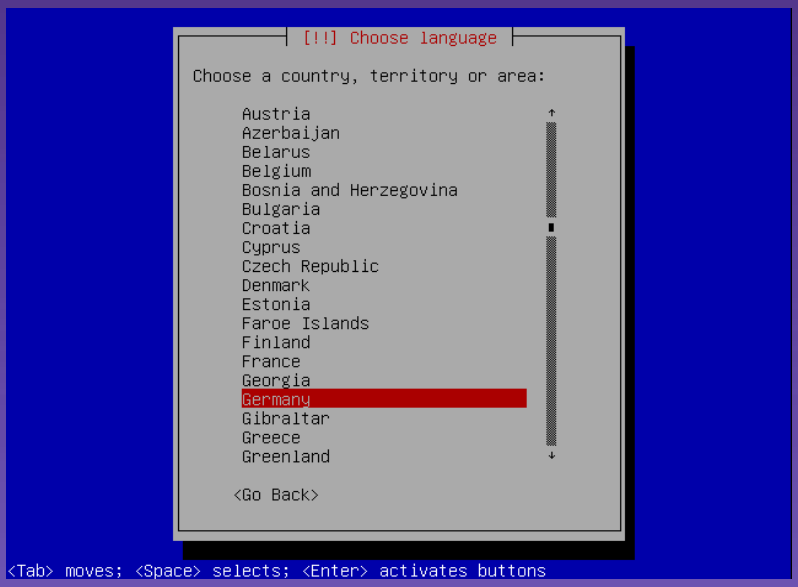
When you thought that every step is followed by the next one by simply clicking "next" button you hit the first error. But you are not the only one who will face these problems. Let me tell you how I've managed to install Debian GNU/Linux. It didn't take an hour or two, it took almost a day but I wasn't discouraged .

I will show you step by step the instructions on how to install Debian GNU/Linux. First of all I was supposed to make sure my PC is configured to boot from CD/DVD. This meant I had to enter the BIOS setup. For some this might have been easy, but for me it wasn't. I had to reboot my system for at least 3 times before I could entered the setup. When I finally got there I had to make sure that the CD is the first booted driver so that the installation from the CD I have inserted will start. If the boot from the DVD is successful then you should see an image of Debian where you will see the following options: Install, Advanced options, Graphical Install and Help. Of course I pressed Install.



First I had to select the language for the installation process and for the menu. I chose English because I was already familiar with the commands. You have the freedom of choosing any other language give in the menu,

such as Greek or Chinese if you wish.



After this I saw a short list of locations based on the language I had selected before. I chose *Other* because my location was not on the list. After I saw another screen which contained a list of continents of which I selected of course, Europe. After I saw a list of countries in Europe from which I selected Germany because this was the country where I installed my Linux OS.

Next, another grey screen appeared and I was asked to choose my locale settings. I didn't know very much what was this about but I found out after that this meant a short list of parameters which symbolized the language, the country and a special variant preferences that the user wants to see in the user interface.

After this I chose my keyboard language in English because this is the keyboard I use although it was a little difficult because the signs form German keyboard don't match. That's way after finishing the installation when I was supposed to write a command I usually mistaken some signs and it had taken me more time to write and execute the command.

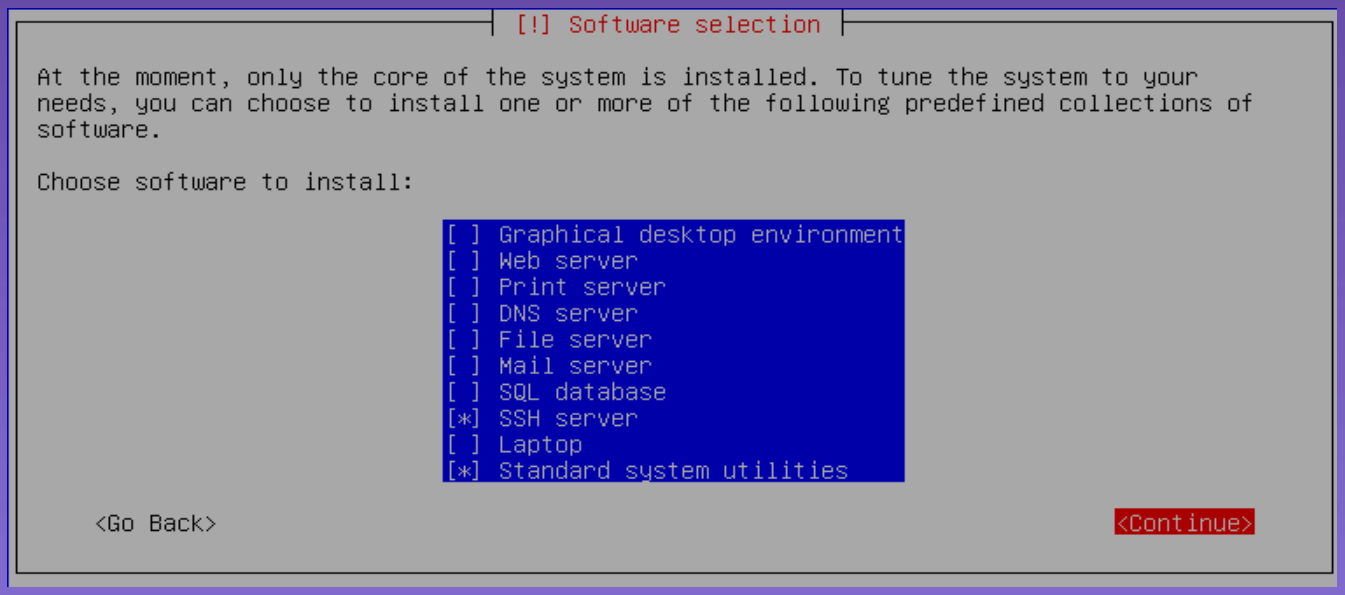
From now on each setup step would be very important for the right configuration of my Linux OS. The next grey screen showed the partition disks. I didn't know much about this part of the installation of a OS because every time I had my Windows reinstalled somebody did this part so I've chosen USE ENTIRE DISK.

I was told to press enter a couple of times but not too fast so that I couldn't miss the grey screen where I was supposed to set up the users and passwords . Be careful to write them down in case you forget them. The default password for the root user was in my case *test* so that we all had the same one. I was able to change this password after finishing the installation so that no one could access the administrative account.

I was now asked to enter my full name (the real one) and next to set up my username and my password. Be careful that your username starts with a lower case letter which can be followed by any combination of numbers and other lower case letters. I was asked to use them each time I started up my computer which had a Linux OS installed.

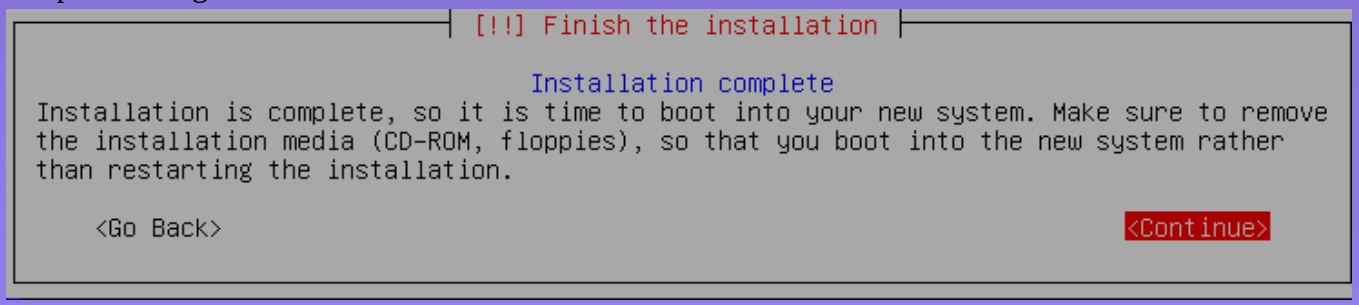
Once I set up a username and a password (good enough so that no one will ever find out) I was asked to select the package manager (don't get scared about the name because is just a list of countries and you just have to select your real one). Next I got a list of mirrors from the country I have just selected. I didn't know much about this new concept and my tutor explained me that a mirror site is an exact copy of another Internet site. To understand better he told me they are multiple sources of the same information and provide reliable access to large downloads. To allow faster downloads for users at a specific geographical location I was asked to choose the real country I was installing the OS.

After this I reached the top of the installation where I was supposed to choose the software for my new computer. Here I chose only the SSH server, the Standard System Utilities and the DNS server. I didn't chose the Graphical Desktop Environment because it was taking a lot of time (approximately 2-3 hours). So after only one hour I had my Linux OS installed but without a graphical desktop so I had to



enter this command manually by entering in a terminal and typing *aptitude install gnome*. But before this my system rebooted and that meant I installed an OS correctly.

If you manage to get here you are now one of the 60 887 879 users that have a Linux OS on their computer. Congratulations!



There are more than 10 years since I've been using a computer and since I was 8 years old I've got used with the window's 4 colored sign. In time it had become my other window, the one which I saw every day and I couldn't imagine that in time I will use another interface or another Operating System.

Linux? What's Linux? All my life I've got used with the small window in the background of my desktop and now a penguin took its place."I will never use Linux" I promised, but guess what…not only that I used it for 3 weeks but I also learned how to install it and how to work in it.

You will probably ask what was my motivation to do that or the reason that I've spent almost 3 weeks trying to get familiar with it. We were told that the world biggest companies and the international universities are using this type of OS. The bank and school systems nowadays prefer Linux/GNU OS over Windows for many and good justified reasons.

So here comes the problem. I was stubborn enough not to want to change my old-friend windows with one I've never heard of but I had the chance of trying to find out what it's all about. I could have skipped this problem but I realized that I might face with it in the future in a more harsh way.

The moment I've found out that I could have both Linux and windows on my PC was the moment that I decided to give Linux a try and to insert the installation CD in the CD-ROM…and this is even simpler than it sounds. First I've chosen a version of Linux that will function in parallel with Windows(such as Mint, Ubuntu or Fedora)after I had to enter the BIOS and to make sure that the first bootable device was the CD ROM and after I've continued the installation as I've presented before. After following again the normal steps of the Debian installation there is a YES/NO question that you will be asked in order to have both your Windows and Linux OS. Good luck switching between them!;)

Chapter III - Basic Commands

In the previous chapters, we described how we learned to install the Linux OS on our computers, so now, as we are sitting in front of them, we are going to see how certain commands are used to carry out different tasks.

First of all, each of us learned that the commands must be typed in a console, and the way we control the computer's activity is named CLI (command line interface). There are 6 consoles, which can be opened pressing the following combination of keys on our keyboards: Ctrl+Alt+F1, Ctrl+Alt+F2, ……, Ctrl+Alt+F6. If you press Ctrl+Alt+F7, you can return to the graphical interface, also known as X Server.

The commands that we have written on the command line are going to be read and interpreted by the shell. The shell is a programming environment that can be used to perform tasks. We used the most common shell, called bash (the Bourne-Again Shell - /bin/bash). Other shells we learnt about were: the Bourne Shell (/bin/sh), the Korn Shell (/bin/ksh), the C Shell (/bin/csh) and Tom's C Shell (/bin/tcsh).

From now on we are going to concentrate only on the commands we use, taking into consideration that every user has to know them very well, and type them correctly, because if we mistype a letter or a space nothing or something wrong will happen. We also have to know exactly what command is used for a certain task. After every command you must press Enter.

1. User administration

The first question of any new user would be how to log in. Let us tell you what we have done. We have to open the Root Terminal (Applications- Accessories- Root Terminal) and if we want to log in as a user, we have to type **su username** and then the password of the user. We also needed to add new users, so the tutor presented the command that we should use: **adduser username**. Be careful! You will have difficulties if you forget or mistype the password, as you will not be able to log in. Believe us, we have already experienced such issues. You can also change you password with the command **passwd username**. What would you do if you forgot who you are? We have a suggestion that may help you: the command **whoami**, which shows the current user. When we were logging out, we were using the **exit** command. And because we discussed this subject, we thought that **last** (who was logged in) and **w** (who is logged in) commands should be also reminded. We also learned how to create (**groupadd newgroup**) and delete groups (**groupdel newgroup**). To add a user to a certain group, we had to issue the command **gpasswd –a username group**. It would be useful to know the IP address assigned to your computer, as it must be used when your neighbour want to connect to your PC, so you can use the command **ifconfig**.

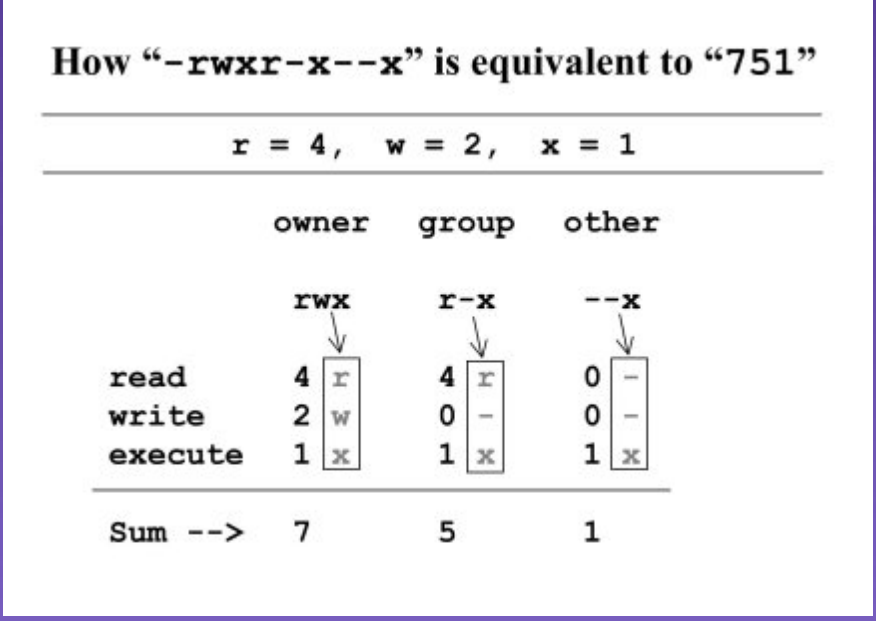
```
eth0      Link encap:Ethernet  HWaddr 00:17:31:83:99:20
          inet addr:192.168.0.253  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::217:31ff:fe83:9920/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2314 errors:0 dropped:0 overruns:0 frame:0
          TX packets:932 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:781222 (762.9 KiB)  TX bytes:109565 (106.9 KiB)
          Interrupt:20 Base address:0xdc00

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
RX bytes:480 (480.0 B)  TX bytes:480 (480.0 B)
```

If we wanted a new IP address, we would use the **dhcpcd** command, which allows hosts on a TCP/IP network to request and be assigned this kind of addresses. Being part of a group means also to share files, so we were taught how to change permissions and ownerships. We can operate with a file in 3 ways: we can read, write or execute it, so there is a kind of code that is assigned to every file showing what can an user, group or another person do with it.

This is what such a code looks like: -rwx------

Now, you maybe are asking yourself what should you understand from this succession of characters. Well, each group of three letters stands for owner, the group he belongs to or the others. To change the permissions and the ownerships we must use **chmod SUM filename/directory** command. And now, let's take a look on how the sum can be obtained. A number is assigned to each action between reading, writing and executing as follows:



The sum is a number that has three figures, each of these standing for the owner, group or the others. The figures are calculated by adding the numbers assigned to the actions that can be undertaken by them. To change the ownership of a file, we use **chown**.

2. ***Files & Directories***

Someone who is using a computer has to work with files and directories. To see all the contents of the current directory, we must type **ls**. The command **ls –al** is showing a list of all files. We also had to find a file that we hadn't know anything about. We used the command **ls –a**, showing the hidden files. In order to create one or more directories, the command used is **mkdir /home/username/folder1** or **mkdir –p /home/username/folder1/folder2/folder3**, while the its removing can be done with **rm –rf /home/username/folder1** . We can also create files that can be found in a certain directory with the command **touch filename > folder1**. To remove them, we use to type **rm filename**. If we want to write something The in a file we can use the following command: **echo text >> filename**. To see the content of a file, we can choose between **cat filename** and **less filename**. We are able to copy files with the command **cp filename1 filename2** and their moving can be realized by **mv filename1 filename2** (Tip: this command can be used also for renaming the file).

```
root@teodoracrasmareanu:/home/teodora# touch filename1
root@teodoracrasmareanu:/home/teodora# touch filename2
root@teodoracrasmareanu:/home/teodora# echo text1 > filename1
root@teodoracrasmareanu:/home/teodora# cat filename1
text1
root@teodoracrasmareanu:/home/teodora# echo text12 > filename1
root@teodoracrasmareanu:/home/teodora# cat filename1
text12
root@teodoracrasmareanu:/home/teodora# echo text13 >> filename1
root@teodoracrasmareanu:/home/teodora# cat filename1
text12
text13
root@teodoracrasmareanu:/home/teodora# mv filename1 filename2
root@teodoracrasmareanu:/home/teodora# cat filename2
text12
text13
```

The directories can be copied by using **cp –r /dir1/ /dir2/** . The command **pwd** shows the current directory.

```
root@teodoracrasmareanu:/home/teodora# pwd
/home/teodora
```

In order to change the directory, we have to use the command **cd**. For example, with **cd ~** we can change the directory to home, with **cd ..** we can go one directory up and with **cd /path/directory** we can change

to the directory /path/directory. To make a symbolic link from a file to another we should use the command **ln –s file1 file2** (Example: **ln –s /dev/null /dev/NULL**, in this case the first file is treated as a link to the second one).

3. *Finding Files and Applications*

To find a file, we can search it in the database with **locate filename**, or start the searching from the root with **find / -name filename**. For example, our tutor asked us to carry out some tasks that included also finding all the files in a certain directory that were owned by an user. What would you do? Well, we thought that would be a good idea to use this command: **find /DIRECTORY –user USERNAME**. Now, let's take a look on the commands used to find applications. Firstly, we learned about **which** and **whereis** commands, and then we were told that this action can be also done with **aptitude search APPLICATION_NAME**.

4. *Connection to the internet*

There were situations in which we had to check the connection to the internet. In that case we used the **mii-tool** command. Now, then we are sure that our PC is connected, we are able to download files and applications from any internet page, using the non-interactive command **wget** (Example: **wget http://192.168.0.210/Excercise1.pdf**).

5. *System Information*

The first test we had, at the beginning of the last week, revealed that we mistook **df –h** for **fdisk –l /dev/sda** (instead of sda, there can be also sdb, sdc, or sdd, for different hard drives). While **df –h** shows the harddisk usage, **fdisk –l /dev/sda** lists the current harddisk layout (the partition tables).

```
root@teodoracrasmareanu:/home/teodora# fdisk -l /dev/sda


Disk /dev/sda: 160.0 GB, 160041885696 bytes
255 heads, 63 sectors/track, 19457 cylinders, total 312581808 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000af5c4


   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *        2048   308654079   154326016   83  Linux
/dev/sda2        308656126   312580095     1961985    5  Extended
/dev/sda5        308656128   312580095     1961984   82  Linux swap / Solaris


root@teodoracrasmareanu:/# df -h
Filesystem                                      Size  Used Avail Use% Mounted on
rootfs                                          147G   14G  126G  10% /
udev                                             10M     0   10M   0% /dev
tmpfs                                            95M  632K   94M   1% /run
/dev/disk/by-uuid/5460451b-4ad6-4407-b112-0fd17e406f83  147G   14G  126G  10% /
tmpfs                                           5.0M     0  5.0M   0% /run/lock
tmpfs                                           190M  152K  189M   1% /tmp
tmpfs                                           190M  568K  189M   1% /run/shm
```

Other two important commands are **lspci** (displaying information about all the PCI devices) and **lsusb** (showing information about all the USB devices).

```
root@teodoracrasmareanu:/# lspci
00:00.0 Host bridge: Advanced Micro Devices [AMD] nee ATI Radeon Xpress 200 Host Bridge (rev 01)
00:01.0 PCI bridge: Advanced Micro Devices [AMD] nee ATI RS480 PCI Bridge
00:12.0 IDE interface: Advanced Micro Devices [AMD] nee ATI IXP SB400 Serial ATA Controller
00:13.0 USB controller: Advanced Micro Devices [AMD] nee ATI IXP SB400 USB Host Controller
00:13.1 USB controller: Advanced Micro Devices [AMD] nee ATI IXP SB400 USB Host Controller
00:13.2 USB controller: Advanced Micro Devices [AMD] nee ATI IXP SB400 USB2 Host Controller
00:14.0 SMBus: Advanced Micro Devices [AMD] nee ATI IXP SB400 SMBus Controller (rev 11)
00:14.1 IDE interface: Advanced Micro Devices [AMD] nee ATI IXP SB400 IDE Controller
00:14.3 ISA bridge: Advanced Micro Devices [AMD] nee ATI IXP SB400 PCI-ISA Bridge
00:14.4 PCI bridge: Advanced Micro Devices [AMD] nee ATI IXP SB400 PCI-PCI Bridge
00:14.5 Multimedia audio controller: Advanced Micro Devices [AMD] nee ATI IXP SB400 AC'97 Audio Controller (rev 02)
01:05.0 VGA compatible controller: Advanced Micro Devices [AMD] nee ATI RS400 [Radeon Xpress 200]
02:02.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL-8139/8139C/8139C+ (rev 10)
02:0a.0 FireWire (IEEE 1394): VIA Technologies, Inc. VT6306/7/8 [Fire II(M)] IEEE 1394 OHCI Controller (rev 80)
root@teodoracrasmareanu:/# lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 003 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 001 Device 003: ID 0424:2228 Standard Microsystems Corp. 9-in-2 Card Reader
Bus 002 Device 002: ID 046d:c046 Logitech, Inc. RX1000 Laser Mouse
```

Now, in the centre of our attention is another important group of commands: **cat /proc/**. The first one (**cat /proc/**) shows the IDs for all the processes that were running on the system. Then comes **cat /proc/ devices** (displaying information about the existing devices), **cat /proc/meminfo** (displaying information about the free and used memory, both physical and swap, on the system, as well as the shared memory and buffers used by the kernel) and **cat /proc/cpuinfo** (showing information about CPU – Central Processing Unit and its architecture). We also learned about **ldd** command, which prints the shared libraries required by each program or shared library specified on the command line (Example: **ldd /bin/bash** , **ldd /sbin/init**). So, we are going to take a look on the last group of commands, which concern the process management. Firstly, **ps ux** is supposed to show all the processes of the current user. Then comes **ps T**, which displays the processes on the current terminal. Last but not least, the **ps aux** command is used to show a list of all the processes. For example, we learned how to find the PID controller of a certain process (PID of the bash process: **ps aux |grep bash** , PID of init process – which has the value of 1: **ps aux |grep init**).

6. Stopping processes

If a process doesn't work properly, we have to stop it, so we can choose from 2 main commands: **kill** and **killall**. The processes can be stopped in many ways, depending on the signal we are using:

1 or SIGHUP = hangup or disconnect the process

2 or SIGINT = interrupt the process (same as Ctrl+C)

3 or SIGQUIT = quit

9 or SIGKILL = kill the process through a kernel call

15 or SIGTERM = terminate a process "nicely" (this is the default signal)

One example would be the following one: **kill -9 5445** (**kill –SIGNAL PID of the process**)

If we don't know the PID of the process, we can use **killall** command (**killall –SIGNAL name of the process**, for example **killall -9 gimp**)

7. Mounting Filesystems

Mounting is a mechanism used by the OS to access resources on a hard drive. There are some useful commands, that serves to attach the file system found on some device to a certain directory and we learned about: **mount filesystem directory** (Example: **mount/dev/cdrom/mnt**) and **umount filesystem directory (**Example: **umount/dev/cdrom).**

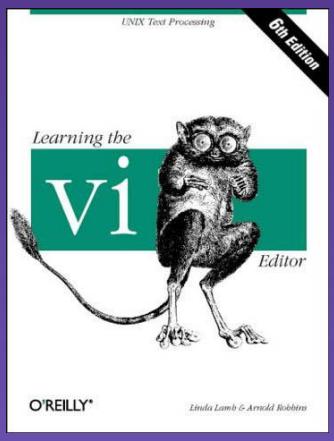
8. Bringing the system up to date

Every time we wanted to install a new pregramm, we had to use the commands **aptitude update** and **aptitude upgrade**. If we needed the PC to do this in a non-interactive manner we have just typed **aptitude update –y && aptitude upgrade -y** .

9. Starting and Stopping

A computer with Linux OS is easy to turn off, only with **halt** command. If we want to reboot the system we use **reboot** command.

Chapter IV – Programs



1.When I first heard about VIM, I was like "All right, let's see how it works.". But every story has a beginning and an ending, and this "one-day" story starts with one short description of this program. I found out that the original program was VI, which we used only to see the differences between the old form and the new one. According to our tutor, VIM stands for "VI IMproved" and, as the name says, it has to be better than his ancestor. Will see.

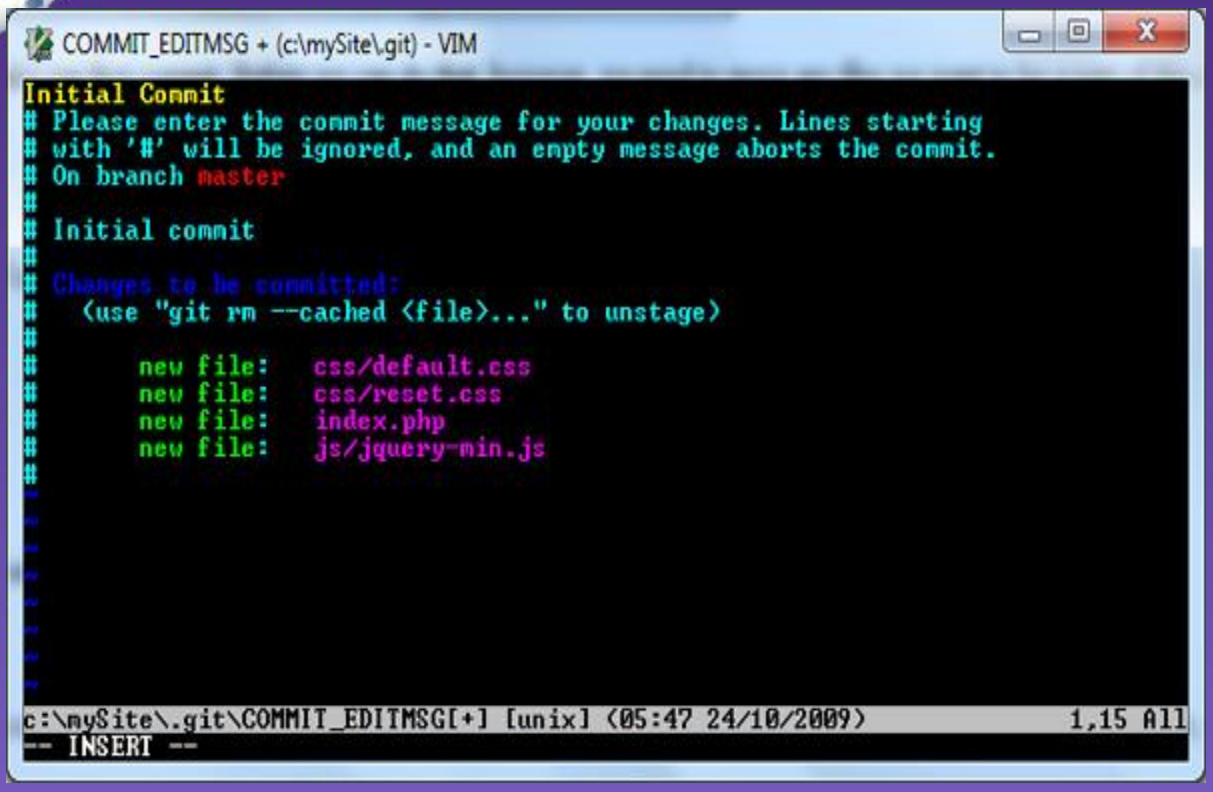
The editor VIM can be used to modify all kinds of plain text and we agree that it is especially useful for changing the characteristics of the programs. At this point, we know that every command should be done from the terminal, so just open the black screen, log in as root and the fun begins.

Firstly, a program is a program. We are all free to use it as long as it exists in our pc's. So, prepare yourself and type carefully : "aptitude install vim". Hard, isn't it ? I guess we are the best at installing different things on Linux. Why are we so good ? Because we all know what should be done before the actual installation of the program. Our tutor insisted on getting our system up to date, so firstly, do the universal steps "aptitude update" and "aptitude upgrade". This way, your VIM will work nicely every time.

A command that was used, as far as I know, more than once, was the online-help for the VIM program. I just typed ":help" and my friend, the Internet, showed me every single detail about the text editor. That's how we learned to launch VIM with the simple command "vim [filename]". From now on, only our tutor could reveal the dark secrets hidden in those empty lines or in the strange so-called words which appeared in front of us.

Well, keep calm and learn how to use VIM. Don't panic because you can't leave the page and you don't know how to write anything on the screen. Just follow the tutor's instructions. First of all, press "INSERT". That would work fine if we knew what was the German version. I forgot to specify that we all have German keyboards. So "ENTF" instead of the popular "INSERT".

Magic ! Now we can move the cursor and delete some parts of the file or add something to it. Well, in a file opened with VIM, you can use every single basic command you learned until now. Then, you just need to press ":w" to save all the changes you have made and, to exit the program, issue the command ":q". If you don't want to modify anything to the file, well, you could just leave VIM by pressing ":q!" and everything will be just fine.



I think the main difference between VI and VIM is the text color. VI only operates with white words, written on a black screen. But, when using VIM, a burst of color is noticed instantly. Every command has its color, so it's easier to find a certain section in a huge file. But that's not the only difference : we found out that we don't need to press any key to edit the text in VI. Just act like you are using Microsoft Word, for example, or Notepad. The exit could be done by pressing the "CTRL + X" combination; after that, you will be asked if you want to save your changes. That's up to you. We now use the VIM program for editing almost every file. We have even created our own file, which we edited as we wanted. The freedom can easily be felt while using Linux programs. The single condition, from as I know, is to type correctly every single command. Otherwise, your Gentoo installation, for example, won't work properly and you will start to count the errors shown by newly installed version of Linux.

1. The Debian operating system looked ok, from my point of view. But, in the first place, we learned how to install and configure it, so things were not as pink as I thought. Thanks to our tutor, the installation soon became completed. The last thing we needed to do was to install a graphical desktop to our pc, using the terminal commands. Because we didn't know much about Linux operating systems, we were rookies then, our tutor presented us a short description of Linux desktops, starting with Gnome Classic.

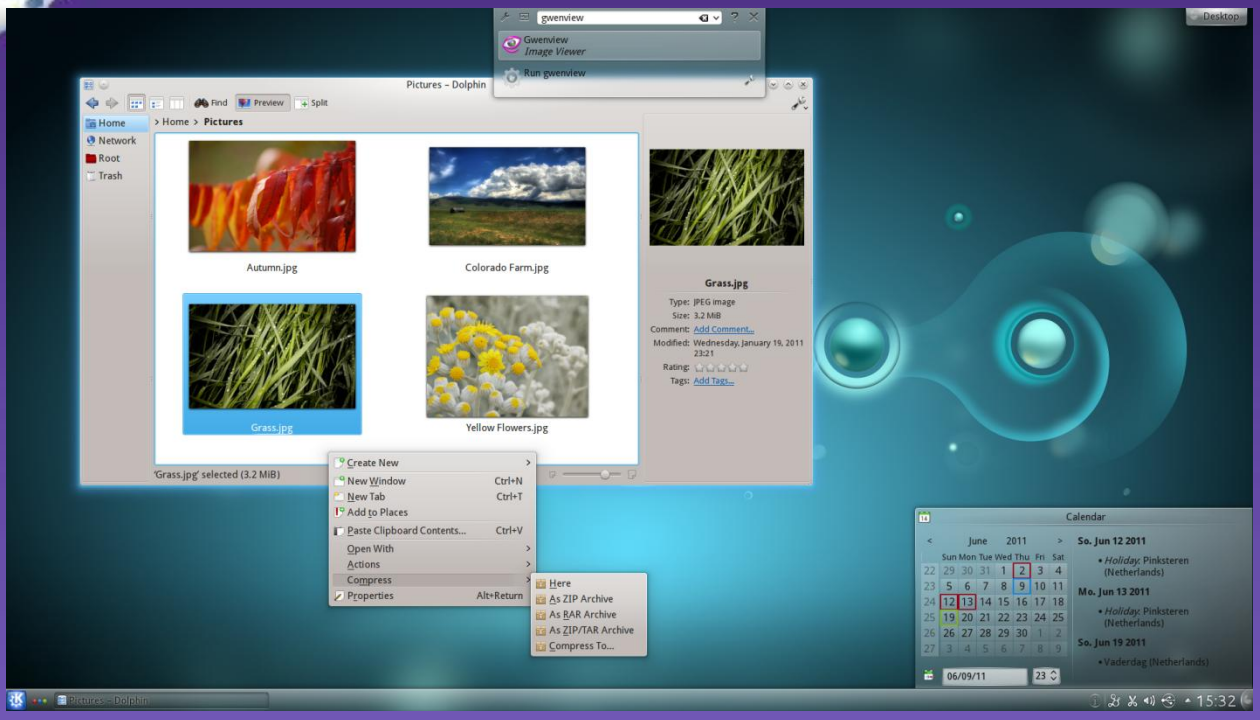
That's how we actually entered the Linux environment : by discovering that Gnome is an open source, a GPL product, used by most of the recent operating systems just like Fedora, Ubuntu or RedHat. That desktop exists as a confirmation of the so called "free" concept developed by Linux. You can use it everywhere, whatever you want and you can share it with whoever you want. I think this is the main idea of a program created by Linux.

But the question is : how are we going to install this to our computer ? Remember the "aptitude install" command I mentioned before ? Hope so, because this is the combination we were going to use : "aptitude install gnome". After pressing "Enter", we all thought that this process would last forever. Packing, unpacking, answering questions .. too many things to do when you just want to see some colors on a screen.

No errors. That seemed to be good. And it was. When we opened our computers the next day, we had a nice background, which looked exactly like this: :

Of course, it wasn't easy to get used to such interface, but we managed to find all we needed in the "Accessories"-and-"Places" bar, located somewhere up the screen. Everything was pretty categorized and organized, so we could only click on the tool we wanted to use. For me, that came just like a bless because my personal Windows desktop is full of useless pictograms. Now, all made sense to me.
Gnome is not the single graphical desktop offered by the Linux Corporation. No. We were amazed while we listened to our tutor speaking about KDE, another desktop environment, which is based on a different philosophy : this had to be fully equipped, completed and upgraded. Gnome is usually compared with this type of desktop. Well, we heard something about Enlightment, but only few managed to install this too.
Back to KDE, well, the process wasn't as simple as we all thought. We firstly needed to search for the full name of the program. Open the console and type carefully what ? You guessed, "aptitude search KDE". Wasn't that hard, I like these commands. So, we selected the "kde-full" package and we pressed Enter.
After a few minutes, we observed that the computer wasn't even installing something. The command we entered wasn't good, so repeat the "hide-and-seek" process. Meanwhile, we also installed the KDM, login manager for the desktop. That worked properly, we were happy about that.

Next, we installed the interface using the command "aptitude install kde-plasma-desktop" and everything was really good. Better than anything, considering the fact that we were all tired of waiting for a new desktop. After we waited a while, the process completed and we needed to restart our pc's and choose from a table which interface we wanted to use. I felt special because I had a variety of possibilities (only 2, but in Windows, there is only 1, as far as I know).
The screen started to act weirdly. Now, it's cyan, with a great design in the middle of the desktop. I liked it very much and I wanted to use it for the rest of the probation, but it was easier to follow the tutor's steps when we used the same version of desktop. Just my luck.
I started to activate every single button because that environment was so nice. I really think that another job of it is to calm down the spirits, owe to his colors. It was even a Dolphin in the left corner. That was the file manager and it was cute. Press F3 and voila! Two tabs appeared in the Dolphin program.



Playing with programs in Linux is funny, but actually personalize your newly installed interface is even more attractive. This lesson was my favorite. Colors, buttons to press, drawings, what else do you wish on a computer ? Or in the real life ?

-GENERAL IMPRESSION-

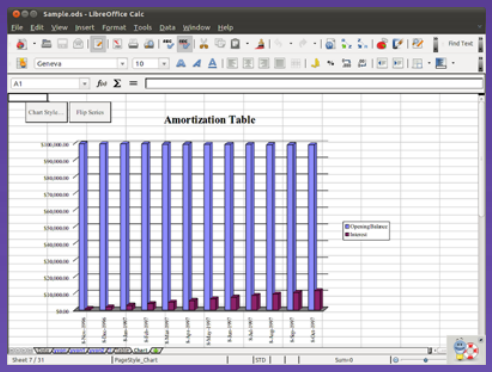
How easy might a project become once you have learnt the way to use the right instruments and by this, I want to recall you the presentation of Office Application. I was expecting, of course, to encounter another program which I haven't met before, but with the same characteristics as the ones that I use day by day. That is as much to say that we weren't expecting for something new when hearing the title of the chapter. Was it that way? Well, after I had a great breath of fresh air, I started to get used to the idea that I shall study one more theme, even if is not as exciting as we might expect, so I paid all my attention to the presentation. The first impression was of completely agreement with the fact that even if
I hadn't ever worked with Linux OS, using Office, which is so similar to Microsoft, was a piece of cake. I could appeal with it without s too many difficulties. As it came one with the whole packet taken from the CD, I was something like " Oh, how great to study a more accessible lesson without compiling, rebooting or reinstalling other auxiliary programs". Yeah, that was really going to be enjoyable and beyond the practical applications and the presentation itself, it was almost impossible for anyone not to understand the chapter: I really considered it a breath of fresh air. Not only is it free, but also is used worldwide, being available in more than 30 languages and for all major operating systems, so one is alike to consider it a high-quality alternative.

HOW TO WORK WITH IT?

As I mentioned before, it presents many similarities with Microsoft, which made me pretty pleased with the offered packet, giving us six feature- rich application for all kinds of documents and data processing needs: Writer, Calc, Impress, Draw, Math and Base.

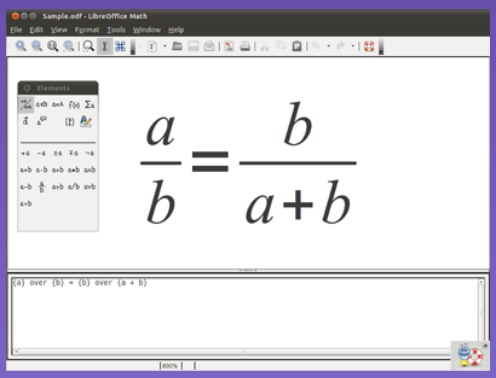
ADVANTAGES-

- **WRITER:**



- multiple use: dashing quick letters, producing tables of contents, embedded illustrations, bibliographies and diagrams;
- auto-completion, auto-formatting and automatic spelling checking;
- powerful enough to tackle desktop publishing tasks(creating multi-column newsletters and brochures)

- **CALC:**



- facility of introducing complex formulas;
- production of statistical analyses
- accessibility for suitable graphics (line, area, bar, pie)

- **IMPRESS**
  - Fastest and easiest way to create multimedia presentations.

DISADVANTAGES

- Working with graphics elements is sometimes perceived to be cumbersome
- Low quality of bitmap file export (e.g. size, spell checker markup)

-DESIGN-

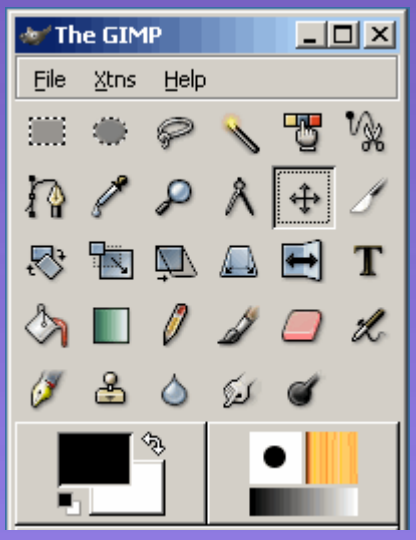
Moreover, when it comes to design, Libra Office is pretty similar with an older version of Microsoft, maybe one that belongs to 2003s' year. It seemed to me practical enough to be accessible even for the ones that haven't ever worked with it, which is maybe the best advantage for a novice user (who needs a simple and basic interface).

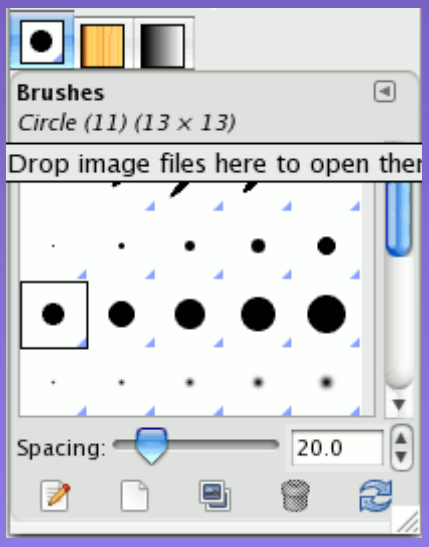
GIMP

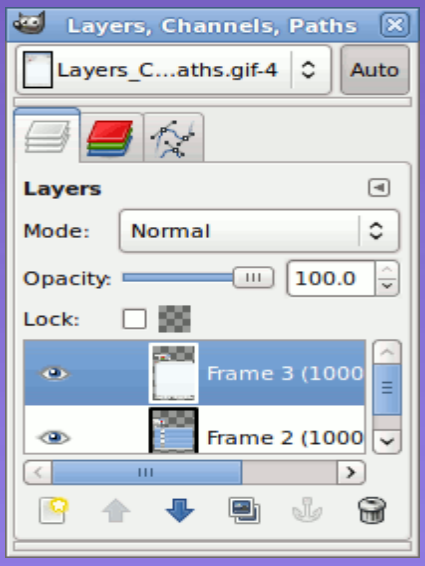
GIMP or GNU Image Manipulation Program is practically a tool used to retouch and edit pictures. What's great about GIMP is that is free, and not just for Linux, even though it comes together only with this operating system, but for all the systems including Windows, Mac, Solaris.

We all know that the material aspect is very important but I think the best thing about this program is that is so easy to use. Let's face it doesn't matter that it's free as long as you don't use it, then it becomes just another program from your computer, occupying useless bytes. You don't need to read a tutorial to edit images, you just open GIMP and then you start playing with the options the program provides and you get used to it very quickly and then it becomes a really fun way to spend time when you are really bored.

Well, of course it's not all rainbows and butterflies. Being so easy to work with can be also a disadvantage because its easiness comes from the fact that there aren't that many tools, like in Adobe Photoshop, and for those interested in editing every single aspect of a picture, GIMP is not that useful but I strongly believe that it's the best choice for basic editing.

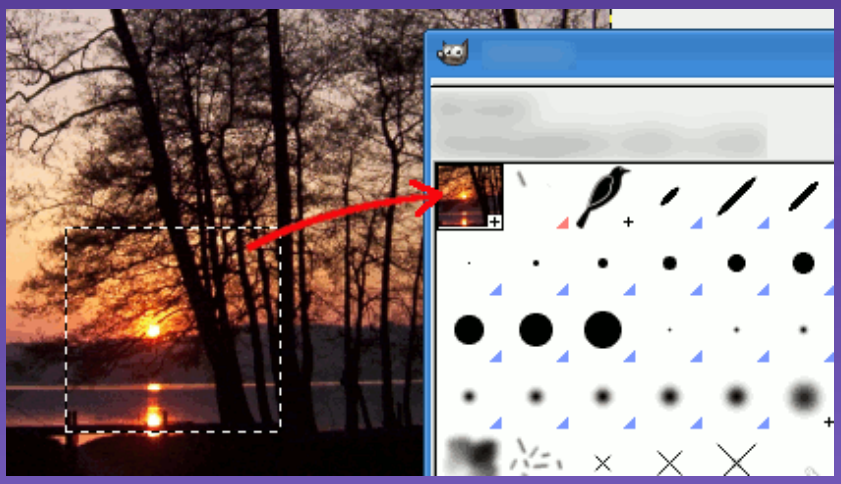


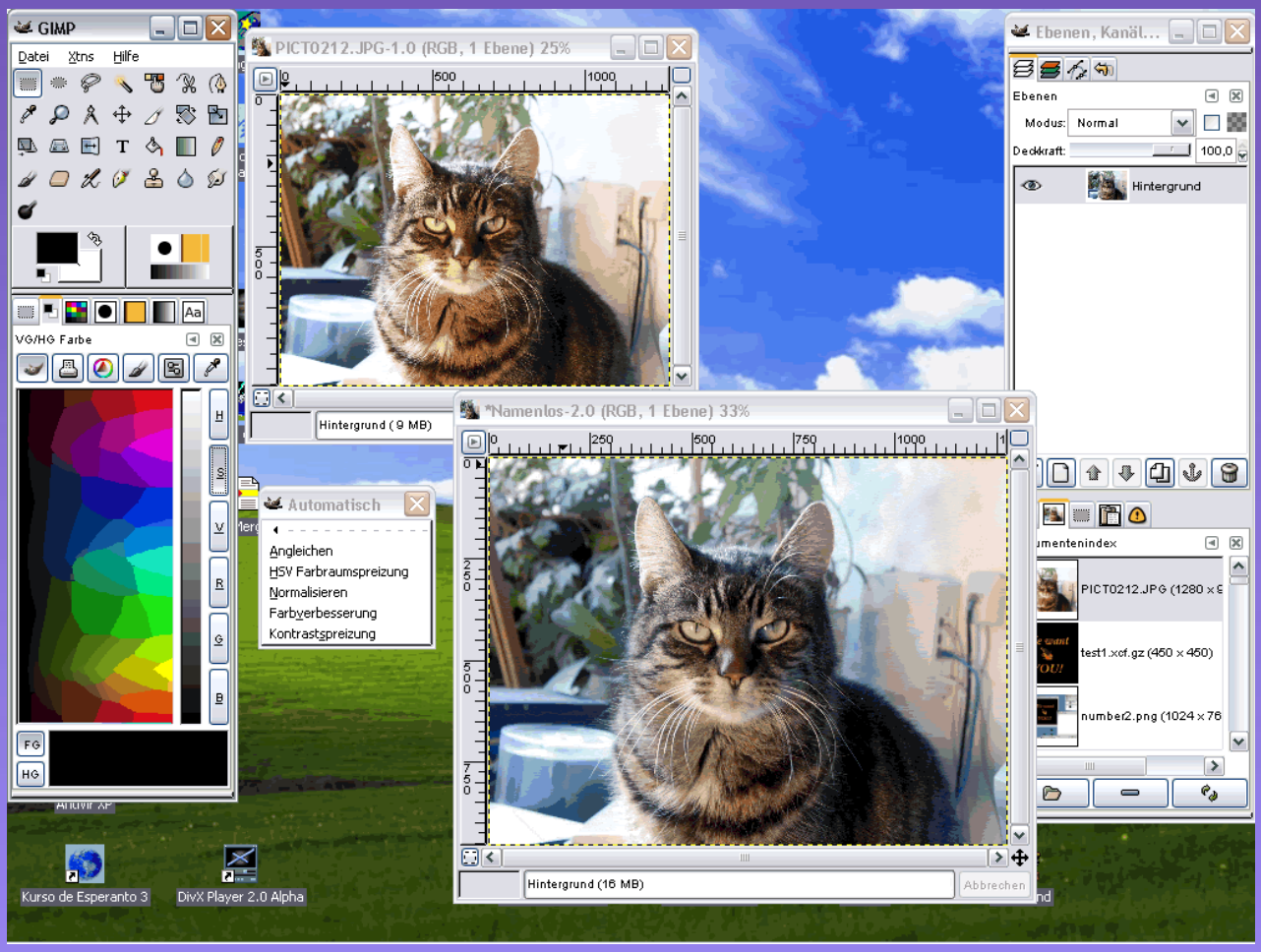




The tools used to edit the images can be accessed from the toolbar, through menus and dialogue windows. You will find filters and brushes and tools for transformation, selection , layer and masking. A funny thing is that when you paint with the brush the output color slowly changes. Confusingly, GIMP uses two types of color palettes (indexed and normal) but maximum number of

colors allowed in an indexed palette is 256. This palette is called "indexed" because each color is associated with an index number. Actually, the colors in the ordinary palette are numbered as well, but the numbers have no functional significance. Another thing that caught my eye is that you can create a brush from an image. It's like this: you select, whit a select tool, of course the area that you want to convert into a brush then you copy it and then you see this copy in the brush dialogue.

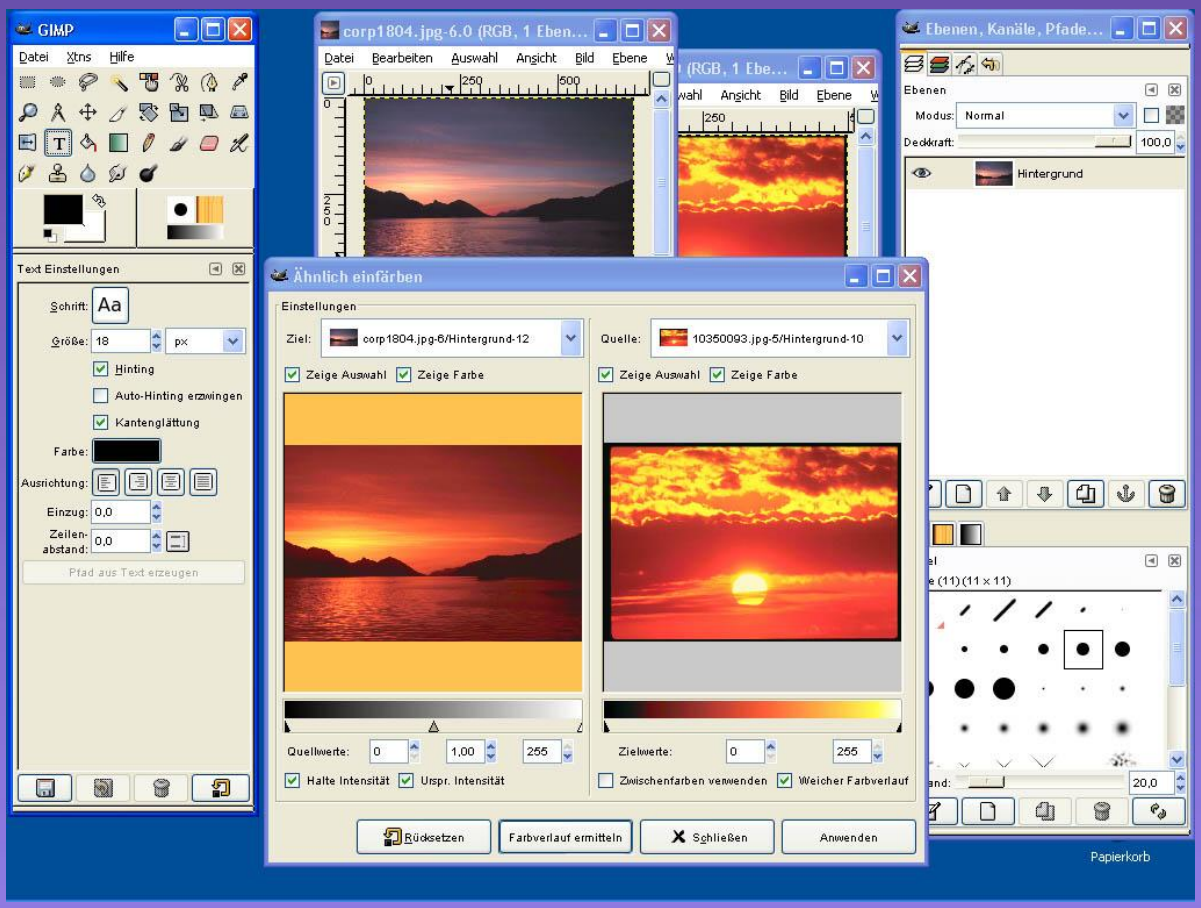




You can adjust the lightness of a picture, as you can see in the example above, change the eye color, the size of a particular feature or you can simply draw anything you want, so if we consider this last matter, GIMP it's fun for kids too, so if you want your child to be quiet just leave him playing in GIMP.

Another great feature of this program is that you can export a picture in almost any format you want, including BMP, JPEG, PNG, GIF, and just like this, GIMP becomes a very easy way to change the format of a picture, adjusting it to your needs.

I could go on and on about this program because there's so much to say, but there's no point. I give you the chance to try GIMP and discover yourself the advantages brought by it.



Chapter V - Networking

In order to talk about Linux network particularities, we must first understand what a network is in general. One of the most common mistake people made, is confusing the internet with the network. A computer network is a network of data processing nodes which are interconnected for the purpose of data communication. Meanwhile, the internet is the interconnection between many different and individual networks. Nowadays, it evolved so much that it serves millions of users and has a multitude of purposes such as messaging, file transfers, online video conferences, etc.

A few properties of networks :

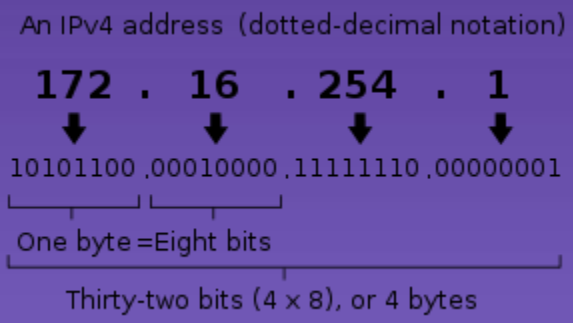
1) **Facilitate communications**
   - Using a network, people can communicate efficiently and easily via email, real-time messaging programs, chat rooms and video conferencing.
2) **Permit sharing of files, data, and other types of information**
   - In a network environment, authorized users may access data and information stored of other computers on the network.
3) **Share network and computing resources**
   - In a networked environment, each computer on a network may access and use resources provided by devices on the network, such as printing a document on shared network printer.
4) **May be insecure**
   - A computer network may be used by computer hackers to deploy computer viruses of worms on devices connected to the network.
5) **May be difficult to set up**
   - A complex computer network may be difficult to set up by average users. Also, it may be expensive for companies to hire trained men for this job.

As you can see, similar to the most useful things, it has some drawbacks.

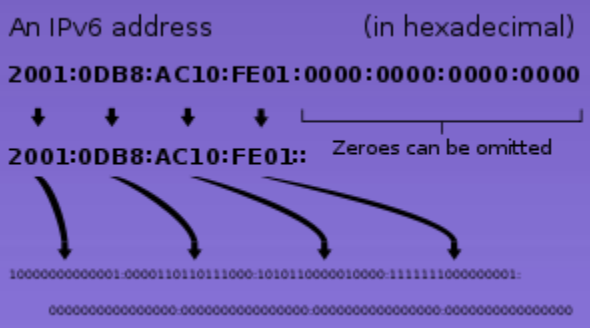
The IP address

To obtain an IP address, you must first plug in the Ethernet cable (double-check with **mii-tool**) and then issue the **dhcpcd** command, if it's not done by the system. The ip address is a numeric label attached to each device connected to a computer network that uses **I**nternet **P**rotocol for communications. There are two versions of Internet Protocol in use: IP Version 4 (IPv4) and Version 6 (IPv6).

1. IPv4 is an address which consists of 32 bits, limiting the address space to 4 294 967 296 (2^{32})possible unique addresses. An IPv4 address is represented in dot-decimal notation, which consists of four decimal numbers, each ranging from 0 to 254 separated by dots. Each part represents a group of 8 bits of the address. Computers not connected to the internet such as factory machines that communicate only with each other via TCP/IP don't need unique IP addresses. There are currently 3 types of IPv4 addresses for private networks: 10.x.x.x., 172.y.x.x, 192.168.x.x, where x is a number from 0 to 254, and y form 16 to 31.



2. The rapid exhaustion of IPv4 address space, determined engineers to explore new technologies to expand the Internet's addressing capability. The solution was to redesign the Internet Protocol itself. The new generation's size was increased from 32 to 128 bits. The new address provides the potentioal for a maximum of 2^{128} unique addresses.

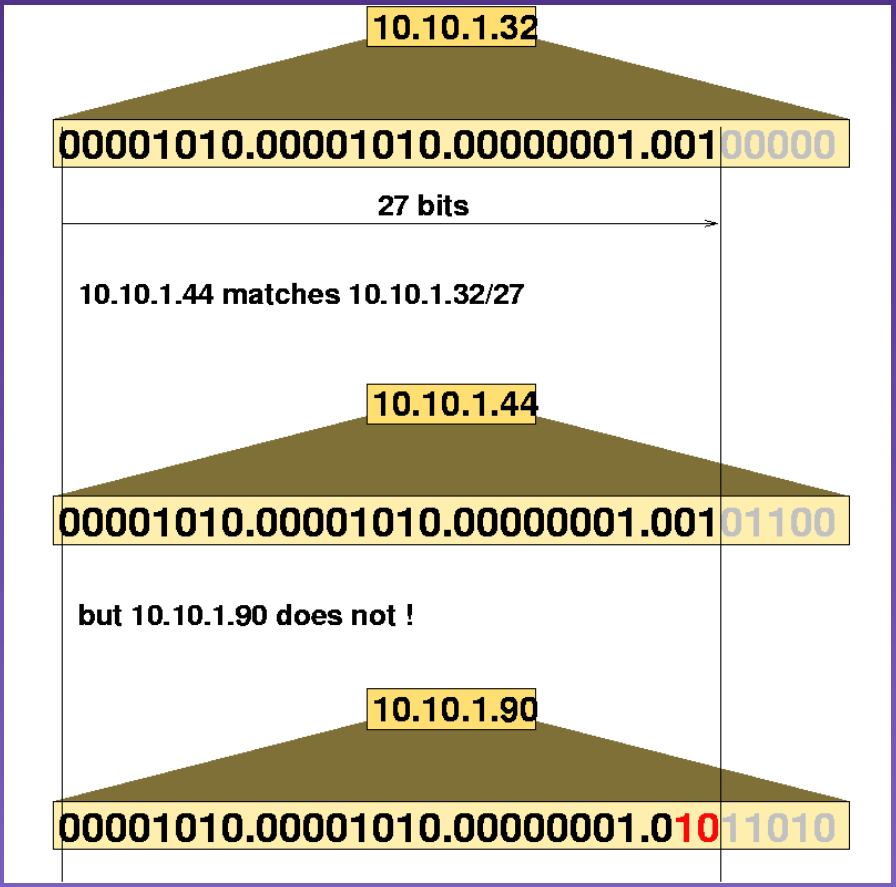


Firewalls and IP blocking

Firewalls are the most important tools for a network's security. They are meant to protect networks from unauthorized access. Usually, they control the access to networks based on the IP address of the client computer. The blocked IP address, is the one perceived of the client, meaning that if he is using a proxy server, blocking one IP address may block more than one computer. We had to modify the firewall's settings a few times, due to the high security level of the network.

CIDR

Classless **I**nter-**D**omain **R**outing is a method for allocating IP addresses to different machines in the same network. All IPv4 addresses are structured like a.b.c.d/n. The a,b,c,d represent the standard decimals of the IP itself. However, n, represents how many binary digits have to form the so-called prefix. For example, in the picture we see 10.10.1.32/**27**, meaning 27 digits have to coincide in orde to belong to the same network.



SSH

Secure Shell (SSH) is a network protocol for data communication between two networked computers, a server and a client, that it connects via a secure channel.

The best-known application of the protocol is accessing Unix-like operating systems. It was designed as a replacement for other insecure protocols, which send information, notably passwords, in plain text, making tem easy to steal. The encryption used by SSH is intended to provide confidentiality and integrity of data over an unsecured network, such as the Internet.

SSH uses public-key cryptography to authenticate the remote computer. Anyone can produce a matching pair of different keys (public and private). The public key is placed

on all computers that must allow access to the owner of the matching private key (the owner keeps the private key secret). While authentication is based on the private key, the key itself is never transferred through the network during authentication.

SSH only verifies if the same person offering the public key also owns the matching private key. Hence, in all versions of SSH it is important to verify public keys before accepting them. Accepting an attacker's public key will authorize an unauthorized attacker as a valid user and might bring serious harm to your computer.

On Unix-like systems, such as Linux, the list of authorized keys is stored in the home folder of the user on which you are allowed to log on to, in the file authorized_keys, not a very original name. When the public key is present on one side and the matching private key is present on another side, typing in the password is no longer required, so the standard password option is usually removed, that way only people whose public key you've accepted can log onto your PC. However, for additional security the private key itself can be locked with a passphrase.

The SSH command is typically used to transfer files at high speed or to remotely access a computer and issue various commands in order to help (or not) .

```
File  Edit  View  Search  Terminal  Help
root@tudor:/home/turby# ssh -p5555 root@192.168.0.191
The authenticity of host '[192.168.0.191]:5555 ([192.168.0.191]:5555)' can't be
established.
ECDSA key fingerprint is 6e:01:af:1c:16:de:4d:24:d6:7a:b3:1d:79:f0:80:72.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[192.168.0.191]:5555' (ECDSA) to the list of known h
osts.
root@192.168.0.191's password:
Linux mickeypc 3.2.0-2-686-pae #1 SMP Mon Jun 11 18:27:04 UTC 2012 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jun 26 11:29:03 2012 from localhost
root@mickeypc:~# killall -9 chromium
root@mickeypc:~#
root@mickeypc:~# reboot
```

Even if it seems like configuring ssh is not that hard, it took us a while to get it working, so here's what we did:

1.First of all we had to find and edit two files (ssh_config and sshd_config), in order to change our Port number and refuse standard password authentication.

```
File  Edit  View  Search  Terminal  Help
root@tudor:~# cd /etc/ssh
root@tudor:/etc/ssh# ls -al
total 184
drwxr-xr-x   2 root root   4096 Jun 26 11:24 .
drwxr-xr-x 128 root root  12288 Jun 27 09:00 ..
-rw-r--r--   1 root root 136156 May 26 15:03 moduli
-rw-r--r--   1 root root   1671 Jun 26 11:24 ssh_config
-rw-r--r--   1 root root   2489 Jun 26 10:05 sshd_config
-rw-------   1 root root    668 Jun 12 12:34 ssh_host_dsa_key
-rw-r--r--   1 root root    600 Jun 12 12:34 ssh_host_dsa_key.pub
-rw-------   1 root root    227 Jun 12 12:34 ssh_host_ecdsa_key
-rw-r--r--   1 root root    172 Jun 12 12:34 ssh_host_ecdsa_key.pub
-rw-------   1 root root   1679 Jun 12 12:34 ssh_host_rsa_key
-rw-r--r--   1 root root    392 Jun 12 12:34 ssh_host_rsa_key.pub
root@tudor:/etc/ssh#
root@tudor:/etc/ssh#
root@tudor:/etc/ssh#
root@tudor:/etc/ssh# vim /etc/ssh/ssh_config && vim /etc/ssh/sshd_config
```

Once we found the files we edited the Ports and the Password Authentication option.

It is best if you have the VIM program preinstalled for easy editing.

```
File  Edit  View  Search  Terminal  Help
#   CheckHostIP yes
#   AddressFamily any
#   ConnectTimeout 0
#   StrictHostKeyChecking ask
#   IdentityFile ~/.ssh/identity
#   IdentityFile ~/.ssh/id_rsa
#   IdentityFile ~/.ssh/id_dsa
    Port 1111
#   Protocol 2,1
#   Cipher 3des
#   Ciphers aes128-ctr,aes192-ctr,aes256-ctr,arcfour256,arcfour128,aes128-cbc,3d
es-cbc
#   MACs hmac-md5,hmac-sha1,umac-64@openssh.com,hmac-ripemd160
#   EscapeChar ~
#   Tunnel no
#   TunnelDevice any:any
#   PermitLocalCommand no
#   VisualHostKey no
#   ProxyCommand ssh -q -W %h:%p gateway.example.com
    SendEnv LANG LC_*
    HashKnownHosts yes
    GSSAPIAuthentication yes
    GSSAPIDelegateCredentials no
:wq
```

```
File  Edit  View  Search  Terminal  Help
# Package generated configuration file
# See the sshd_config(5) manpage for details

# What ports, IPs and protocols we listen for
Port 1111
# Use these options to restrict which interfaces/protocols sshd will bind to
#ListenAddress ::
#ListenAddress 0.0.0.0
Protocol 2
.
.
.
.
.

# Change to no to disable tunnelled clear text passwords
PasswordAuthentication no

                                                          44,1          45%
```

2. Then we had to create the ssh keys with the ssh-keygen command and locate them in the specified folder. To check if they work set them as part of our own authorized keys in order to connect to our own PC and see if they work properly. We then had to reset the ssh daemon, unfortunately the restart option wasn't working properly at the time so we had to terminate the process first and then start it again.

```
File  Edit  View  Search  Terminal  Help
turby@tudor:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/turby/.ssh/id_rsa):
/home/turby/.ssh/id_rsa already exists.
Overwrite (y/n)? n
turby@tudor:~$ cd /home/turby/.ssh
turby@tudor:~/.ssh$ ls -al
total 24
drwx------  2 turby turby 4096 Jun 26 11:07 .
drwxr-xr-x 36 turby turby 4096 Jun 27 09:17 ..
-rw-r--r--  1 turby turby  393 Jun 26 11:07 authorized_keys
-rw-------  1 turby turby 1675 Jun 25 12:55 id_rsa
-rw-r--r--  1 turby turby  393 Jun 25 12:55 id_rsa.pub
-rw-r--r--  1 turby turby  984 Jun 22 13:30 known_hosts
turby@tudor:~/.ssh$ cat .ssh/id_rsa.pub >> .ssh/authorized_keys

root@tudor:~# killall -9 sshd
root@tudor:~# /etc/init.d/ssh start
[ ok ] Starting OpenBSD Secure Shell server: sshd.
root@tudor:~#
```

We then tried to connect to our own computer and it worked, even if the connection itself was basically useless, we knew that it worked.

```
turby@tudor:/root$ ssh localhost
The authenticity of host '[localhost]:1111 ([127.0.0.1]:1111)' can't be establis
hed.
ECDSA key fingerprint is 03:4e:85:f5:23:4b:a7:85:2c:24:e4:42:29:af:79:e2.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[localhost]:1111' (ECDSA) to the list of known hosts
.
Linux tudor 3.2.0-2-686-pae #1 SMP Mon Jun 11 18:27:04 UTC 2012 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jun 26 11:27:18 2012 from localhost
turby@tudor:~$
```

3. After that we sent the public key to the target computer via scp, an ssh based command used for sending files.

```
root@tudor:/home/turby# scp -P22 /home/turby/.ssh/id_rsa.pub root@192.168.0.122:
/home/turby/.ssh/authorized_keys
```

We had to use a special parameter (-P22) to change the Port because it was different from our current one. It would have been easier to change the target computer's port beforehand, but we did it afterwards, using the same commands as in step 1.

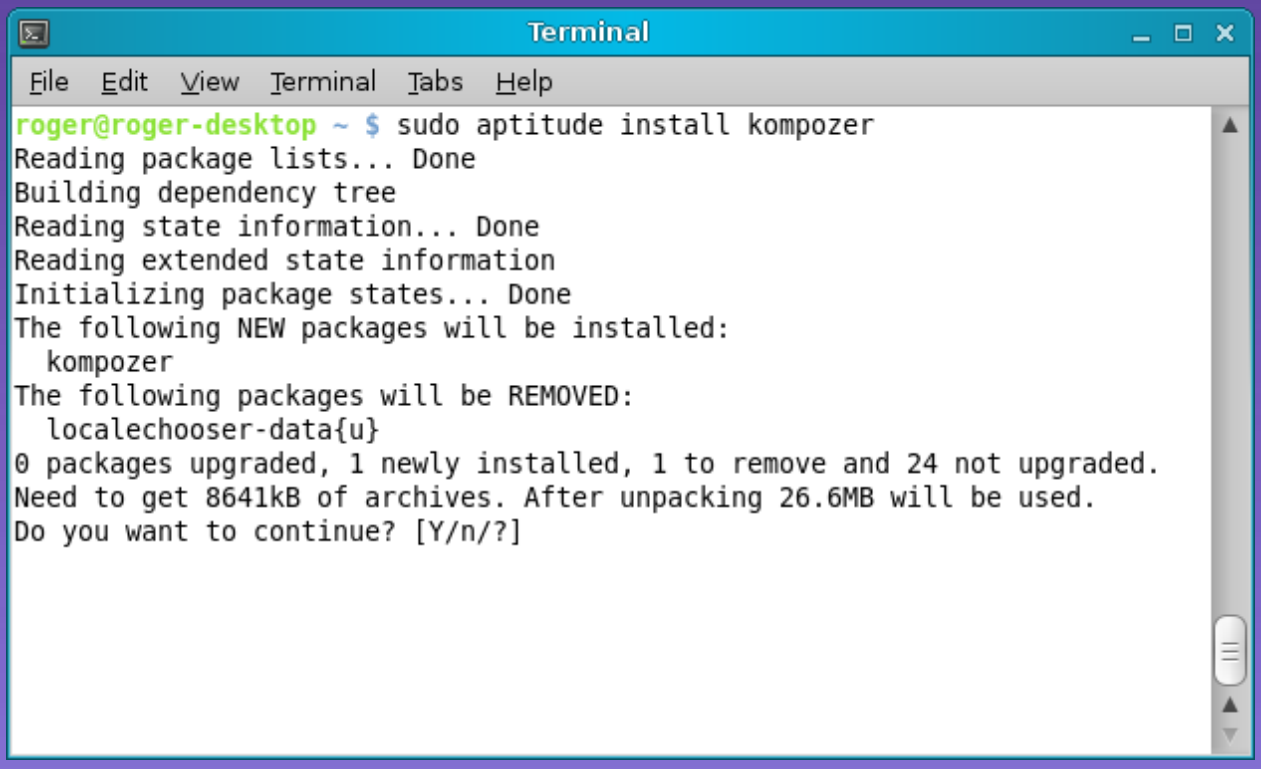
```
root@tudor:/etc/ssh# vim /etc/ssh/ssh_config && vim /etc/ssh/sshd_config
root@tudor:~# killall -9 sshd
root@tudor:~# /etc/init.d/ssh start
[ ok ] Starting OpenBSD Secure Shell server: sshd.
root@tudor:~#
```

4. Having finished everything, we tried the connection and it finally worked, after much trial and error (of course, many mistakes were made during the configuration).

```
root@tudor:/home/turby# ssh turby@192.168.0.122
Last login: Tue Jun 26 15:58:21 CEST 2012 from 192.168.0.206 on pts/1
turby@Tudorel ~ $
```

Aptitude command

Aptitude is the menu-driven package manager for Debian-based Linux distributions (including Ubuntu and other derivatives).It allows me to view the list of packages and to perform package management tasks such as installing, upgrading or removing packages. All this needs is a network connection for downloading the packages. Firstly I didn't know anything about Linux commands especially aptitude. Then I found out that aptitude is one of the most important commands because you can add almost everything to your system by using it.



Command-line actions:

Install command will Install one or more packages. The packages should be listed after the "install" command; if a package name contains a '~', it will be treated as a search pattern and every package matching the pattern will be installed. The "install" command will modify aptitude's stored information about what actions to perform. Therefore, if you issue the command "aptitude install squid" and then abort the installation, you will need to run "aptitude remove squid" to cancel that order.

Example: aptitude install games; aptitude install vim etc.

Search command searches for packages matching one or more expressions. The expressions should be listed after the "search" command. All packages matching the given expressions will be

displayed. As usual, this may be a full search expression; for instance, "aptitude search '~N'" will list all "new" packages.
Example: aptitude search squid; aptitude search iTunes etc.

Remove, purge, hold commands are the same as "install", but apply the named action to all packages unless specified otherwise. For example the command "aptitude remove 'x'" will remove all packages whose name contains 'x'.

Clean command removes all downloaded and cached packages

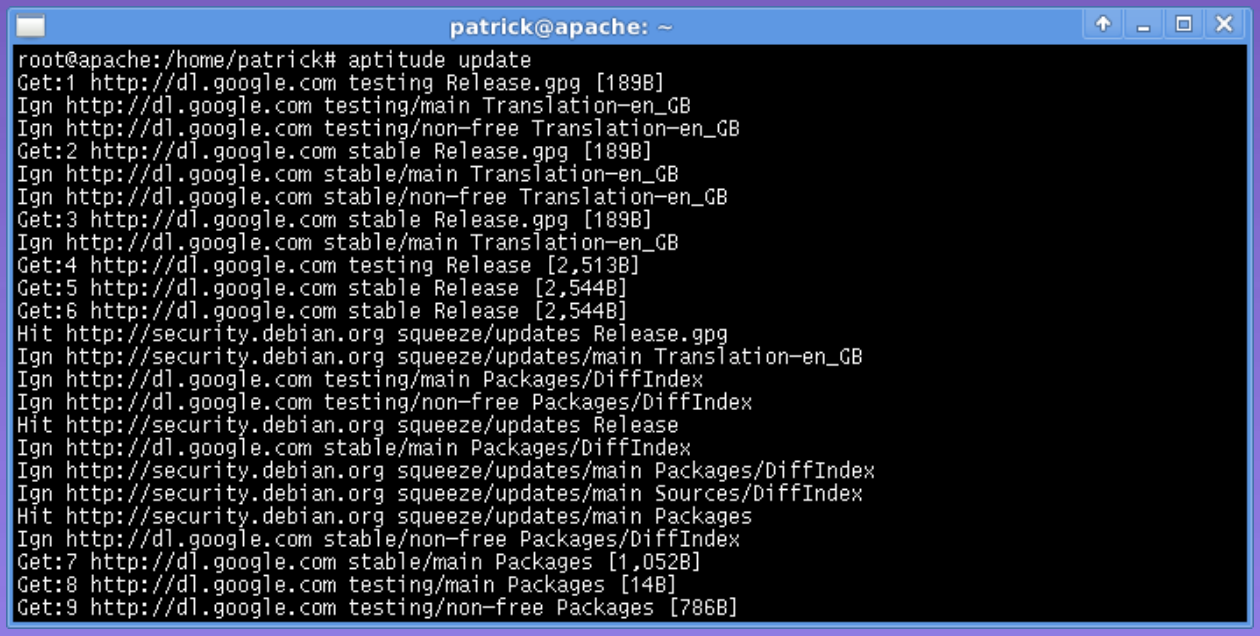
Markauto, unmarkauto mark packages as automatically installed or manually installed, respectively. You may specify packages using the same syntax as before, including specifying actions to be performed. For example "aptitude markauto 'libs'" will mark all packages in the 'libs' section as automatically installed.

Very important: aptitude update and aptitude upgrade!

Aptitude update updates the list of available packages from the master service

Aptitude upgrade upgrades installed packages to their most recent version. Installed packages will not be removed unless they are unused; packages which are not currently installed will not be installed.

Note: To have the latest versions of your applications, you should update and upgrade daily.

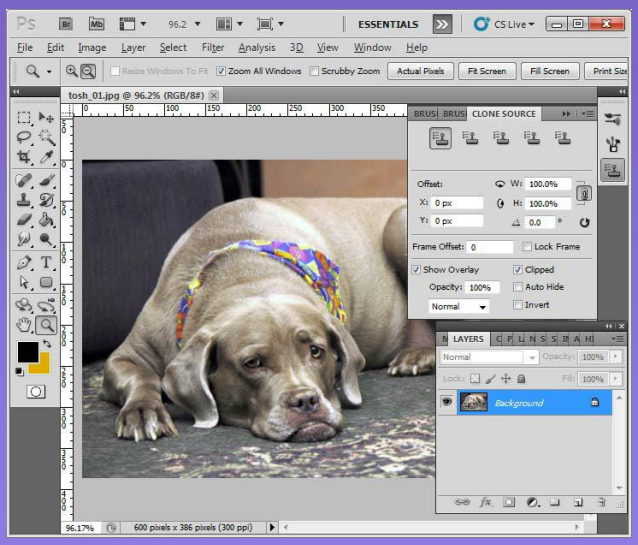


Chapter VI – LINUX vs. WINDOWS

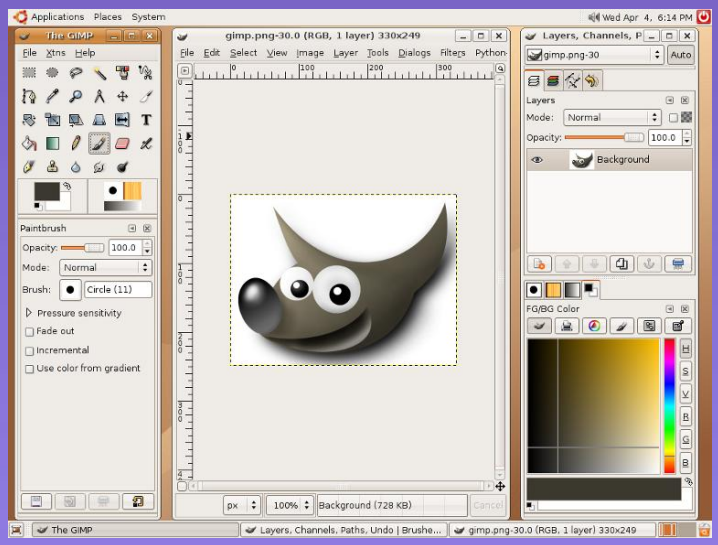
After spending 3 weeks working almost every day with Linux I had to face a bigger challenge: how can I use what I had learned when most of the computers around me are using Windows OS. The solution seemed simple: to use the alternative programs that exist on Linux: GIMP and the OpenSource package. However, I encountered different problems, starting from compatibility issues to people's skepticism regarding Linux.

1. GIMP (The GNU Image Manipulator Program)
– the alternative to Adobe Photoshop –

As a regular user of Adobe Photoshop during the last few years I was pleasantly surprised because I didn't need more than 5 minutes to get used to it. The design, as shown below, is very similar so that I could easily find everything I wanted to use. Later, I found that GIMP lacks some of the advanced features of Adobe Photoshop that fortunately I almost never used. However this would be a major disadvantage for a professional user such as a graphic designer who needs a certain file format in order to be able to adequately print his work.



Adobe Photoshop CS5, worth 699$



GIMP, free license

All in all I found this 25 Mb program suitable for my needs – also because of its portability: I could have it on my USB and use it on every computer I wanted to. By choosing GIMP I preferred work over money because 25Mb means that sometimes I have to work harder in order to achieve the same result as in Photoshop.

In conclusion, instead of pirating Photoshop, it's better for a student to use a free licensed software like GIMP in order to achieve comparable results when editing images.

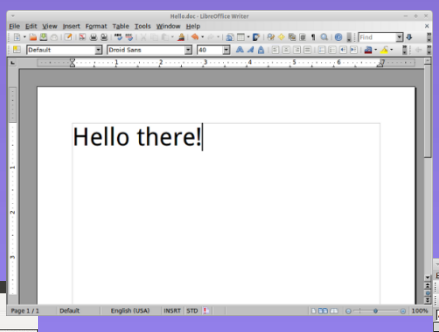
2. OpenOffice – an alternative to Microsoft Office –

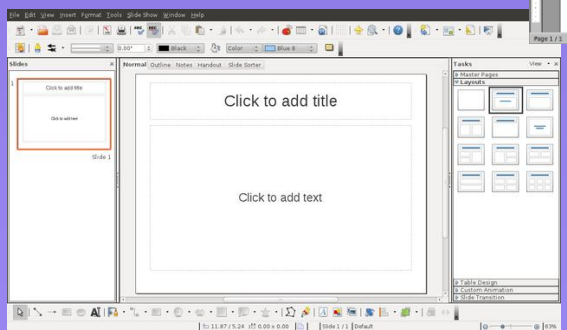
Everybody I know uses Microsoft Office for creating various documents, from presentations to databases. But what happens when you are using Linux? The answer: you start using OpenOffice, also known as the LibreOffice.

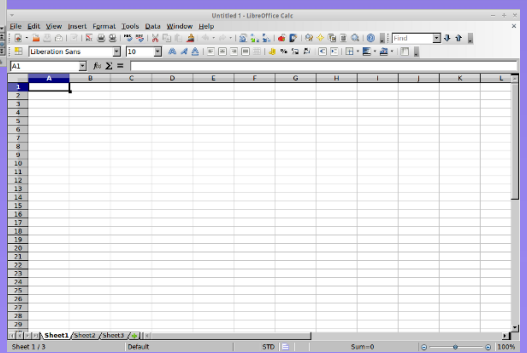
At first, everything seemed different except the design. It presented the same buttons and windows, but it had other names for the same features used until then by Windows. For example, the font's names were completely new for me: Liberation Sans, Déjà vu Sans. I was nicely surprised to see that when I typed the name of the font I used on Windows, OpenOffice found it and I was able to use it without having to install it. Overall, I got along really well with this program after a short period.

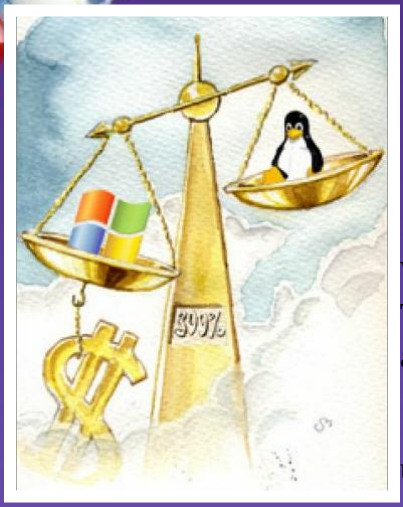
Hoping I could use OpenOffice when making homework together with my Windows-using partner, I tried the Impress program, the alternative to Microsoft Office PowerPoint. I had a problem: it looked great on my computer and completely chaotic on others. Happily, the computers from school are running a minimal version of Ubuntu, so our presentation looked great.

My personal opinion: OpenOffice is an extremely useful package, almost vital for a student. It's only drawback is that computers running Microsoft Office can't see it correctly, but if the others are using Linux – it's definitely worth learning how to use it.









Advantages of using Linux

Although Linux it's not as popular with as many people as Windows, what makes users choose a completely new operating system? The answer is simple: it offers a certain series of advantages that cannot and will never be offered by Windows.

Below, you have a table of both advantages and disadvantages of using Linux instead of Windows.

Topic	Linux	Windows
Open Source	Linux operating systems allow users to read and modify the source code.	Users are not allowed to enter the source code.
Price	Most versions of Linux are and will remain free.	Microsoft Windows' license costs around 100$.
Ease	In order to efficiently use Linux, the user must know the basic commands.	For a new user, Windows is still the easiest choice.
Security	Linux is and has always been a very secure system.	Most viruses, malware and other dangerous programs are designed for Windows OS.
Hardware	Most companies don't include Linux support for their devices.	Because it is the most popular, all the companies make their devices compatible with Windows.
Support	Support is given by a large community of users, but sometimes it can be very slow.	Windows has its own help but it can also provide paid but fast support.
Flexibility	The user can configure the system to his own needs.	In Windows you are stuck with what they think it's good for you.

The trump card of Linux OS is his philosophy which states that "a file is always a file". This means that every file, despite its role, can be and is encouraged to be edited. This way the user can modify the operating system itself, not only the installed programs as in Windows, and make it suitable for what they need. For example, on Linux I can choose between at least three desktops (GNOME, KDE and Enlightenment) while on Windows I can only sort the shortcuts and add a nice photo as a background (and maybe play with the screensaver).

On the other hand is compatibility. Windows does not bother to create an OS which is compatible with Linux because they have the majority on the market. This popularity leads major companies to create hardware mostly for Windows: you can have an unpleasant surprise if you run Linux and buy a Win piece of device. Also, most software are designed for Windows. Small companies can't afford to design a product which is expected to have only a few users.

An incredibly important advantage for teenagers is that you can browse the Internet without having to worry about downloading unwanted programs, such as viruses, because every change of the system requires root permission so the user will always have to agree on any modification of the operating system. This is the reason I chose to install Linux on my computer: go on the Internet on every site I want, with satisfying speed and no risks.

Linux OS is definitely not a "mere" operating system. It means much more than that. It isn't a weapon against Windows, it's not only a free thing. Linux is about freedom: freedom to share, freedom to modify, freedom to help others. Linux means friendship, devotion. If you want to be in the driver's seat and the OS only to hand you the keys, you'll have to devote your time to learning how to operate on Linux. The result will be worth it, because you'll have an OS that you can make sit up and dance.