



Lifelong
Learning
Programme

Programul Leonardo da Vinci
Proiectul LLP-LdV/IVT/2011/RO/188:
Inițiere profesională în Linux – Let's WIN LIN!



Let's WIN LIN



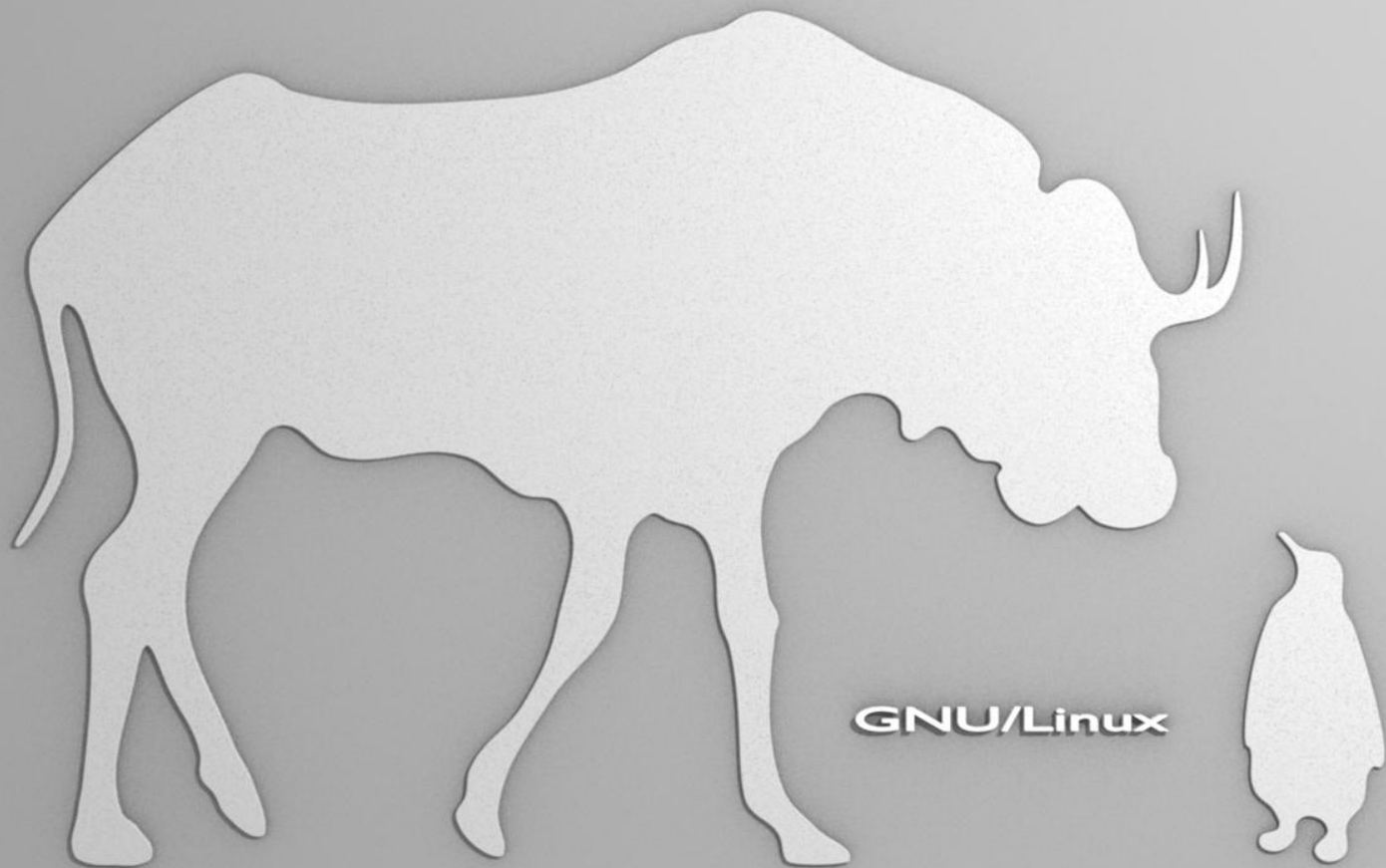
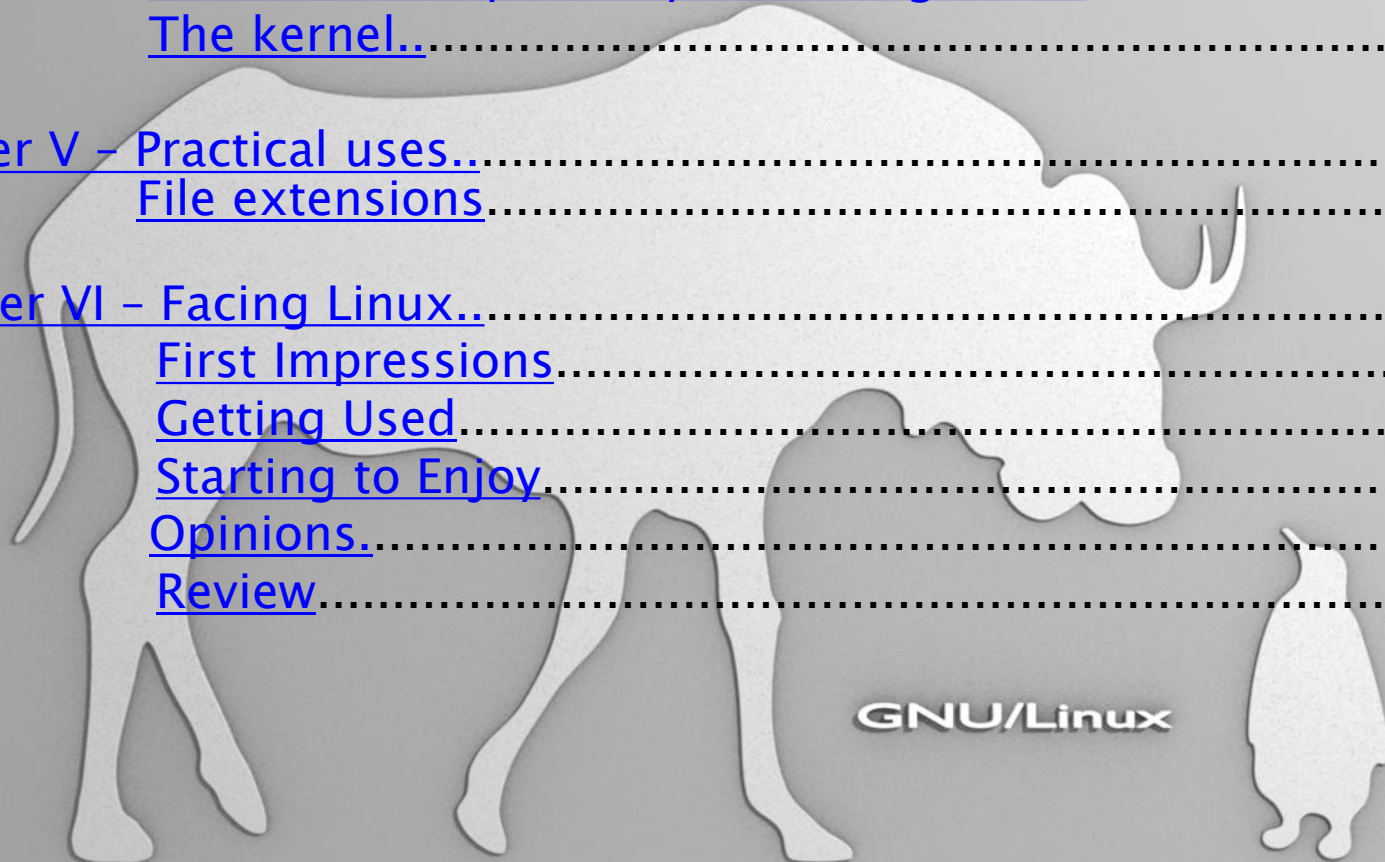


Table of contents

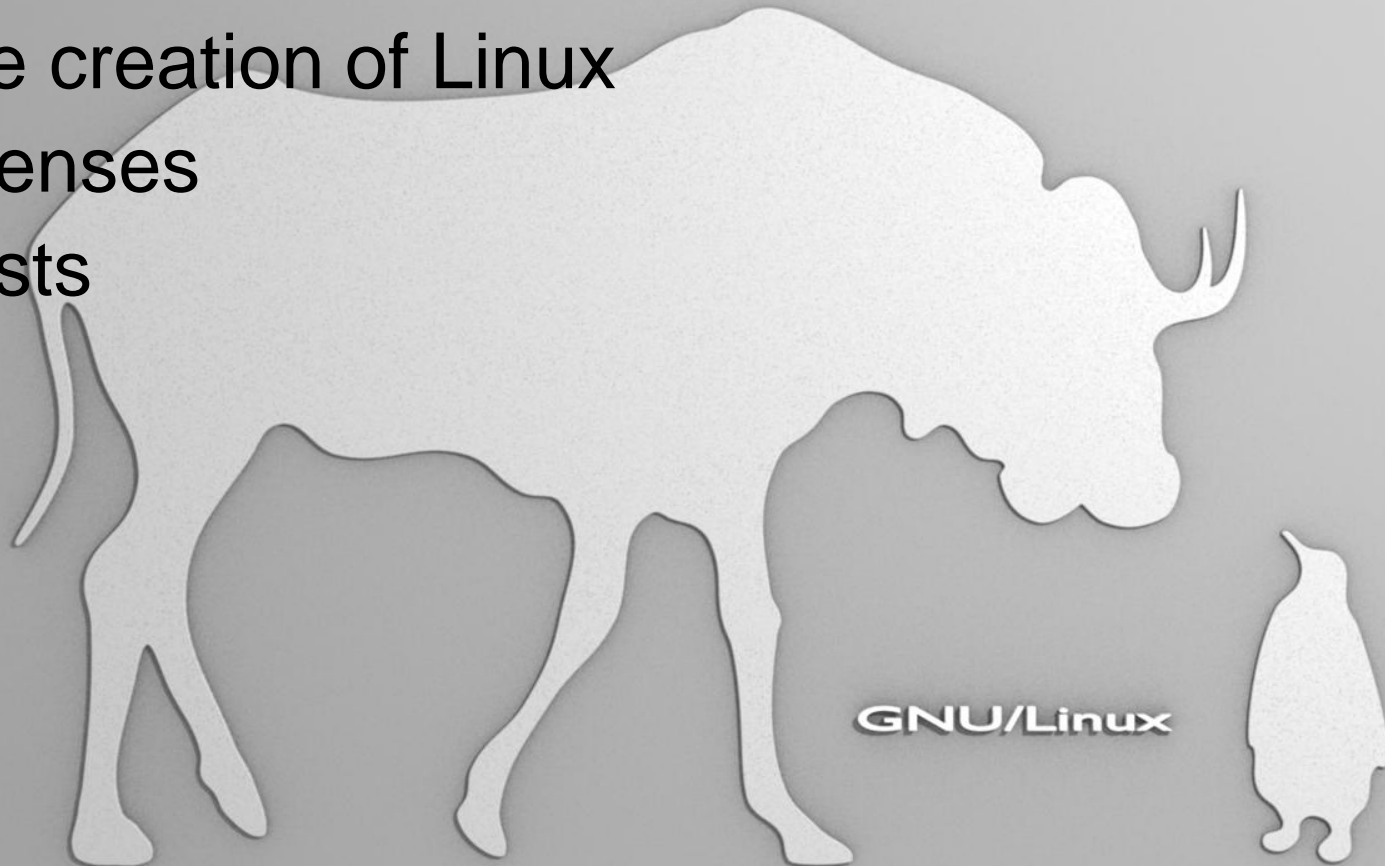
Title	Page
Chapter I – History of Linux	5
GNU/Linux naming controversy	9
Open Source and Linux Foundation	10
BSD License	16
Costs	18
Chapter II – Linux a multiuser system	21
TTYs	23
Group management	26
Virtual Machines	30
Chapter III – Installation and applications	35
Linux Installation	36
Linux Distributions	37
Installation of applications	42
Window managers and desktop environments	47
Program equivalents	51

Chapter IV – Connecting to Linux	57
About Hardware	58
Connecting a device to the PC...	60
Hardware compatibility and recognition.	61
The kernel..	64
 Chapter V – Practical uses..	71
File extensions	80
 Chapter VI – Facing Linux..	81
First Impressions	83
Getting Used	85
Starting to Enjoy	86
Opinions.	89
Review	90



Chapter I – History of Linux

- The creation of Linux
- Licenses
- Costs



Events leading to the creation of Linux

It all started in 1969 when Ken Thompson and Dennis Ritchie, both from AT&T Bell Laboratories conceived and implemented the Unix operating system. It was first released in 1970, and was widely adopted, copied and modified by academic institutions and businesses, because of its availability and portability. The authors of other systems were strongly influenced by the design of this operating system.



GNU/Linux

In 1983, Richard Stallman started the GNU project with the goal of creating a free UNIX-like operating system. As part of this work, he wrote the GNU General Public License (GPL). By the early 1990s there was almost enough available software to create a full operating system. Because of the failure of the GNU kernel(Hurd) to attract enough attention from developers, the GNU project was left incomplete.

Another free operating system project, initially released in 1977, was the Berkeley Software Distribution (BSD). In 1986, Maurice J. Bach, of AT&T Bell Labs, published *The Design of the UNIX Operating System*. This definitive description principally covered the System V Release 2 kernel, with some new features from Release 3 and BSD.

These factors and the lack of a widely adopted, free kernel provided the impetus for Torvalds's starting his project. Linus was convinced that if either the GNU or 386BSD kernels were available at the time, he surely would not have written his own.



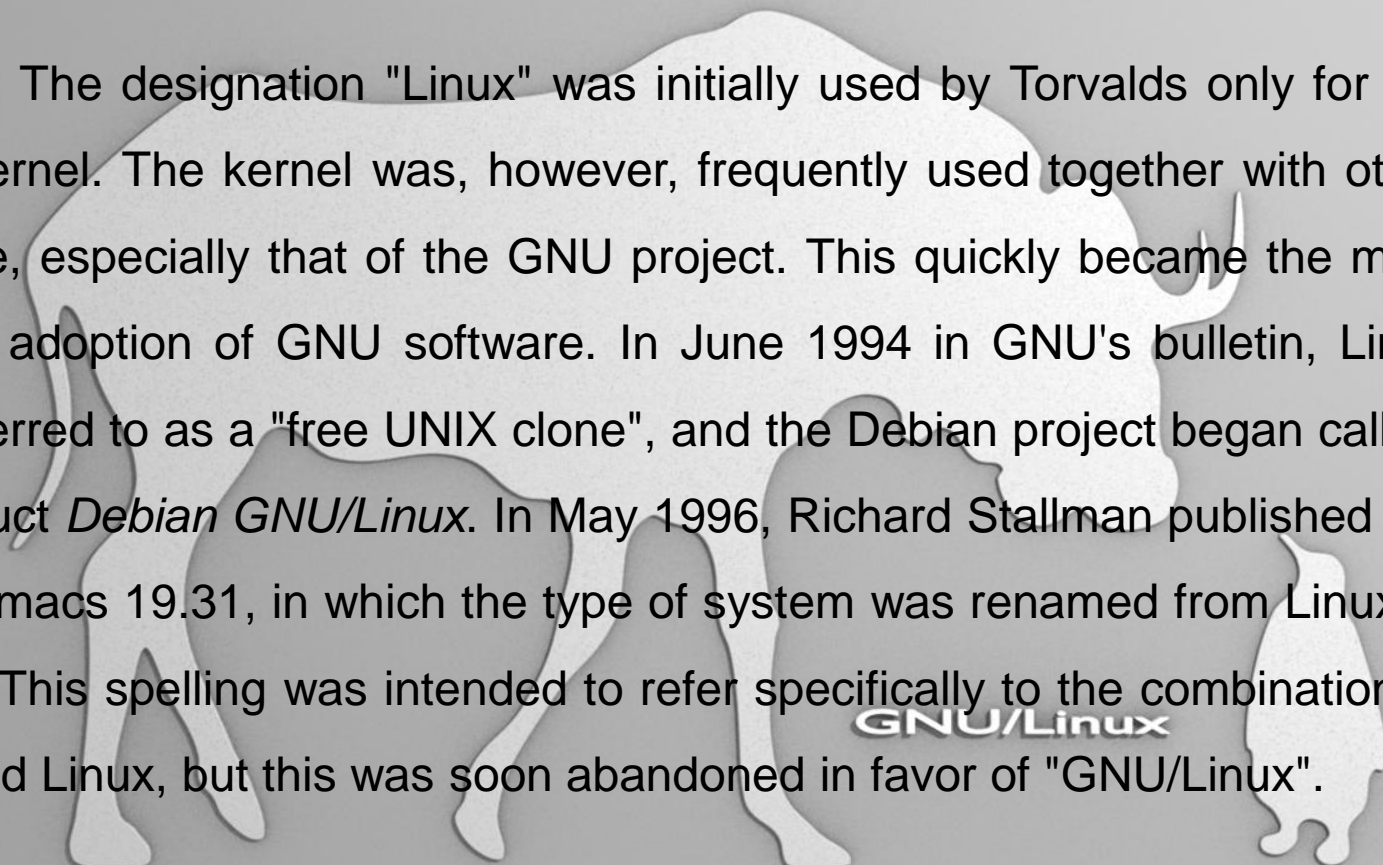
GNU/Linux

The creation of Linux

In 1991, in Helsinki, Linus Torvalds began a project that later became the Linux kernel. It was initially a terminal emulator, which Torvalds used to access the large UNIX servers of the university. He wrote the program specifically for the hardware he was using and independent of an operating system because he wanted to use the functions of his new PC with an 80386 processor. Development was done on MINIX using the GNU C compiler, which is still the main choice for compiling Linux today (although the code can be built with other compilers, such as the Intel C Compiler).



GNU/Linux naming controversy

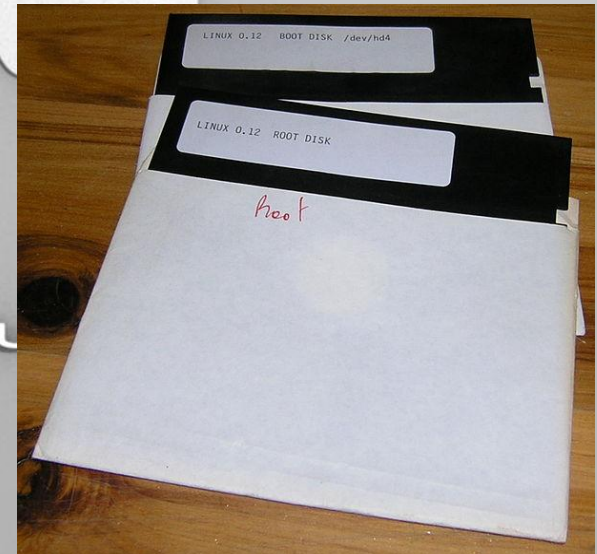


The designation "Linux" was initially used by Torvalds only for the Linux kernel. The kernel was, however, frequently used together with other software, especially that of the GNU project. This quickly became the most popular adoption of GNU software. In June 1994 in GNU's bulletin, Linux was referred to as a "free UNIX clone", and the Debian project began calling its product *Debian GNU/Linux*. In May 1996, Richard Stallman published the editor Emacs 19.31, in which the type of system was renamed from Linux to Lignux. This spelling was intended to refer specifically to the combination of GNU and Linux, but this was soon abandoned in favor of "GNU/Linux".

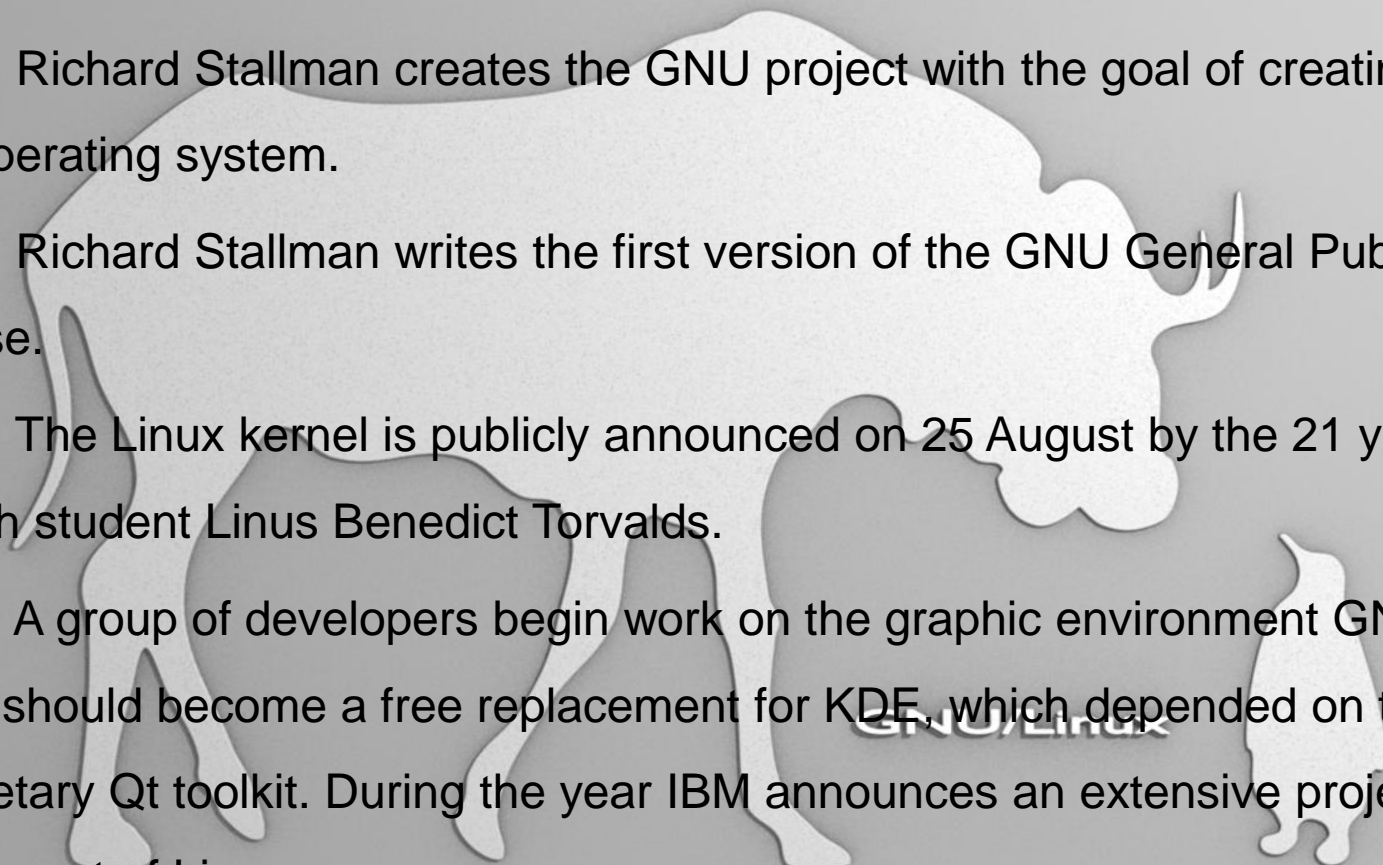
Open Source Development Lab and Linux Foundation

The Open Source Development Lab (OSDL) was created in the year 2000, and is an independent nonprofit organization which pursues the goal of optimizing Linux for employment in data centers and in the carrier range. It served as sponsored working premises for Linus Torvalds and also for Andrew Morton (until the middle of 2006 when Morton transferred to Google). Torvalds works full-time on behalf of OSDL, developing the Linux Kernels.

On January 22, 2007, OSDL and the Free Standards Group merged to form The Linux Foundation, narrowing their respective focuses to that of promoting GNU/Linux in competition with Microsoft Windows.



Short chronology

- 1983: Richard Stallman creates the GNU project with the goal of creating a free operating system.
 - 1989: Richard Stallman writes the first version of the GNU General Public License.
 - 1991: The Linux kernel is publicly announced on 25 August by the 21 year old Finnish student Linus Benedict Torvalds.
 - 1999: A group of developers begin work on the graphic environment GNOME, which should become a free replacement for KDE, which depended on the then proprietary Qt toolkit. During the year IBM announces an extensive project for the support of Linux.
- 

GNU General Public License

A free software license differs from a usual software license by granting recipients of the software rights prohibited by copyright law, rights like modifying or redistributing

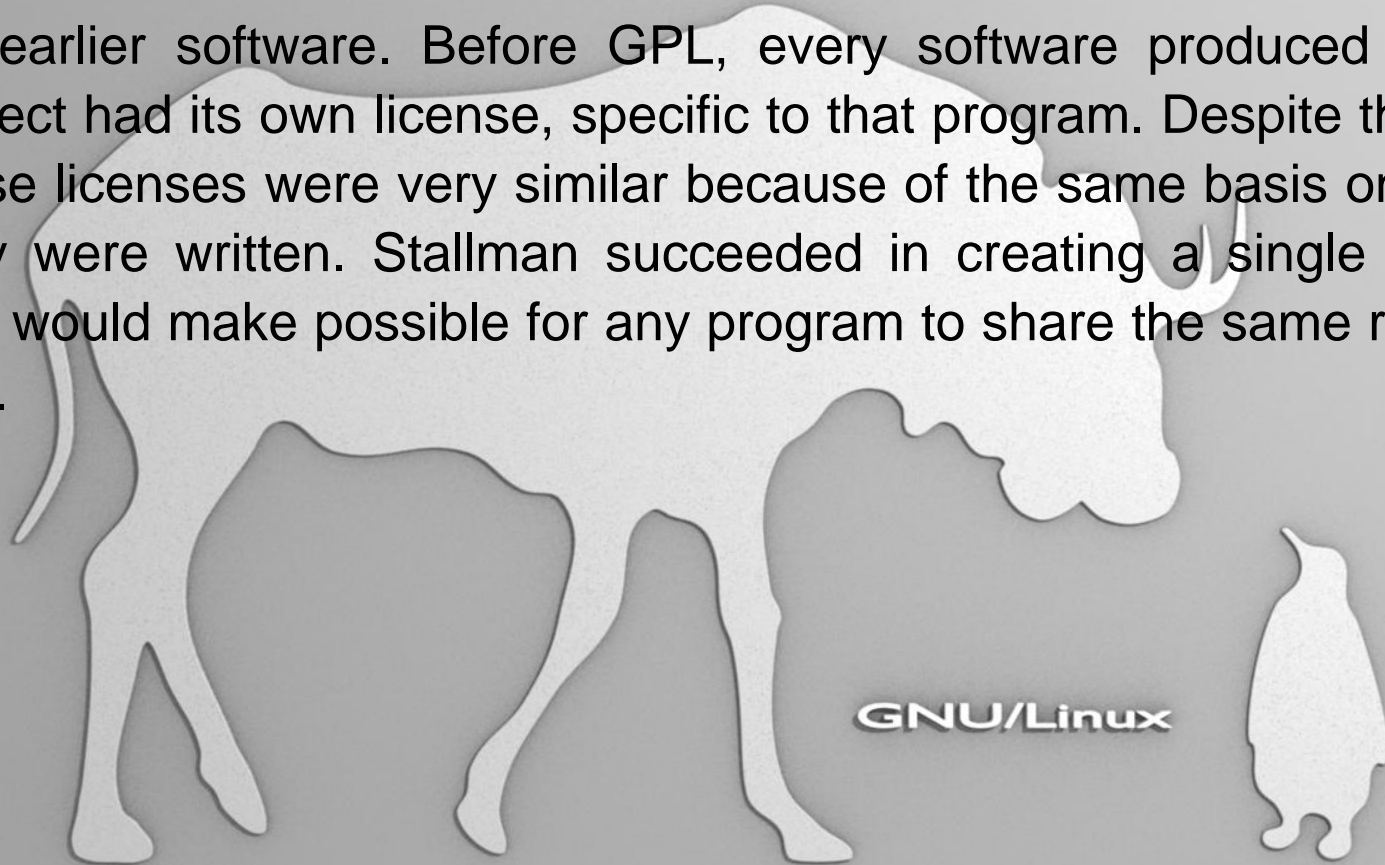


In 1989, Richard Stallman created the GNU General Public License (GPL) for the use of the project. Today it is the most popular free software license

GNU/Linux

GPL allows recipients to distribute or modify works, but with the requirement of preserving these rights for the following users. Creators of this practicing called it “copyleft”, because it uses copyright law to offer these rights.

At first GPL was a combination of licenses created and used by GNU for earlier software. Before GPL, every software produced by the project had its own license, specific to that program. Despite this fact, these licenses were very similar because of the same basis on which they were written. Stallman succeeded in creating a single license that would make possible for any program to share the same rights of use.



Terms and conditions

Every user of a software that is under the GPL license has the right to be aware of the terms and conditions of the GPL. Any recipient that doesn't disobey the GPL main license has the rights stipulated in the GPL. Moreover, the recipient is allowed to put any price on the program, if he wants to sell it. This fact makes GPL different from other licenses, which prohibit commercial use.

The GPL doesn't allow a recipient to impose “further restrictions on the rights granted by the GPL”. This is one of the reasons why distribution under a non-disclosure agreement is not allowed. Also, when someone distributes a GPL plus his/her own changes, the requirements for distributing the whole work cannot be any greater than the requirements that are in the GPL (copyleft).

Because GPL work is copyrighted, one has no right to redistribute a program, except under the terms of the GPL license. The only requirement for being able to exercise the rights normally restricted by copyright law is to adhere to the terms of the GPL. If one disobeys the terms of the GPL while distributing copies of a work, he/she can be sued by the original author under copyright law.



GNU/Linux

BSD License

BSD are a set of very permissive software licenses. The first BSD license that was written was used for the Berkeley Software Distribution, a Unix-like operating system that gave the name of the license too.

The original BSD license was created at the University of California. This first version was revised repeatedly and the resulting licenses are called BSD licenses.

BSD is more permissive than GPL, placing minimal restrictions on the redistribution of a software. This means that one can do almost anything he/she wants with the software, even considering it his/her proprietary. This is in contrast to copyleft licenses.

GNU/Linux

It was told about this license that you can "do what you want with the code as long as you include this notice, don't use our names to endorse your product, and don't sue us".



Costs

Because Linux systems are based on open source software, they provide you greater flexibility than proprietary platforms. Virtually any element of the desktop can be tailored to your specific needs. Unlike Windows, advanced configuration on Linux does not require a separate software stack that costs you additional licensing and training fees. Many people are 100% sure that Linux is genuinely free. Everybody can borrow a CD with Linux from anybody and install as many copies as they like without paying anything for licensing. Therefore, a comparison between the Linux license and the cost of a Microsoft server operating system license should have as result a \$0 for Linux to over \$800 for Windows, depending on hardware, usage and the number of clients allowed.

GNU/Linux

The officials from Microsoft still insist on the idea that Windows is cheaper than Linux. As proof, they brought again some arguments on their side to put their OS in a better view. Furthermore, they adduced an upfront cost of **\$4.000** for Windows **2000 Advanced Server** plus **\$5.000** for the **SQL Server**, in contrast to **Red Hat Enterprise Linux** at **\$2.500** for one year premium support plus **DB2** at **\$7.500**, or **Oracle** at **\$15.000**.

According to a study which examined the costs of running various operating systems over three years, the downtime and maintenance cost areas represent 75% of TCO over a 3-year period, while software acquisition costs represent 7% of TCO over that same period. The Windows technicians, however, only managed an average of 10 machines each, while Linux admins can generally handle several times that, because the network and systems management capabilities in Linux are extremely robust. Red Hat's business is based on annual subscriptions for OS support - you pay a subscription for every server, every year. And, if you want 24/7 support, you'll pay more. Even though, the cost of running Linux is roughly **40%** that of Microsoft, Linux costs **\$75.000** over three years, while a Windows deployment costs **\$190.000**.

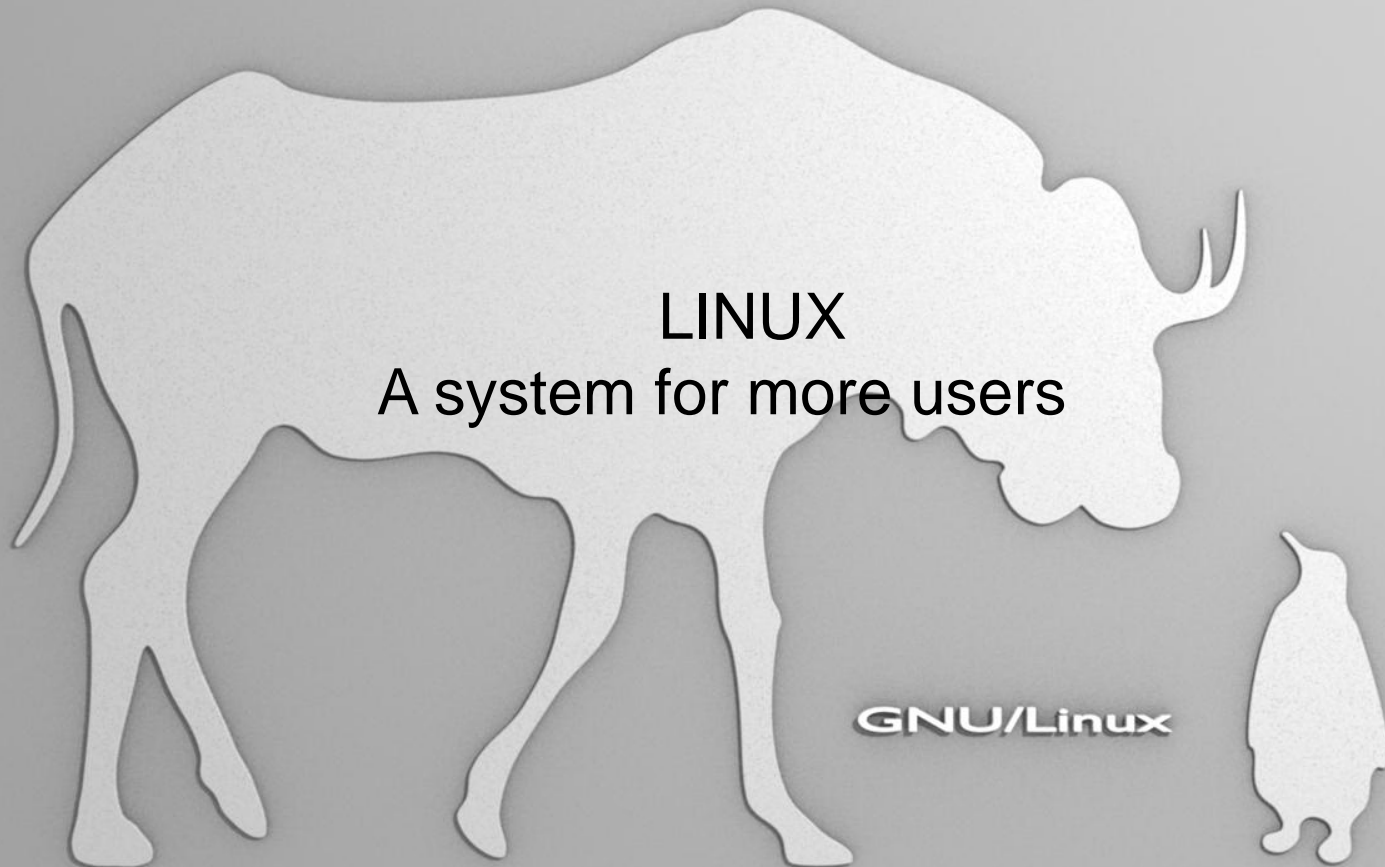
At large scale, the organizational benefits to Linux do not only come from a one-time capital cost reduction, but from long-term IT cost control coupled with an increase in IT's ability to contribute to organizational productivity. Most important, the overall robustness and stability of Linux operating systems means critical applications stay up and running - keeping your business focused on the work at hand. Investments in Linux continue to be valuable throughout the fiscal years.

Another great advantage alongside with those presented earlier is the multiuser function of the operating system.



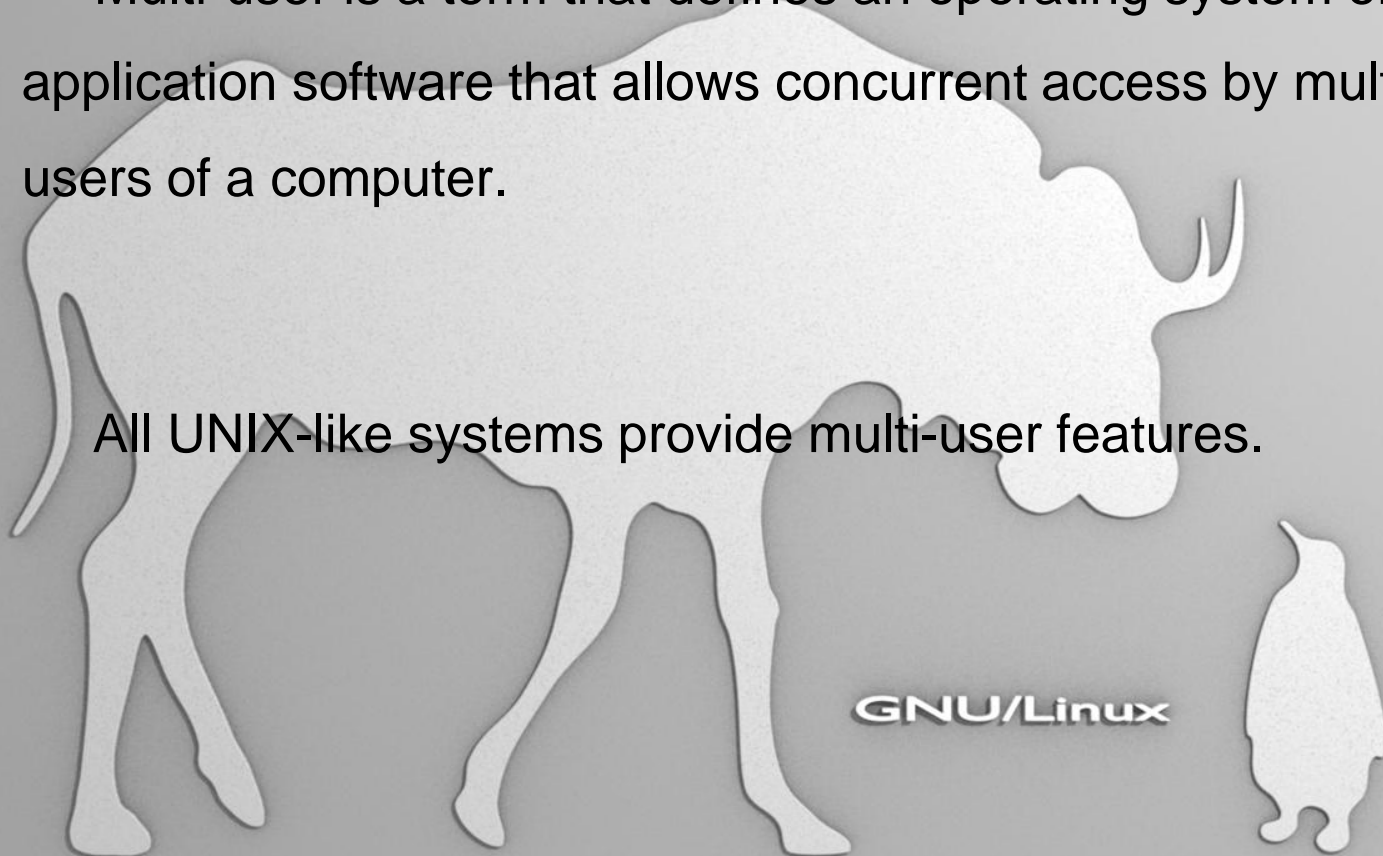
GNU/Linux

Chapter II



Multi-user is a term that defines an operating system or application software that allows concurrent access by multiple users of a computer.

All UNIX-like systems provide multi-user features.



TTYs

One of the major benefits of LINUX-based OS is their innate and intuitive multi-user capability, innate in the sense of requiring no third party software to function.

Firstly, Linux provides the capability of starting multiple terminals, or TTYs, on the same machine, which can thereby be accessed by different users without the need for a graphical interface. There are, by default, 6 TTYs; however, the root user can adjust this number, as he can do with all quantifiers in Linux in general.

GNU/Linux

```

~ : bash <2>
File Edit View Bookmarks Settings Help
-rw-r--r-- 1 buruiana buruiana 689917 Sep 7 13:46 GNUlinux_Day2.pdf
-rw-r--r-- 1 buruiana buruiana 596631 Sep 7 13:26 GNUlinux_Day3.pdf
drwx----- 2 buruiana buruiana 4096 Sep 13 08:27 .gnupg
drwxr-xr-x 2 buruiana buruiana 4096 Sep 3 14:18 .gstreamer-0.10
-rw-r--r-- 1 buruiana buruiana 152 Sep 7 10:00 .gtk-bookmarks
drwx----- 2 buruiana buruiana 4096 Aug 28 15:25 .gvfs
-rw----- 1 buruiana buruiana 2970 Sep 7 10:00 .ICEauthority
-rw-r--r-- 1 buruiana buruiana 52711 Sep 10 10:13 index.html
-rw-r--r-- 1 libvirt-qemu kvm 171642880 Sep 4 08:49 install-amd64-minimal-20120621.iso
-rwxr-xr-x 1 buruiana buruiana 1560728 Sep 12 08:58 installer.exe
-rw-r--r-- 1 buruiana buruiana
-rw-r--r-- 1 buruiana buruiana
drwxr-xr-x 2 buruiana buruiana
drwxr-xr-x 2 buruiana buruiana
drwx----- 4 buruiana buruiana
-rw-r--r-- 1 buruiana buruiana
-rw-r--r-- 1 buruiana buruiana
-rw-r--r-- 1 buruiana buruiana
drwxr-xr-x 3 buruiana buruiana
-rw-r--r-- 1 buruiana buruiana
-rwxrwxrwx 1 buruiana buruiana
drwx----- 3 buruiana buruiana
drwx----- 4 buruiana buruiana
drwxr-xr-x 2 buruiana buruiana
-rw----- 1 buruiana buruiana
drwxrwxr-x 8 buruiana buruiana
drwxr-xr-x 2 buruiana buruiana
drwx----- 3 buruiana buruiana
-rw-r--r-- 1 buruiana buruiana
drwxr-xr-x 2 buruiana buruiana
drwx----- 2 buruiana buruiana
-rw----- 1 buruiana buruiana
-rw-r--r-- 1 buruiana buruiana
-rw-r--r-- 1 buruiana buruiana
-rw-r--r-- 1 buruiana buruiana
drwxr-xr-x 3 buruiana buruiana
-rw-r--r-- 1 buruiana buruiana
drwxr-xr-x 2 buruiana buruiana
drwxr-xr-x 2 buruiana buruiana
-rw-r--r-- 1 buruiana buruiana
drwxr-xr-x 1 buruiana buruiana
-rw-r--r-- 1 buruiana buruiana
-rw-r--r-- 1 buruiana buruiana
drwx----- 5 buruiana buruiana
-rw-r--r-- 1 libvirt-qemu kvm 171642880 Sep 4 08:49 install-amd64-minimal-20120621.iso
drwxr-xr-x 5 buruiana buruiana
-rw-r--r-- 1 buruiana buruiana
drwxr-xr-x 2 buruiana buruiana
-rw----- 1 buruiana buruiana
drwxr-xr-x 2 buruiana buruiana
-rw-r--r-- 1 buruiana buruiana
-rw-r--r-- 1 buruiana buruiana
drwx----- 2 buruiana buruiana
drwx----- 2 buruiana buruiana
buruiana@Sebastian:~$

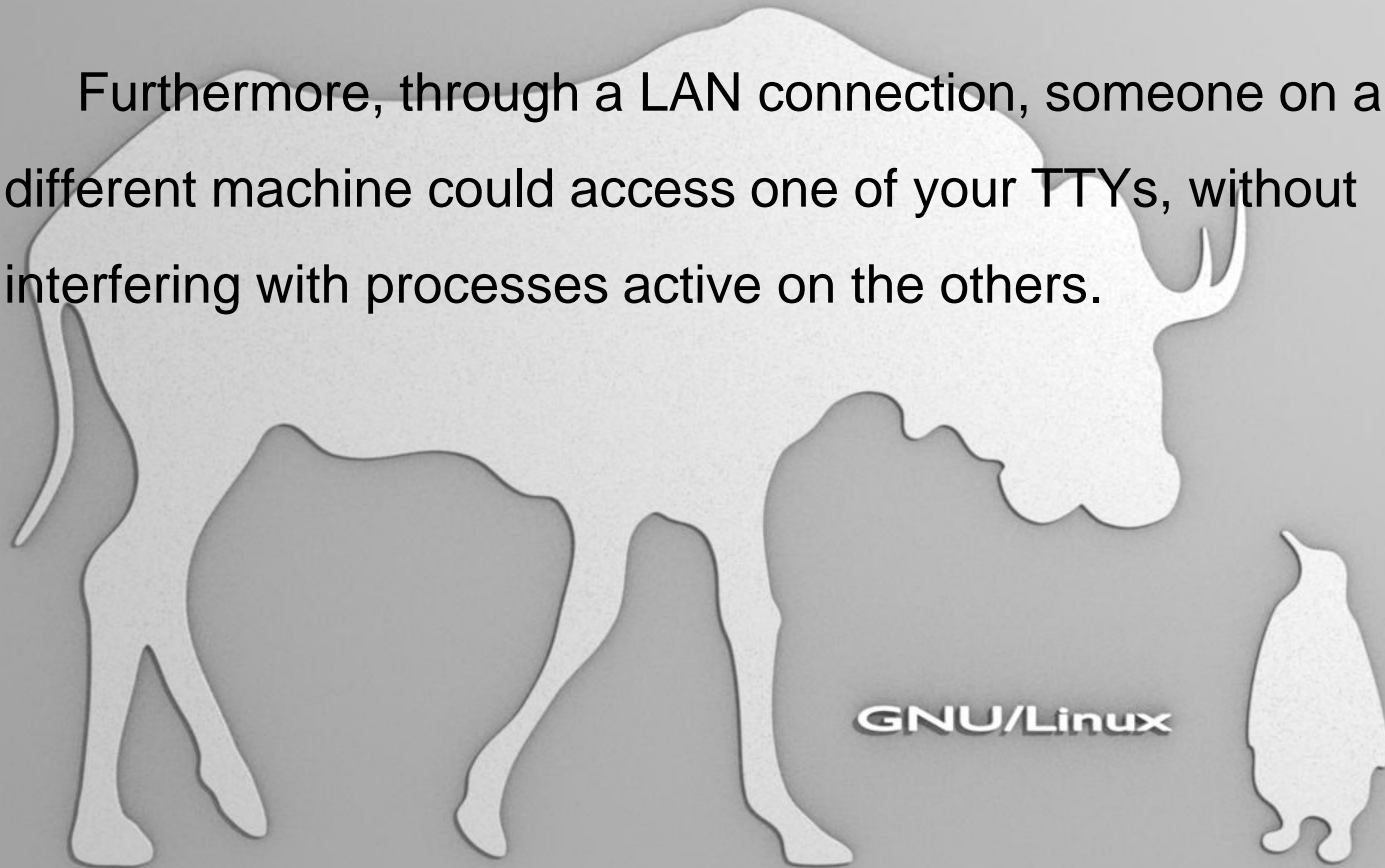
~ : htop
File Edit View Bookmarks Settings Help
1 [|||||] Tasks: 126, 318 thr: 2 running
2 [|||||] Load average: 0.77 0.68 0.51
Mem[|||||] Uptime: 04:47:31
Swp[|||||]

CPU: 5.4%
CPU: 9.3%
Mem: 920/1929MB
Swp: 19/3925MB

PID USER PPID PRI NI VSZ RSS TIO S CPU MEM% TIO S Command
16760 buruiana 20 0 378M 38116 10576 S 6.0 1.9 0:21.05 /usr/bin/gtk-gnash -u http://s.ytimg.com/yts/sw/bin/cp... www.youtube.c
3164 root 20 0 23116 2612 984 S 2.0 0.1 5:20.06 /usr/bin/htop d l
2353 root 20 0 215M 40836 29580 S 2.0 2.1 3:08.31 /usr/bin/Xorg :0 -br -verbose -novtswitch -auth /var/run/gdm3... or-Debian-gdm-L5h9ua/dat
3347 buruiana 20 0 670M 37096 10020 S 1.0 1.9 2:12.71 kwin -session 10134d9cbe2000134623078700000029100000_1347004754... 70
18941 buruiana 20 0 24552 3032 1380 R 0.0 0.2 0:00.43 htop
3520 buruiana 20 0 961M 75744 21364 S 0.0 3.8 0:36.99 /usr/bin/plasma-desktop
4903 buruiana 20 0 485M 41456 20684 S 0.0 2.1 0:35.82 /usr/bin/konsole
1 root 20 0 10636 624 580 S 0.0 0.0 0:00.63 init [2]
349 root 20 0 21820 1484 556 S 0.0 0.1 0:00.12 udevd --daemon
1739 root 20 0 18956 580 500 S 0.0 0.0 0:00.02 /sbin/rpcbind -w
1770 statd 20 0 23328 604 600 S 0.0 0.0 0:00.00 /sbin/rpc.statd
1784 root 20 0 25276 136 136 S 0.0 0.0 0:00.00 /usr/sbin/rpc.idmapd
2135 root 20 0 53028 700 640 S 0.0 0.0 0:00.02 /usr/sbin/rsyslogd -c5
2136 root 20 0 53028 700 640 S 0.0 0.0 0:00.01 /usr/sbin/rsyslogd -c5
2137 root 20 0 53028 700 640 S 0.0 0.0 0:00.00 /usr/sbin/rsyslogd -c5
2116 root 20 0 53028 700 640 S 0.0 0.0 0:00.05 /usr/sbin/rsyslogd -c5
2178 root 20 0 4100 432 404 S 0.0 0.0 0:00.54 /usr/sbin/acpid
2199 dirnmgr 20 0 17164 548 472 S 0.0 0.0 0:01.18 /usr/bin/dirnmgr --daemon --sh
2221 messagebu 20 0 30916 1940 792 S 0.0 0.1 0:00.57 /usr/bin/dbus-daemon --system
2352 root 20 0 83820 1520 1320 S 0.0 0.1 0:00.00 /usr/sbin/gdm3
2320 root 20 0 83820 1520 1320 S 0.0 0.1 0:00.02 /usr/sbin/gdm3
1 root 20 0 166M 1528 1500 S 0.0 0.1 0:00.00 /usr/lib/gdm3/gdm-simple-slave --display-id /org/gnome/DisplayManager/Display1
2351 root 20 0 166M 1528 1500 S 0.0 0.1 0:00.02 /usr/lib/gdm3/gdm-simple-slave --display-id /org/gnome/DisplayManager/Display1
2355 root 20 0 71836 816 768 S 0.0 0.0 0:00.45 /usr/sbin/apache2 -k start
2358 www-data 20 0 71568 348 336 S 0.0 0.0 0:00.00 /usr/sbin/apache2 -k start
2366 www-data 20 0 288M 248 244 S 0.0 0.0 0:00.00 /usr/sbin/apache2 -k start
2367 www-data 20 0 288M 248 244 S 0.0 0.0 0:00.00 /usr/sbin/apache2 -k start
2368 www-data 20 0 288M 248 244 S 0.0 0.0 0:00.00 /usr/sbin/apache2 -k start
2369 www-data 20 0 288M 248 244 S 0.0 0.0 0:00.00 /usr/sbin/apache2 -k start
2397 www-data 20 0 288M 248 244 S 0.0 0.0 0:00.00 /usr/sbin/apache2 -k start
2398 www-data 20 0 288M 248 244 S 0.0 0.0 0:00.00 /usr/sbin/apache2 -k start
2399 www-data 20 0 288M 248 244 S 0.0 0.0 0:00.00 /usr/sbin/apache2 -k start
2400 www-data 20 0 288M 248 244 S 0.0 0.0 0:00.00 /usr/sbin/apache2 -k start
2401 www-data 20 0 288M 248 244 S 0.0 0.0 0:00.00 /usr/sbin/apache2 -k start
2402 www-data 20 0 288M 248 244 S 0.0 0.0 0:00.00 /usr/sbin/apache2 -k start
2403 www-data 20 0 288M 248 244 S 0.0 0.0 0:00.00 /usr/sbin/apache2 -k start
2404 www-data 20 0 288M 248 244 S 0.0 0.0 0:00.00 /usr/sbin/apache2 -k start
2405 www-data 20 0 288M 248 244 S 0.0 0.0 0:00.00 /usr/sbin/apache2 -k start
2406 www-data 20 0 288M 248 244 S 0.0 0.0 0:00.00 /usr/sbin/apache2 -k start
2407 www-data 20 0 288M 248 244 S 0.0 0.0 0:00.00 /usr/sbin/apache2 -k start
2408 www-data 20 0 288M 248 244 S 0.0 0.0 0:00.00 /usr/sbin/apache2 -k start
2409 www-data 20 0 288M 248 244 S 0.0 0.0 0:00.00 /usr/sbin/apache2 -k start
2410 www-data 20 0 288M 248 244 S 0.0 0.0 0:00.00 /usr/sbin/apache2 -k start
2411 www-data 20 0 288M 248 244 S 0.0 0.0 0:00.00 /usr/sbin/apache2 -k start
F1Help F2Setup F3Search F4Filter F5Free F6Sort F7Nice F8Nice F9Kill F10Quit

```

Furthermore, through a LAN connection, someone on a different machine could access one of your TTYs, without interfering with processes active on the others.



Group management

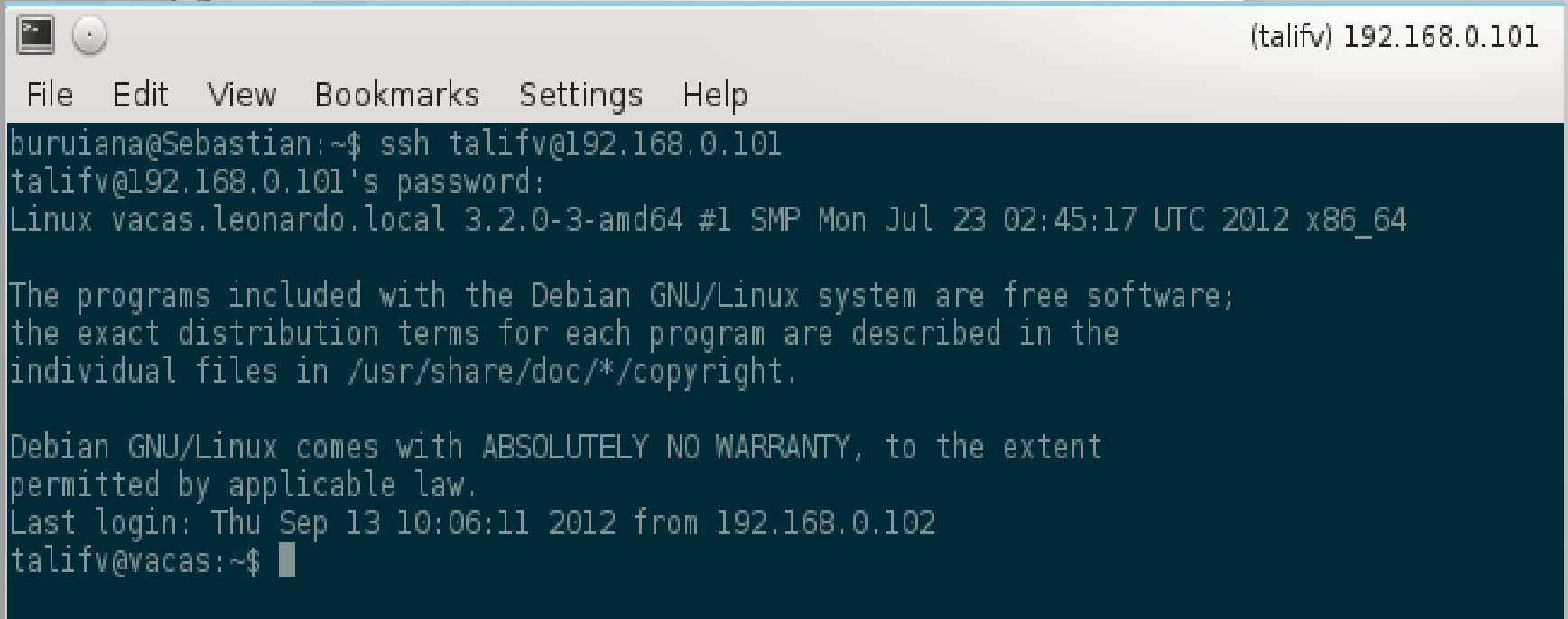
Secondly, Linux comes with a remarkably easy to use feature for creating groups and assigning members to them. This function requires little more than typing several commands, such as 'gpasswd' to add a username to a given group, or 'adduser' followed by a group name to create an account directly in the aforementioned group. This feature comes in very handy on machines working with a large userbase.



GNU/Linux

SSH

One easy way to connect to other machines in your LAN with Linux is via SSH, the Secure Shell Daemon.



```
(talifv) 192.168.0.101
File Edit View Bookmarks Settings Help
buruiana@Sebastian:~$ ssh talifv@192.168.0.101
talifv@192.168.0.101's password:
Linux vacas.leonardo.local 3.2.0-3-amd64 #1 SMP Mon Jul 23 02:45:17 UTC 2012 x86_64

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: Thu Sep 13 10:06:11 2012 from 192.168.0.102
talifv@vacas:~$
```

This is done with the simple issuing of the command `ssh` followed by the name of the user, the `@` symbol, and his IP address. Through `ssh`, users can modify or exchange (via the '`scp`' command) files between their stations, provided sufficient permission is granted.



GNU/Linux

Security



SSH is safe because one cannot log in to the master user (root) without knowing the specific password, and unprotected user accounts, which can be freely accessed through Secure Shell, don't usually have writing permission to sensible files. In addition, if necessary, the root user can always ban another station via 'iptables', reliably cutting all access to his machine from that particular terminal.

Virtual Machines

Most UNIX-like systems also support the installation of multiple virtual machines. A virtual machine is by definition an isolated software implementation of an OS, which is given limited resources.

Virtual Machines are often referred to as VMs.



GNU/Linux

There are two different categories of Virtual Machines :

1. System Virtual Machines
2. Process Virtual Machines



1. System Virtual Machines

A system VM provides a complete system platform, which is required to fully execute the OS and support multiple processes. Its main advantages are :

Multiple OS can co-exist without interfering with each other

The possibility of obtaining a different instruction set architecture (ISA), which gives you the chance of testing and comparing different operating systems at the same time

Easier maintenance, better application management and disaster recovery

GNU/Linux

However, System VMs are less efficient than physical machines not only due to the restricted resource input, but because of the indirect link to the hardware.

2. Process Virtual Machines

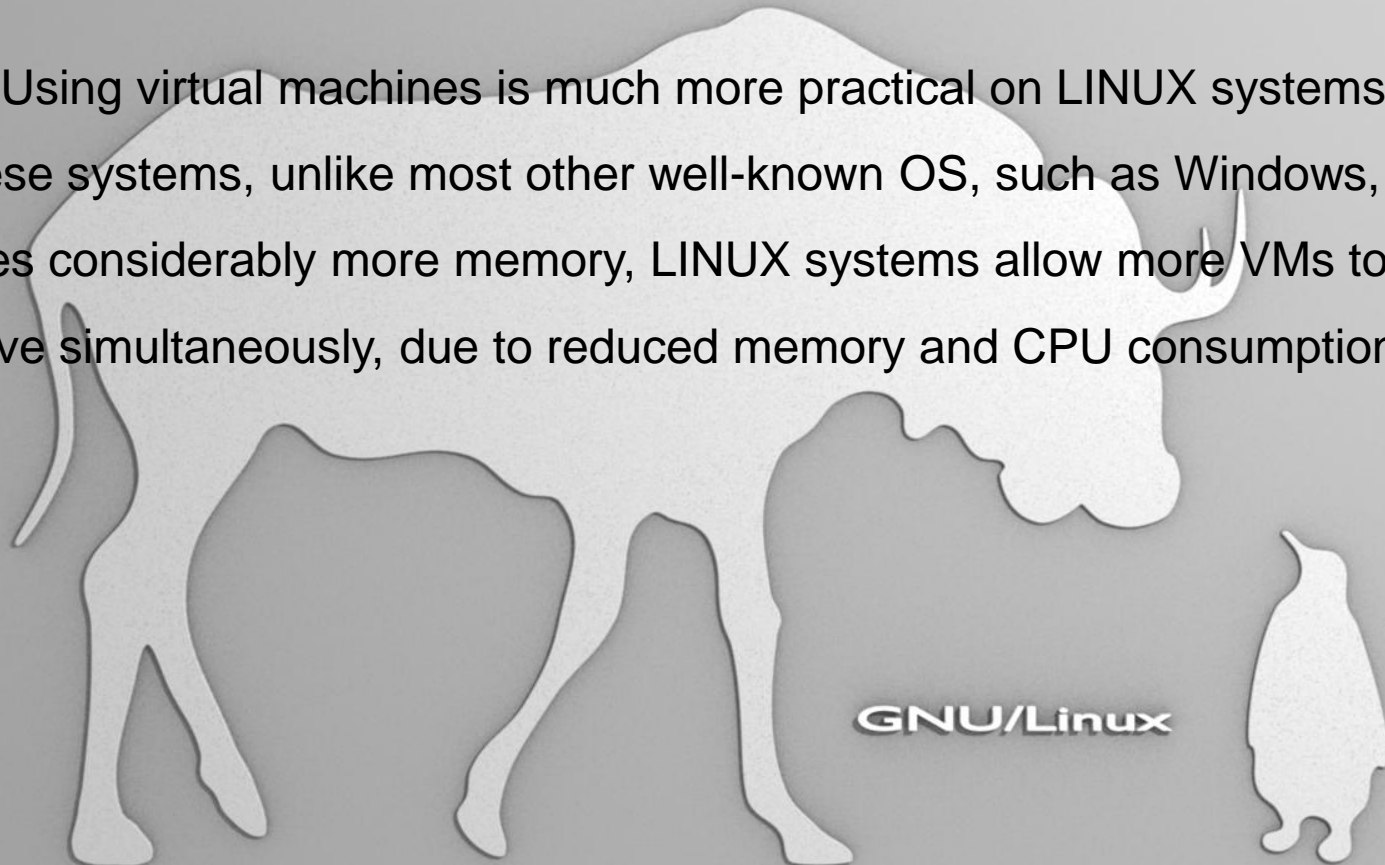
A process VM runs as an application inside another OS and is only able to support a single process. It is created at the start of the process and is deleted when the program is closed. It uses a far more advanced programming language than the system VMs. Therefore, they are most often created using an interpreter.



GNU/Linux

VMs on LINUX-based systems

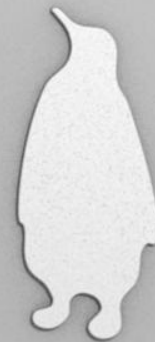
Using virtual machines is much more practical on LINUX systems. These systems, unlike most other well-known OS, such as Windows, which takes considerably more memory, LINUX systems allow more VMs to be active simultaneously, due to reduced memory and CPU consumption.



Chapter III

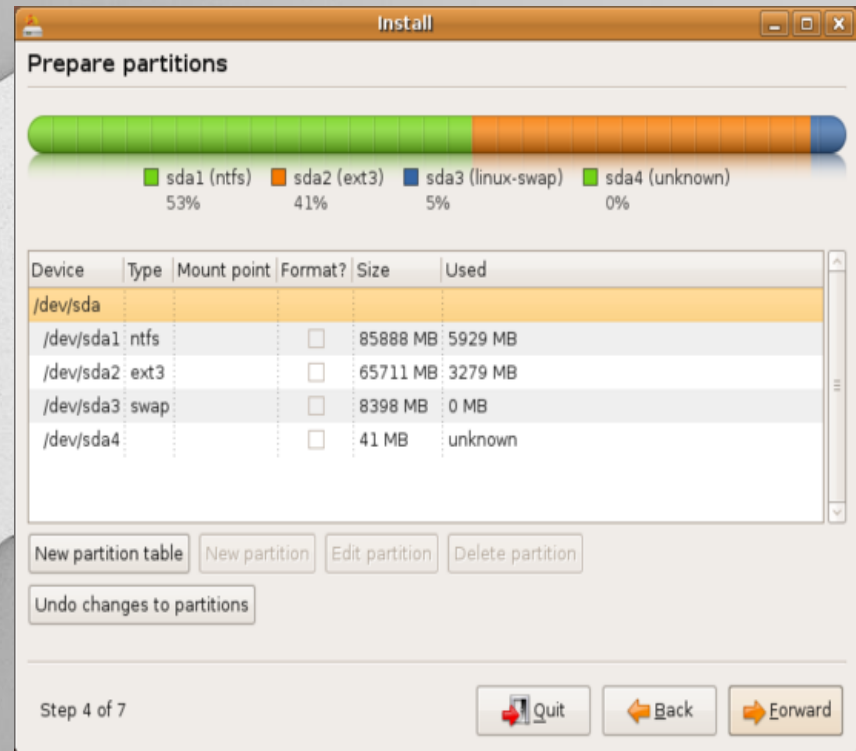
- Installing Linux (DualBoot)
- Distributions
- Installation of applications
- Window managers and desktop environments
- Program equivalents Windows - Linux

GNU/Linux



Linux Installation (DualBoot)

For data exchange between a Linux distribution and Windows it is recommended to create a partition that both systems can read and write. First, the hard drive must be partitioned. Then one should make sure that the PC is configured to boot from CD/DVD (BIOS Setup).



If the boot from DVD is successful you should get a graphical installation guide and you should simply follow the steps recommended by this guide.

Linux Distributions

Distributions (for short Distros) are operating systems which contain a Linux kernel, GNU utilities and libraries and a wide collection of software applications. Due to the fact that most of the Linux kernels and software packages are based on the idea of free and open source software, a wide variety of Linux distributions are now available. There are two main types of Linux distributions:



- the commercially backed distributions (such as Fedora- Red Hat or Ubuntu- Canonical Ltd.)
- the entirely community-driven distributions (such as Gentoo or Debian).



Ubuntu

There are many reasons why Ubuntu is one of the most popular distributions with users: a well web-based infrastructure, free support period, professional approach to the end user. Moreover, Ubuntu is the right choice if you are just starting to work with Linux. On the other hand, one would not pick Ubuntu because of the frequent major changes of this distribution, not to mention the fact that it lacks compatibility with Debian.



Fedora

Perhaps security is one of the most outstanding feature of Fedora distribution. It is also regarded as the most innovative Linux system. One can have problems with desktop usability (the lack of a clear strategy regarding the desktop), since it is not one of Fedora's main priority.



CentOS

Has gained a reputation of a solid and reliable service distribution, being preferred by those who need stability and long-term support instead of the newest software available. Although it does not live up to the expectations of its users regarding security updates and it lacks the newest versions of Linux, the software packages and the Linux kernel are well-tested and stable.



Debian

Developed by more than 1000

volunteers, it is easy to

understand that Debian is rich in

software packages compiled for

11 processor architectures.

However, there are also some

disadvantages when using Debian. For instance,

the release cycle is quite slow (one stable release every 1-3 years)

and sometimes the newest technologies are not included, Debian

being a conservative distribution.



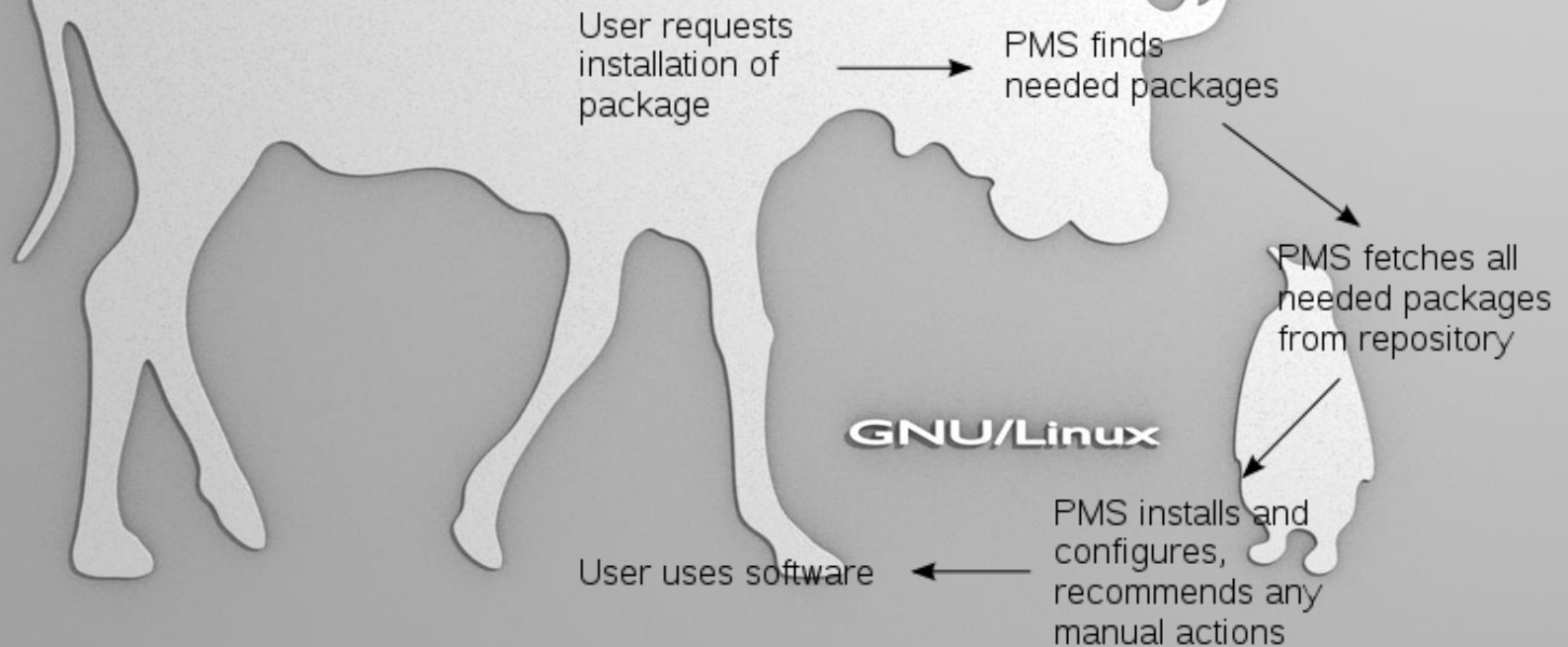
Installation of applications

The installation of programs under UNIX-like systems is different than in Windows in the sense that in Linux, you don't always have to download an installation package from a web site to install a program - very often, they're located in your distribution's repositories. A software repository is a storage location from which software packages may be retrieved and installed on a computer.



GNU/Linux

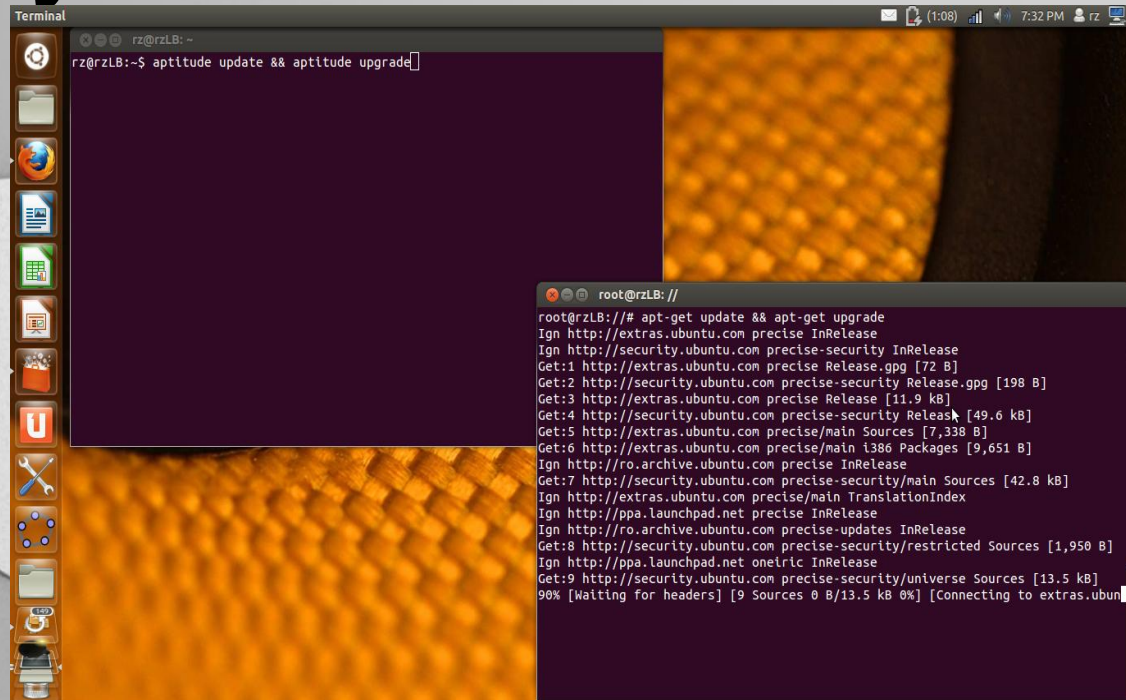
Repositories may be solely for particular programs or for an entire operating system. Operators of such repositories typically provide a package management system, tools intended to search for, install and otherwise manipulate software packages from the repositories.



Types of package management systems

There are types of package management systems according to different distributions of Linux:

- yum – RedHat Linux
- APT – Debian/Ubuntu developed in 1998 by the apt team
- aptitude – the predecessor of apt, developed in 2004 by Daniel Burrows



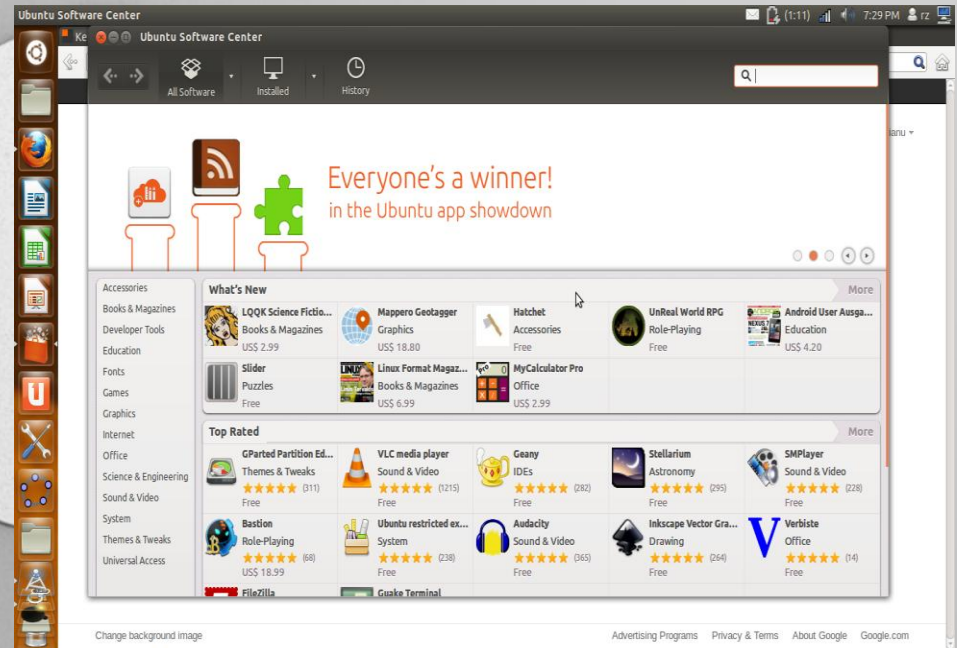
```
Terminal
rz@rzLB: ~
rz@rzLB:~$ aptitude update && aptitude upgrade

root@rzLB: //
root@rzLB: // # apt-get update && apt-get upgrade
Ign http://extras.ubuntu.com precise InRelease
Ign http://security.ubuntu.com precise-security InRelease
Get:1 http://extras.ubuntu.com precise Release.gpg [72 B]
Get:2 http://security.ubuntu.com precise-security Release.gpg [198 B]
Get:3 http://extras.ubuntu.com precise Release [11.9 kB]
Get:4 http://security.ubuntu.com precise-security Release [49.6 kB]
Get:5 http://extras.ubuntu.com precise/main Sources [7,338 B]
Get:6 http://extras.ubuntu.com precise/main i386 Packages [9,651 B]
Ign http://ro.archive.ubuntu.com precise InRelease
Get:7 http://security.ubuntu.com precise-security/main Sources [42.8 kB]
Ign http://extras.ubuntu.com precise/main TranslationIndex
Ign http://ppa.launchpad.net precise InRelease
Ign http://ro.archive.ubuntu.com precise-updates InRelease
Get:8 http://security.ubuntu.com precise-security/restricted Sources [1,950 B]
Ign http://ppa.launchpad.net onelric InRelease
Get:9 http://security.ubuntu.com precise-security/universe Sources [13.5 kB]
90% [Waiting for headers] [9 Sources 0 B/13.5 kB 6%] [Connecting to extras.ubun
```

These are accessible with the use of the console and can be used for a normal installation or upgrade of a single program to the task of updating and upgrading an entire operating system.

Types of package management systems

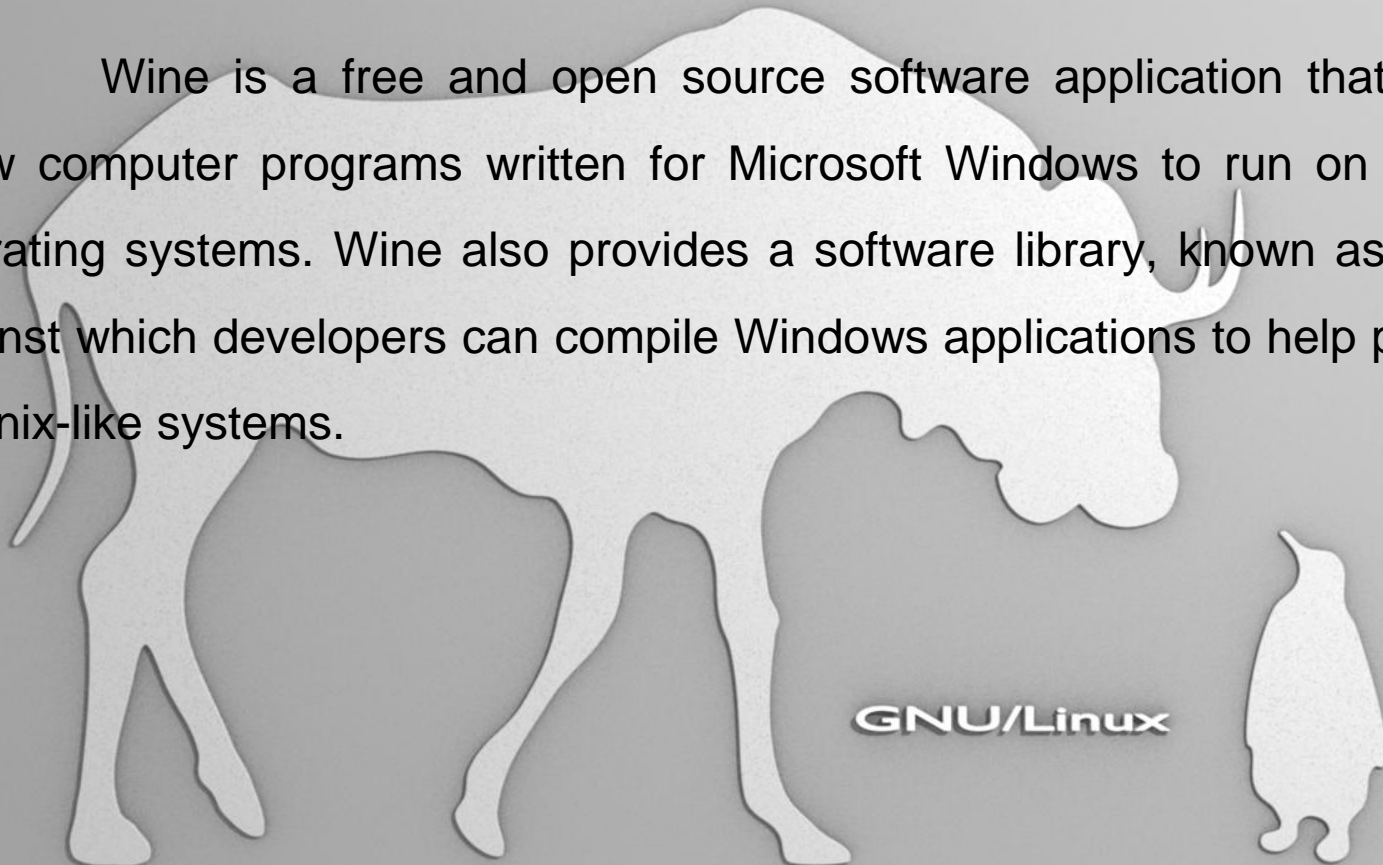
Because they require the use of a console, these applications may seem inaccessible to newcomers, but there are programs that combine the point-and-click simplicity of the graphical user interface with the power of the apt-get command line tool. You can install, remove, configure, or upgrade software packages, browse, sort and search the list of available software packages, manage repositories, or upgrade the whole system.



GNU/Linux

Applications

Wine is a free and open source software application that aims to allow computer programs written for Microsoft Windows to run on Unix-like operating systems. Wine also provides a software library, known as Winelib, against which developers can compile Windows applications to help port them to Unix-like systems.



Window managers and desktop environments

A window manager is a type of software which handles windows, providing the user with the ability to control programs using the graphical user interface.

Some examples of window managers for UNIX-like systems are KWin, Sawfish, BlackBox, IceWM, Metacity and Afterstep.

A desktop environment includes a window manager and builds upon it. There are two main desktop environments which are used for UNIX-like systems: GNOME and KDE.

GNU/Linux



GNOME or KDE?



KDE and GNOME are complete desktop environments that consist of a large number of pieces of software. GNOME uses a window manager called Metacity and KDE uses KWin, but both of these desktops can be used with any other window manager.

These desktop environments can be customized by the user in order for them to meet his needs. However, some users may find one of them to be more useful for the type of activities they wish to undertake.

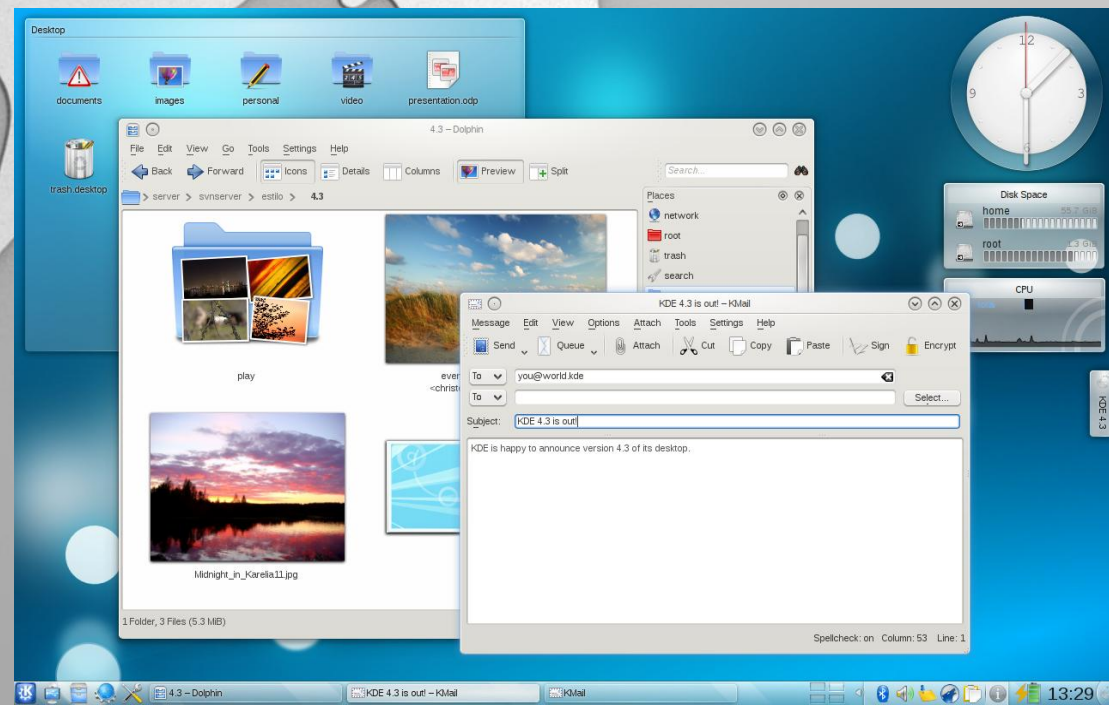
In December 2005, Linus Torvalds explicitly recommended the use of KDE because of it being easily configurable. Despite it aiming to be more user-friendly, GNOME still has options for power user hidden away in GConf (a system used by the GNOME desktop environment for storing configuration settings for the desktop and applications).



GNOME 3.0

[Home](#)

KDE 4.3.0



Differences in usability and performance



GNOME has less options readily visible. Its minimalistic offering of configuring options appeals to new users.

Startup times are generally faster on GNOME desktop environments and so are standard actions, such as opening applications, browsing devices, loading icons etc.

KDE has more options readily visible. It is easily configurable but it can baffle new users.

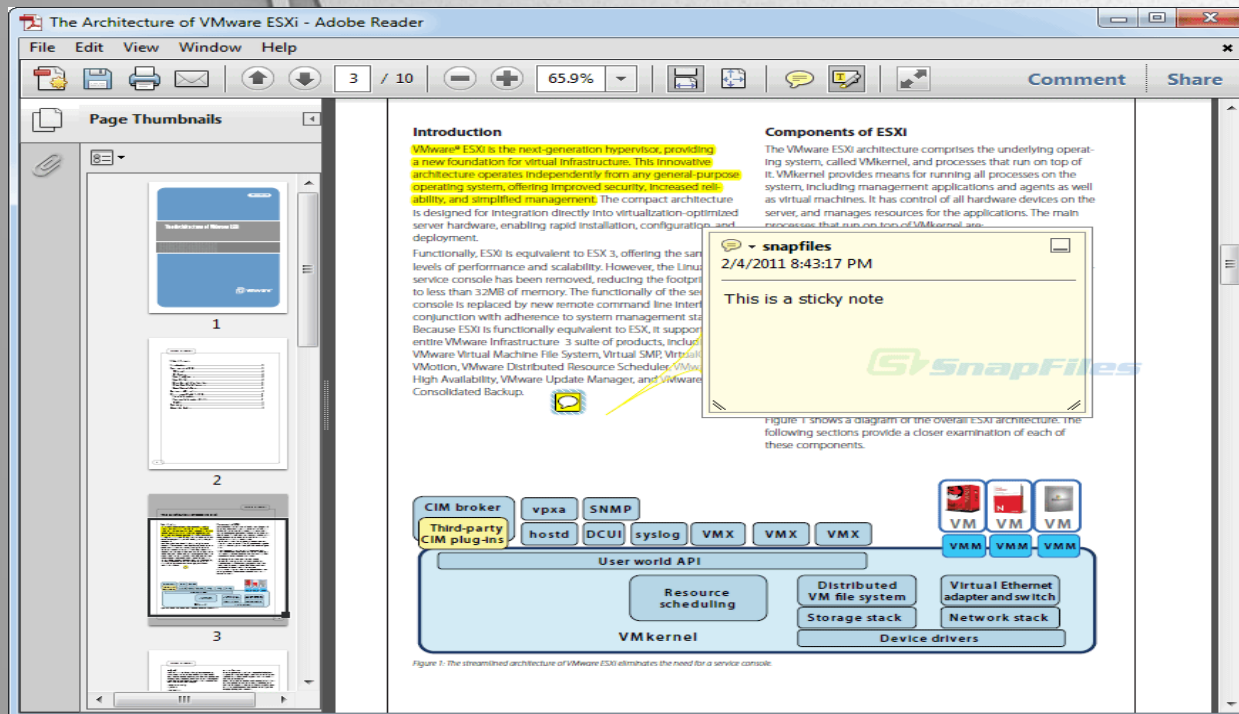
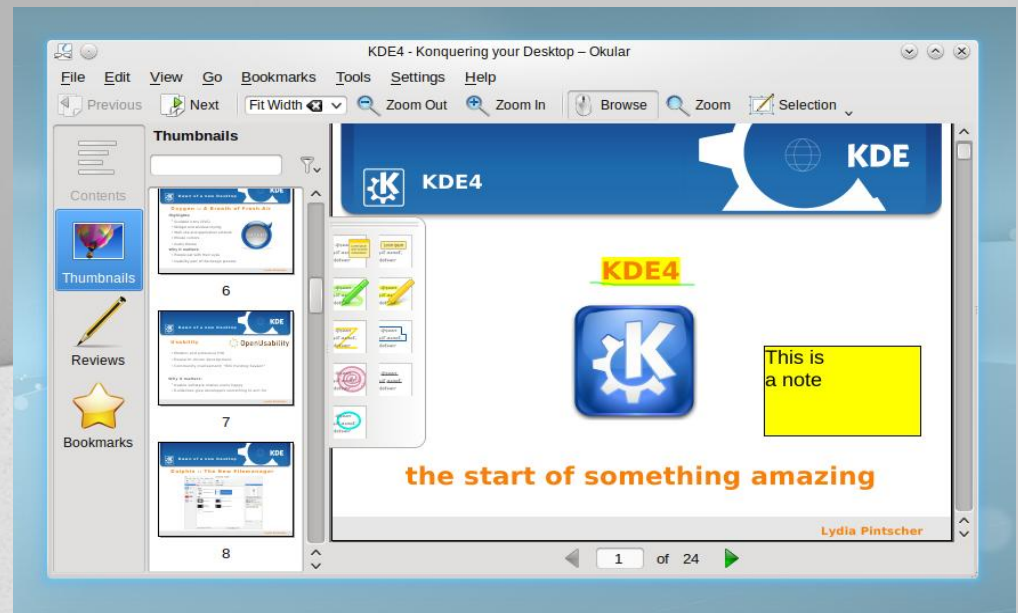
Besides occupying more memory, loading time on KDE desktop environments is longer because they have to load more from disk. A compensation for this, however, is the fact that once the libraries are loaded, applications start quicker.

Program equivalents Windows - Linux

The majority of the **Windows** programs are made on principle "all-in-one" (each developer adds everything to his or her product). This principle is called the "Windows-way". The ideology of **UNIX/Linux** is that one component or one program must execute only one task, but execute it well. ("UNIX-way"). The programs under **Linux** can be thought of as being like the LEGO building blocks.

This principle is very important and it is necessary to know it while searching for analogs of **Windows**-programs in **Linux**.

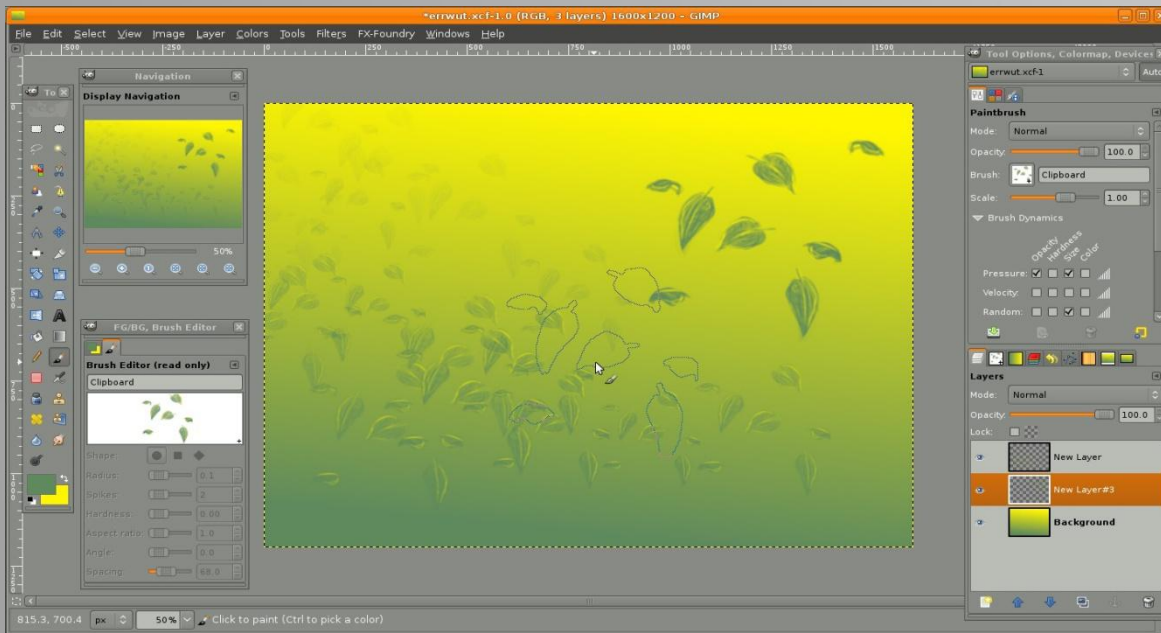




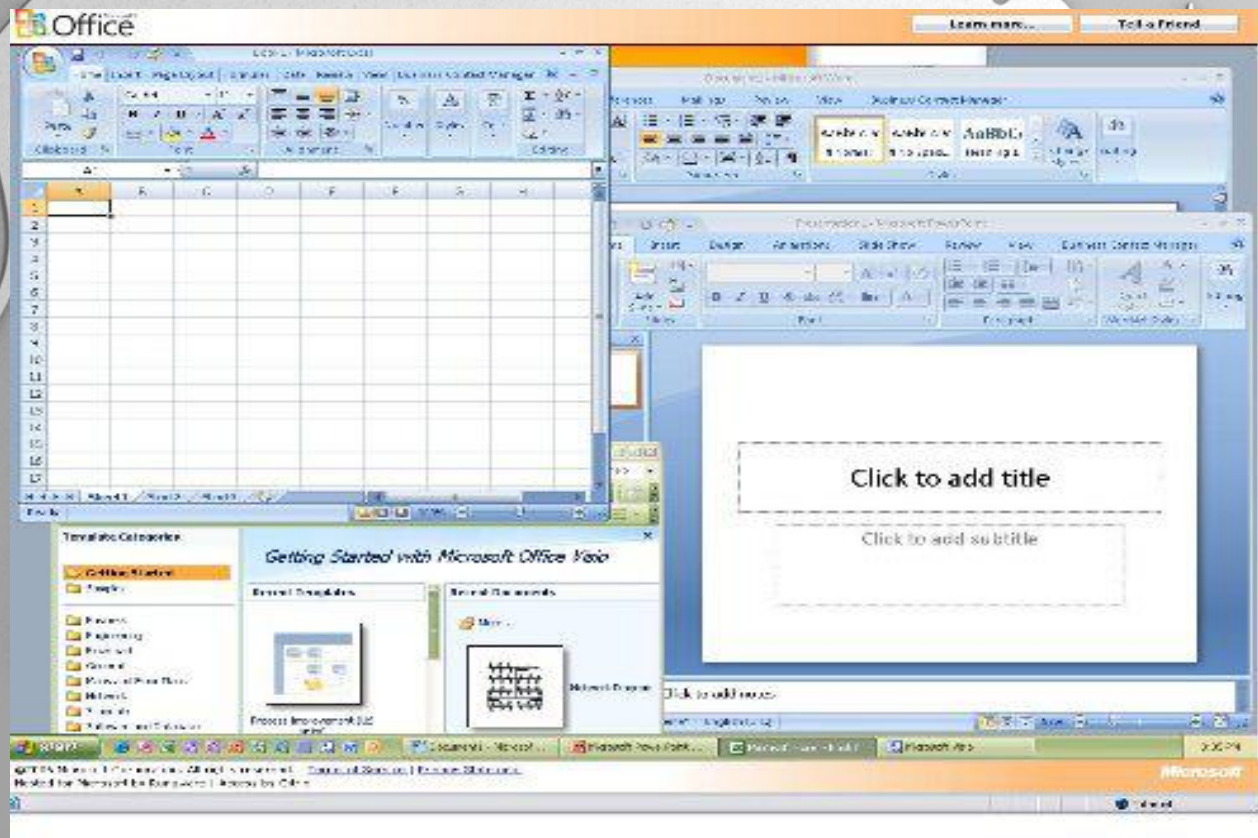
One of the alternatives for the graphics editing program **Adobe Photoshop** is **GIMP** (GNU image manipulation program). **GIMP** has a simple and easy to use Graphical User Interface compared to a more cluttered working environment in **Photoshop**, is faster and requires less disk space.



Professional photographers prefer **Photoshop** over **GIMP**, while **GIMP** is best for hobbyist or amateurs as a free alternative to **Photoshop** or as a learning tool before buying the expensive **Adobe**.

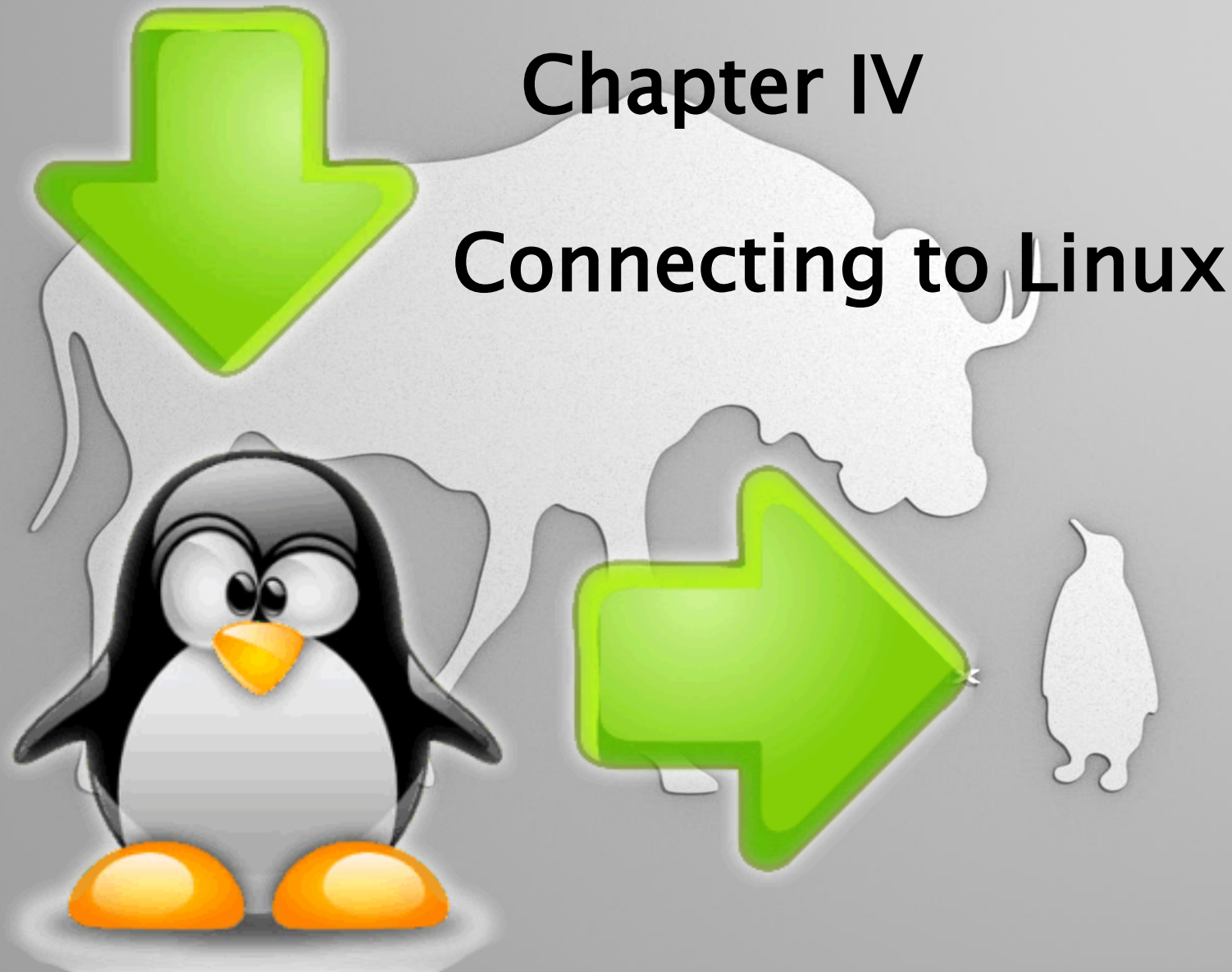


Microsoft Office was one of the first office suites of desktop applications: user-friendly, powerful and with new and unique capabilities. But over time , an open-source alternative, **OpenOffice**, has become increasingly more popular, making it a viable, affordable option for organizations seeking a productivity suite.



OpenOffice has a simple design which doesn't affect the power and functionality of the software but gives the advantage of being more usable and productive. Their performance is fairly similar, but **Microsoft** offers more advanced features in editing and document-viewing. Also, **Microsoft** offers quick and efficient support, in contrast to **OpenOffice** which has useful and extensive user and developer forums and a good deal of support across the net but isn't comparable to what is on offer for the other package.





About Hardware

- Computer hardware is a collection of physical elements that make up a usable PC.
- In today's world, there is vast amount of operating systems and hardware in circulation.
- The command on Debian that allows the user to see all the hardware or peripheral components is „lspci“.
- According to the linux manual, the „lspci“ command is „a utility for displaying information about PCI (Peripheral Component Interconnect) buses in the system and devices connected to them“.
- The devices are listed on the Console as series of numbers (including their PID numbers) as shown in the image below.

Google Custom Search

Search

Check

assembled

AAEON

ABIT

AHTEC

AMD

AMI

AOpen

ASUS

ATI

AZZA

Acer

Advantech

Ahtec

Airis

Albatron

Apple

Aspire

Asrock

Averatec

Axper

BANGHO

BenQ

Biostar

Buffalo

CCE

COMMEL

CREATIVE

Canon

Chaintech

Cisnet

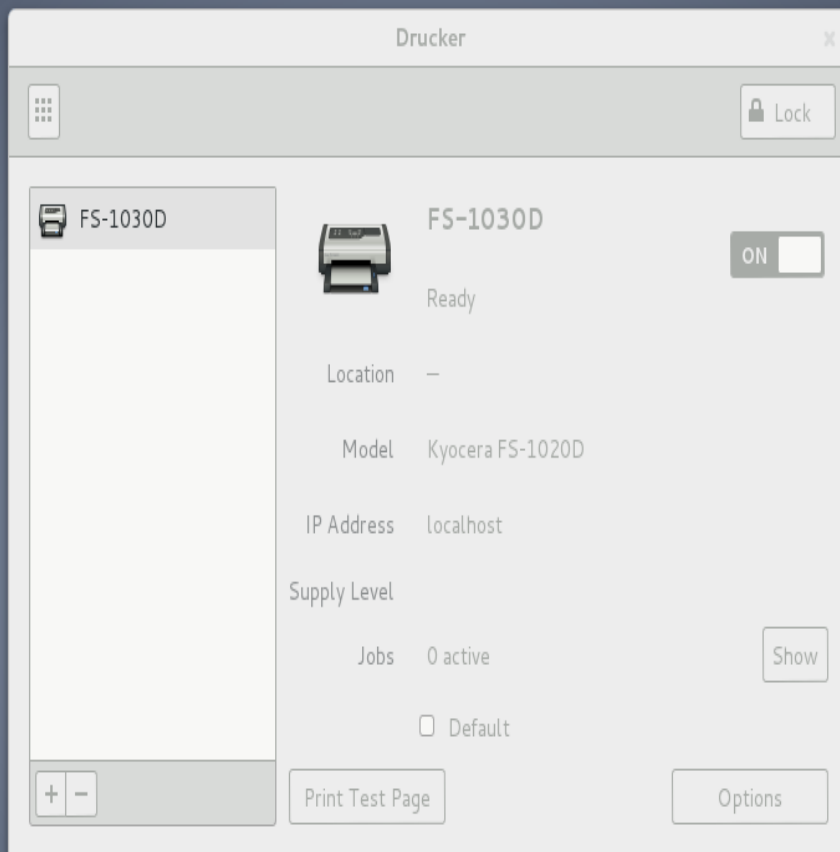
PCI ID	Works?	Vendor	Device	Driver	Kernel
80862e10	Yes	Intel Corporation	4 Series Chipset DRAM Controller	intel-agp	
80862e11		Intel Corporation	4 Series Chipset PCI Express Root Port		
80862e12	Yes	Intel Corporation	4 Series Chipset Integrated Graphics Controller	i915	v2.6.32-
80862e14		Intel Corporation	4 Series Chipset HECI Controller		
80862e16		Intel Corporation	4 Series Chipset PT IDER Controller		
80862e17		Intel Corporation	4 Series Chipset Serial KT Controller		
808610de	Yes	Intel Corporation	82567LM-3 Gigabit Network Connection	e1000e	v2.6.28-
80863a67		Intel Corporation	82801JD/DO (ICH10 Family) USB UHCI Controller #4		
80863a68		Intel Corporation	82801JD/DO (ICH10 Family) USB UHCI Controller #5		
80863a69		Intel Corporation	82801JD/DO (ICH10 Family) USB UHCI Controller #6		
80863a6c		Intel Corporation	82801JD/DO (ICH10 Family) USB2 EHCI Controller #2		
80863a6e	Yes	Intel Corporation	82801JD/DO (ICH10 Family) HD Audio Controller	snd-hda-intel	v3.1.0-
80863a70		Intel Corporation	82801JD/DO (ICH10 Family) PCI Express Port 1		
80863a78		Intel Corporation	82801JD/DO (ICH10 Family) PCI Express Port 5		
80863a64		Intel Corporation	82801JD/DO (ICH10 Family) USB UHCI Controller #1		
80863a65		Intel Corporation	82801JD/DO (ICH10 Family) USB UHCI Controller #2		
80863a66		Intel Corporation	82801JD/DO (ICH10 Family) USB UHCI Controller #3		
80863a6a		Intel Corporation	82801JD/DO (ICH10 Family) USB2 EHCI Controller #1		
8086244e	Yes	Intel Corporation	82801 PCI Bridge	i810_mg	
80863a14	Yes	Intel Corporation	82801JDO (ICH10DO) LPC Interface Controller	iTCO_wdt	v2.6.28-
80863a00	Yes	Intel Corporation	82801JD/DO (ICH10 Family) 4-port SATA IDE Controller	ata_piix	v2.6.25-
80863a60	Yes	Intel Corporation	82801JD/DO (ICH10 Family) SMBus Controller	i2c-i801	v2.6.25-
80863a06	Yes	Intel Corporation	82801JD/DO (ICH10 Family) 2-port SATA IDE Controller	ata_piix	v2.6.25-

You can help us!

Could you give me more information about your machine? Your information (PCI list and text you write below) is used only for making our HCL database better.

„Translated“, each series of numbers is a device (source: kmuto.jp/debian/hcl/index.cgi)

Connecting a device(e.g. printer) to the PC

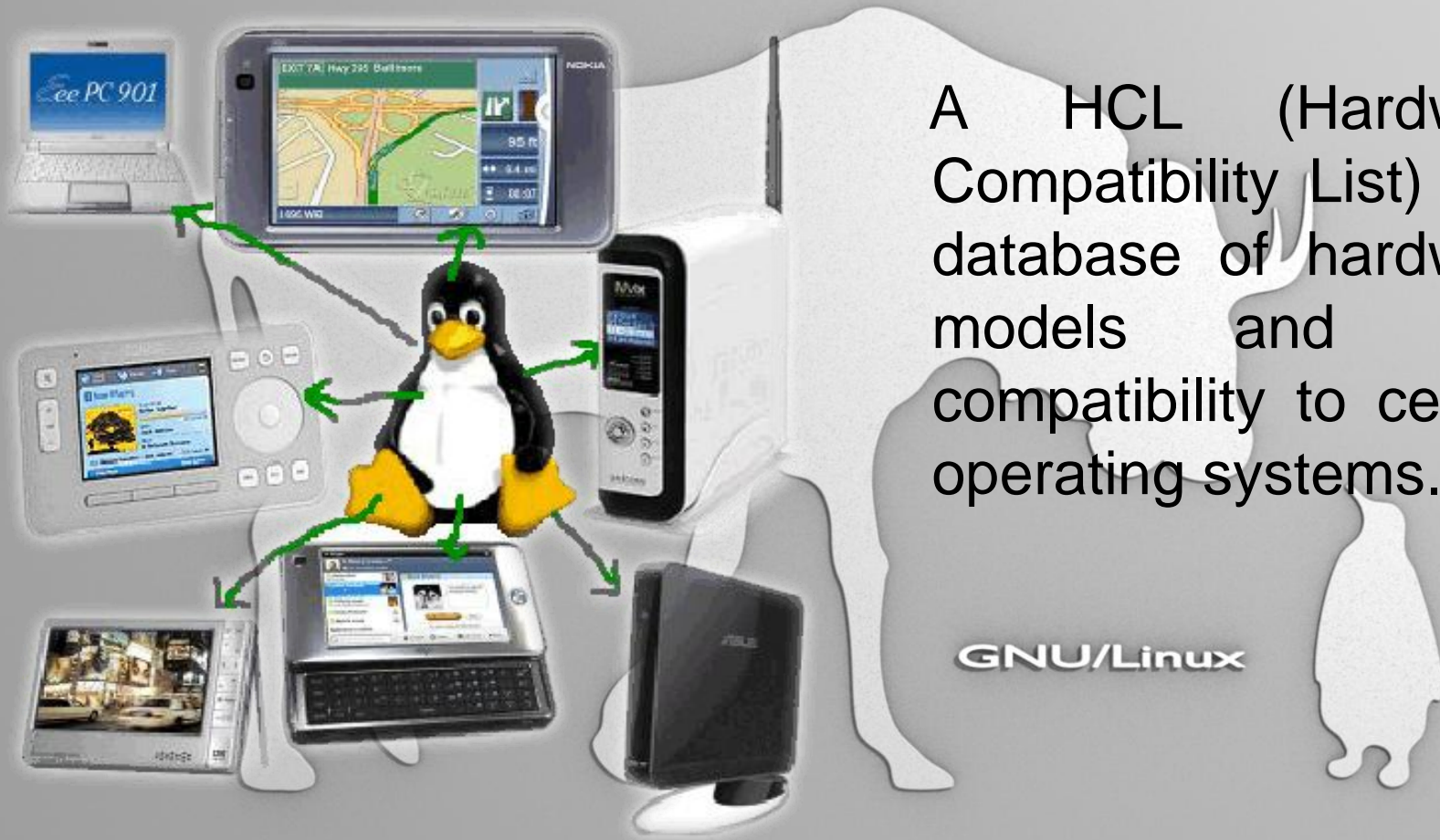


In order to install a printer, one must at first install the „cups“ program using the Console. z

After the printer is connected to the computer, the option of installing it can be found in the System Settings--> Printer.

GNU/Linux

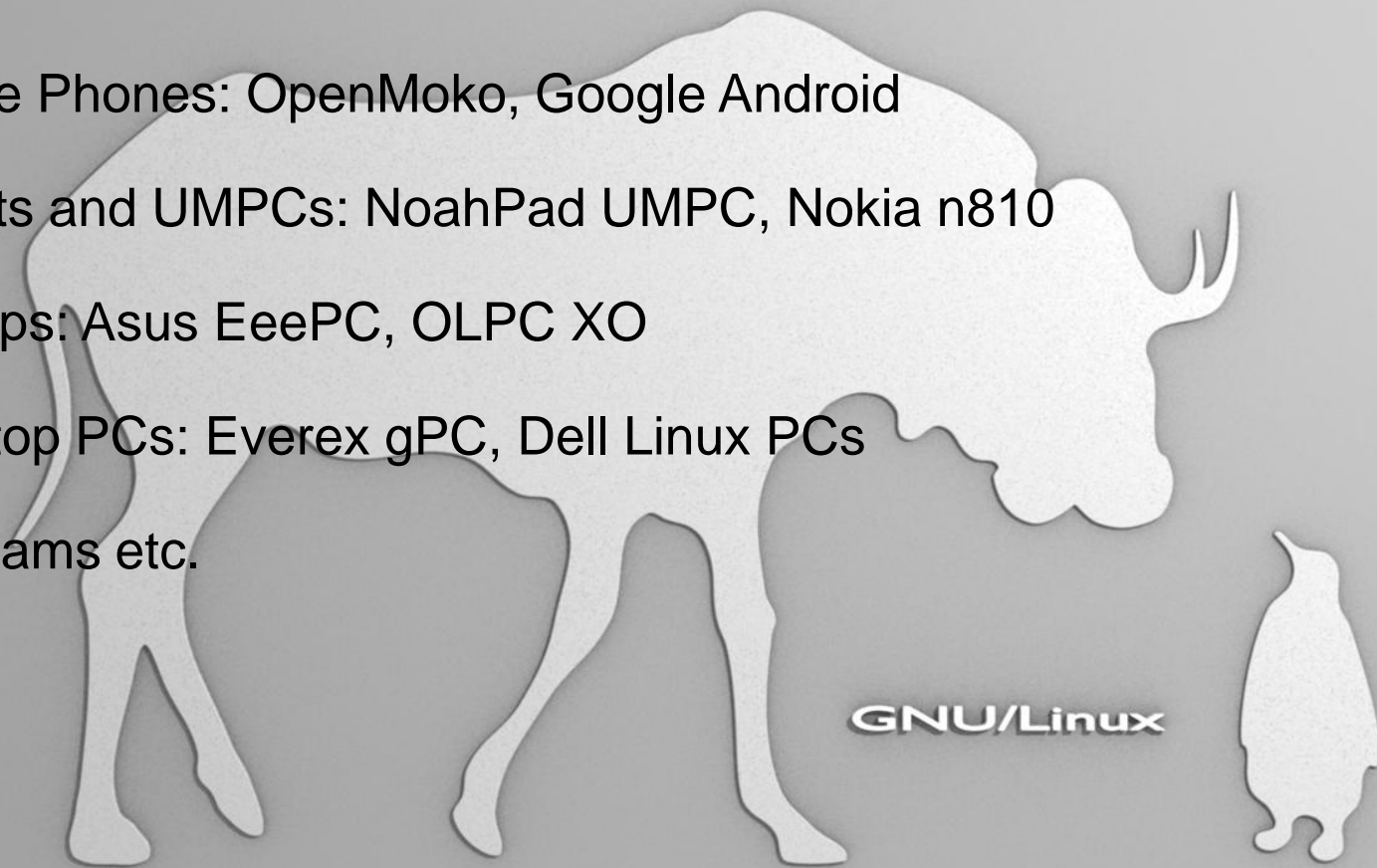
Hardware compatibility and recognition on Linux



A HCL (Hardware Compatibility List) is a database of hardware models and their compatibility to certain operating systems.

Linux Hardware Compatibility List

- Mobile Phones: OpenMoko, Google Android
- Tablets and UMPCs: NoahPad UMPC, Nokia n810
- Laptops: Asus EeePC, OLPC XO
- Desktop PCs: Everex gPC, Dell Linux PCs
- Webcams etc.



Android

The Android system is probably the most successful linux-based device which became, by November 2011, the best-selling platform worldwide with over 200 million Android devices in use.

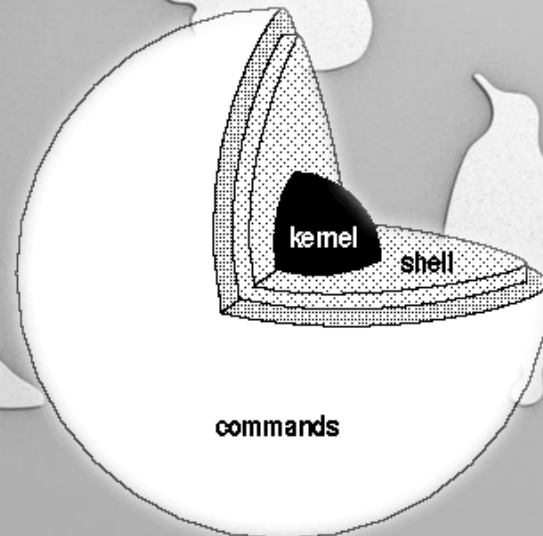
Android's kernel is based on the Linux kernel 2.6 and Linux Kernel 3.x, but has some architectural changes by Google such as autosleep and wakeclocks.



The kernel introduction

The word “kernel” makes someone probably think of the usually edible part of a seed or the most important part of an idea. When talking about computers, it is the connection between the hardware and the software, the central module of an operating system.

a Kernel



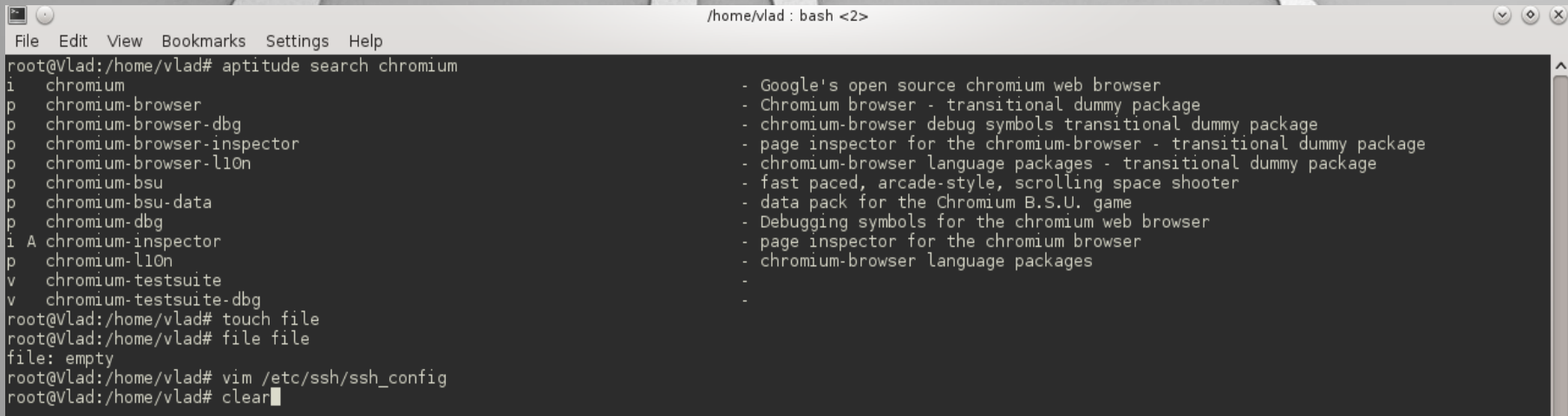
Our meeting with the kernel



Our first meeting with it was when we started the booting process. It is the one loaded first by the bootloader and remains in the main memory while being executed. That's why, it is important to be as small as possible, but still capable of providing the essential services required by the rest of the operating systems and the applications. After that, it hands control to a special process called init, “the mother of all processes”. In addition, it creates two types of virtual files, which can be found in the directories `/dev` and `/proc`.

User and kernel

The kernel code is loaded into a protected area of memory, the kernel space and the user space being separated in order to prevent data from interfering. In Linux, it becomes a shell where the user can write commands to control the applications, make files/directories, manage everything and therefore use the operating system installed.



```
/home/vlad : bash <2>
File Edit View Bookmarks Settings Help
root@Vlad:/home/vlad# aptitude search chromium
i chromium
p chromium-browser
p chromium-browser-dbg
p chromium-browser-inspector
p chromium-browser-ll0n
p chromium-bsu
p chromium-bsu-data
p chromium-dbg
i A chromium-inspector
p chromium-ll0n
v chromium-testsuite
v chromium-testsuite-dbg
root@Vlad:/home/vlad# touch file
root@Vlad:/home/vlad# file file
file: empty
root@Vlad:/home/vlad# vim /etc/ssh/ssh_config
root@Vlad:/home/vlad# clear
```

- Google's open source chromium web browser
- Chromium browser - transitional dummy package
- chromium-browser debug symbols transitional dummy package
- page inspector for the chromium-browser - transitional dummy package
- chromium-browser language packages - transitional dummy package
- fast paced, arcade-style, scrolling space shooter
- data pack for the Chromium B.S.U. game
- Debugging symbols for the chromium web browser
- page inspector for the chromium browser
- chromium-browser language packages
-
-

Kernel's functions

If we try to observe what a kernel does, you can say it has different functions: it plays the part of a **scheduler**, allocating every process the time needed, a **supervisor** who grants them the permission when scheduled, it **handles the problems**, taking care of the processes' requests and interrupting them before anything bad happens and last but not least, it is a **memory manager** who redirects a certain part of it to every process in order to run.



Types of a kernel

There are more types of kernels, the most important being:

- **monolithic**, usually used by the Unix-like operating systems (such as Linux, FreeBSD),
- **hybrid** (Microsoft Windows NT, 2000, XP, Vista, 7 and MAC OS X),
- **microkernels** (MINIX, GNU Hurd)
- **exokernels**, which are very small and still experimental

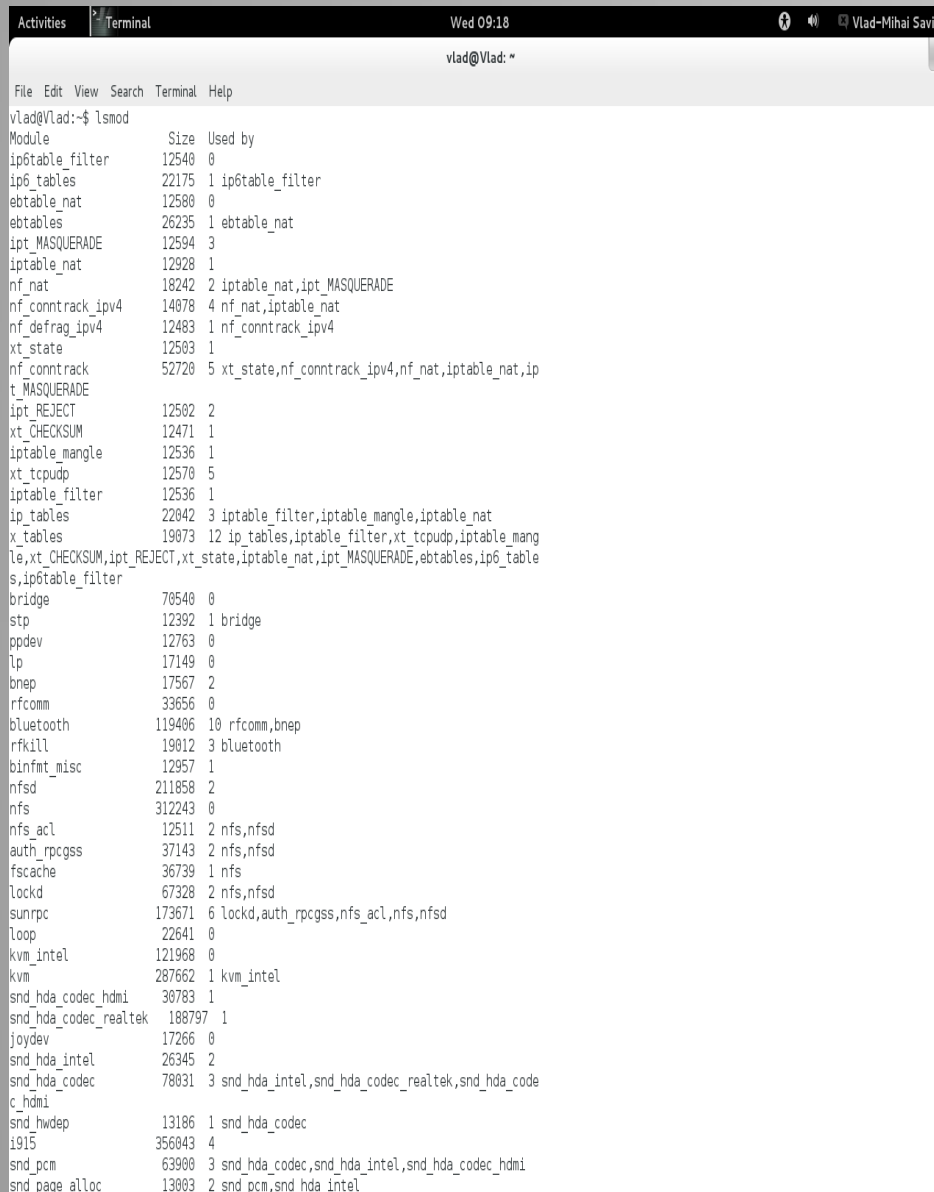
GNU/Linux

The main differences between the first three categories mentioned are: monolithic kernels can load modules on their own, microkernels include some additional code in the kernel space to increase performance and the hybrid ones are made to run swiftly, but slower..

Modules

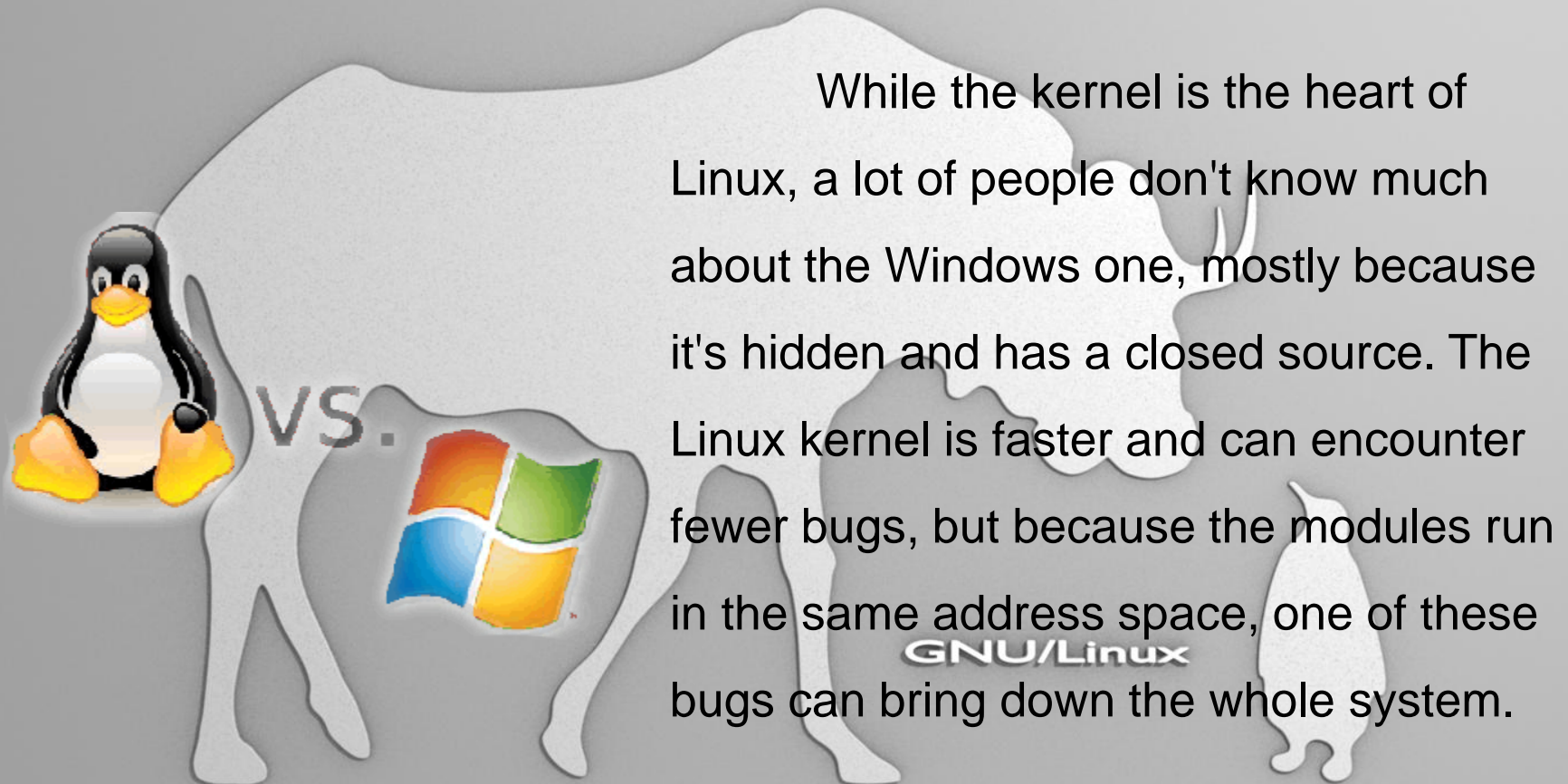
Because Linux is a monolithic kernel, the kernel can use modules. These modules are pieces of code that can be loaded and unloaded when needed for different features requested by the user, but without having to reboot the system. They can be considered an extension of the kernel itself with the same functionality. Therefore, it doesn't have to load everything in the booting process.

For example, one type of a module can be a device driver, which allows the hardware to connect to the system.



```
vlad@Vlad:~$ lsmod
Module                  Size  Used by
ip6table_filter         12540  0
ip6_tables              22175  1 ip6table_filter
ebtable_nat             12580  0
ebtables               26235  1 ebtable_nat
ipt_MASQUERADE          12594  3
iptable_nat            12928  1
nf_nat                 18242  2 iptable_nat,ipt_MASQUERADE
nf_conntrack_ipv4      14078  4 nf_nat,iptable_nat
nf_defrag_ipv4         12483  1 nf_conntrack_ipv4
xt_state               12503  1
nf_conntrack           52720  5 xt_state,nf_conntrack_ipv4,nf_nat,iptable_nat,ipt_MASQUERADE
ipt_REJECT              12502  2
xt_CHECKSUM            12471  1
iptables_mangle        12536  1
xt_tcpudp              12570  5
iptables_filter        12536  1
ip_tables              22042  3 iptables_filter,iptables_mangle,iptable_nat
x_tables               19073  12 ip_tables,iptables_filter,xt_tcpudp,iptables_mangle,xt_CHECKSUM,ipt_REJECT,xt_state,iptable_nat,ipt_MASQUERADE,ebtables,ip6_tables,ip6table_filter
bridge                 70540  0
stp                   12392  1 bridge
ppdev                 12763  0
lp                    17149  0
bnep                  17567  2
rfcomm                33656  0
bluetooth             119406 10 rfcomm,bnep
rfkill                19012  3 bluetooth
binfmt_misc           12957  1
nfsd                   211858 2
nfs                   312243 0
nfs_acl               12511  2 nfs,nfsd
auth_rpcgss           37143  2 nfs,nfsd
fscache               36739  1 nfs
lockd                 67328  2 nfs,nfsd
sunrpc               173671 6 lockd,auth_rpcgss,nfs_acl,nfs,nfsd
loop                  22641  0
kvm_intel             121968 0
kvm                   287662 1 kvm_intel
snd_hda_codec_hdmi    30783  1
snd_hda_codec_realtek 188797 1
joydev                17266  0
snd_hda_intel         26345  2
snd_hda_codec         78831  3 snd_hda_intel,snd_hda_codec_realtek,snd_hda_codec_hdmi
c_hdm_i               13186  1 snd_hda_codec
snd_hwdep             356043 4
1915                  63900 3 snd_hda_codec,snd_hda_intel,snd_hda_codec_hdmi
snd_pcm               13003  2 snd_pcm,snd_hda_intel
```

Linux kernel – Windows kernel



While the kernel is the heart of Linux, a lot of people don't know much about the Windows one, mostly because it's hidden and has a closed source. The Linux kernel is faster and can encounter fewer bugs, but because the modules run in the same address space, one of these bugs can bring down the whole system.

Chapter V

Practical uses and file extensions

GNU/Linux

In 3 weeks we got used with the console and now we don't have problems using commands and handling files and directories.



12.09.2012 08:55

ted@debianTED: ~

File Edit View Search Terminal Help

ted@debianTED:~\$ pwd

/home/ted

ted@debianTED:~\$ ls

123456789	authorized_keys	Desktop	Downloads	index.html	Pictures	set	Templates	test2	www
apache2.conf	BLA BLA	Documents	fstab	Music	Public	temp	test1	Videos	

ted@debianTED:~\$ mkdir directory

ted@debianTED:~\$ ls

123456789	authorized_keys	Desktop	Documents	fstab	Music	Public	temp	test1	Videos
apache2.conf	BLA BLA	<u>directory</u>	Downloads	index.html	Pictures	set	Templates	test2	www

ted@debianTED:~\$ cd directory/

ted@debianTED:~/directory\$ touch file1

ted@debianTED:~/directory\$ touch file2

ted@debianTED:~/directory\$ ls

file1 file2

ted@debianTED:~/directory\$ pico file1

ted@debianTED:~/directory\$ cat file1

This is a text document.

ted@debianTED:~/directory\$ rm file2

ted@debianTED:~/directory\$ cd ..

ted@debianTED:~\$ cd directory/

ted@debianTED:~/directory\$ cd ~

ted@debianTED:~\$ w

08:57:18 up 34 min, 1 user, load average: 0.94, 0.81, 0.70

USER	TTY	FROM	LOGIN@	IDLE	JCPU	PCPU	WHAT
ted	pts/0	:0	08:35	0.00s	0.10s	0.00s	w

ted@debianTED:~\$ whoami

ted

ted@debianTED:~\$ su

Password:

root@debianTED:/home/ted# passwd

Enter new UNIX password:

Retype new UNIX password:

passwd: password updated successfully

root@debianTED:/home/ted# whoami

root

root@debianTED:/home/ted# w

08:58:43 up 36 min, 1 user, load average: 0.37, 0.67, 0.66

USER	TTY	FROM	LOGIN@	IDLE	JCPU	PCPU	WHAT
ted	pts/0	:0	08:35	0.00s	0.10s	2.43s	gnome-terminal

root@debianTED:/home/ted#

```
ted@debianTED: ~  
  
File Edit View Search Terminal Help  
root@debianTED:/home/ted# ls  
123456789  aaal.tar      authorized_keys  Desktop  Documents  fstab      Music      Public  temp      test1  Videos  
aaa        apache2.conf  BLA BLA        directory  Downloads  index.html  Pictures  set      Templates  test2  www  
root@debianTED:/home/ted# cd directory/  
root@debianTED:/home/ted/directory# ls  
file1 file2 file3 file4 file5 file6  
root@debianTED:/home/ted/directory# cd ..  
root@debianTED:/home/ted# tar cvf archive.tar directory  
directory/  
directory/file4  
directory/file6  
directory/file3  
directory/file2  
directory/file1  
directory/file5  
root@debianTED:/home/ted# ls  
123456789  aaal.tar      archive.tar    BLA BLA  directory  Downloads  index.html  Pictures  set  Templates  test2  www  
aaa        apache2.conf  authorized_keys  Desktop  Documents  fstab      Music      Public  temp  test1      Videos  
root@debianTED:/home/ted# rm -r directory  
root@debianTED:/home/ted# tar tvf archive.tar  
drwxr-xr-x root/root      0 2012-09-12 09:39 directory/  
-rw-r--r-- root/root      0 2012-09-12 09:39 directory/file4  
-rw-r--r-- root/root      0 2012-09-12 09:39 directory/file6  
-rw-r--r-- root/root      0 2012-09-12 09:39 directory/file3  
-rw-r--r-- root/root      0 2012-09-12 09:39 directory/file2  
-rw-r--r-- root/root      0 2012-09-12 09:39 directory/file1  
-rw-r--r-- root/root      0 2012-09-12 09:39 directory/file5  
root@debianTED:/home/ted# tar xvf archive.tar  
directory/  
directory/file4  
directory/file6  
directory/file3  
directory/file2  
directory/file1  
directory/file5  
root@debianTED:/home/ted# ls  
123456789  aaal.tar      archive.tar    BLA BLA  directory  Documents  fstab      Music      Public  temp      test1  Videos  
aaa        apache2.conf  authorized_keys  Desktop  directory2  Downloads  index.html  Pictures  set      Templates  test2  www  
root@debianTED:/home/ted# cd directory/  
root@debianTED:/home/ted/directory# ls  
file1 file2 file3 file4 file5 file6  
root@debianTED:/home/ted/directory#
```

ted@debianTED: ~

File Edit View Search Terminal Help

PS(1)

User Commands

PS(1)

NAME

ps - report a snapshot of the current processes.

SYNOPSIS

ps [options]

DESCRIPTION

ps displays information about a selection of the active processes. If you want a repetitive update of the selection and the displayed information, use `top(1)` instead.

This version of **ps** accepts several kinds of options:

- 1 UNIX options, which may be grouped and must be preceded by a dash.
- 2 BSD options, which may be grouped and must not be used with a dash.
- 3 GNU long options, which are preceded by two dashes.

Options of different types may be freely mixed, but conflicts can appear. There are some synonymous options, which are functionally identical, due to the many standards and **ps** implementations that this **ps** is compatible with.

Note that "**ps -aux**" is distinct from "**ps aux**". The POSIX and UNIX standards require that "**ps -aux**" print all processes owned by a user named "x", as well as printing all processes that would be selected by the **-a** option. If the user named "x" does not exist, this **ps** may interpret the command as "**ps aux**" instead and print a warning. This behavior is intended to aid in transitioning old scripts and habits. It is fragile, subject to change, and thus should not be relied upon.

By default, **ps** selects all processes with the same effective user ID (euid=EUID) as the current user and associated with the same terminal as the invoker. It displays the process ID (pid=PID), the terminal associated with the process (tname=TTY), the cumulated CPU time in [DD-]hh:mm:ss format (time=TIME), and the executable name (ucmd=CMD). Output is unsorted by default.

The use of BSD-style options will add process state (stat=STAT) to the default display and show the command args (args=COMMAND) instead of the executable name. You can override this with the **PS_FORMAT** environment variable. The use of BSD-style options will also change the process selection to include processes on other terminals (TTYS) that are owned by you; alternately, this may be described as setting the selection to be the set of all processes filtered to exclude processes owned by other users or not on a terminal. These effects are not considered when options are described as being "identical" below, so **-M** will be considered identical to **Z** and so on.

Except as described below, process selection options are additive. The default selection is discarded, and then the selected processes are added to the set of processes to be displayed. A process will thus be shown if it meets any of the given selection criteria.

EXAMPLES

To see every process on the system using standard syntax:

```
ps -e
ps -ef
ps -eF
ps -ely
```

To see every process on the system using BSD syntax:

```
ps ax
ps axu
```

Manual page ps(1) line 1 (press h for help or q to quit)

Activities

Terminal

File Edit View Search Terminal Help

root@debianTED:/home/ted/laie# man ps

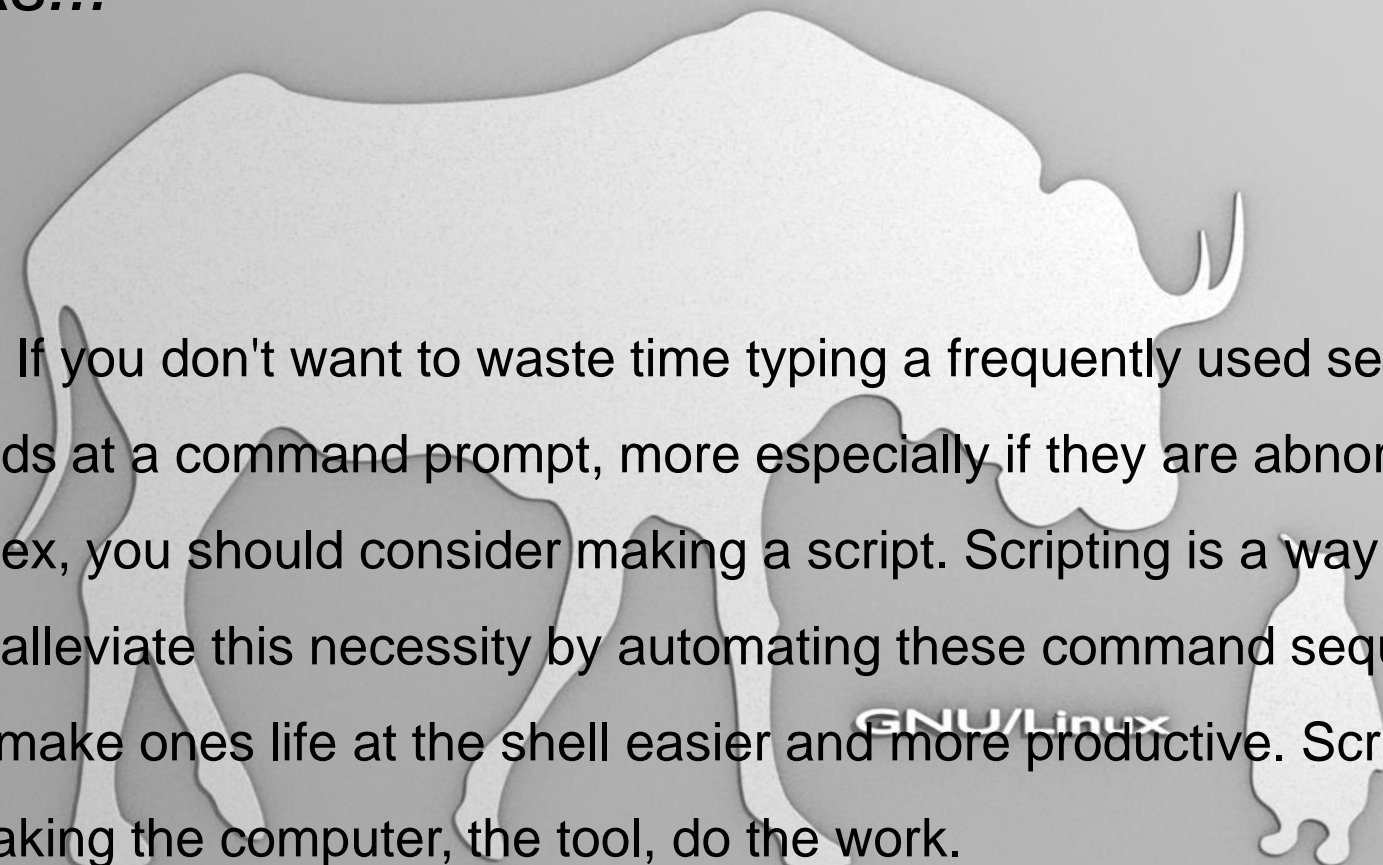
root@debianTED:/home/ted/laie#

File Edit View Search Terminal Help

```
455 man awk
456 awk '[print$1,$2]' laie
457 awk '[print$1,$2]' laie
458 awk '{print$1,$2}' laie
459 cat laie
460 awk '{print$1,$2}' laie
461 man awk
462 man file
463 man file
464 touch aal.au
465 file aal.au
466 man file
467 touch baba
468 pico baba
469 file -s baba
470 pico /etc/fstab
471 pico /etc/mtab
472 clear
473 history
474 clear
475 history
476 man echo
477 man ps
478 man ps
479 clear
480 man ps
481 ls
482 cd laie/
483 cd baie/
484 mkdir bla
485 cd bla/
486 touch script.sh
487 pico script.sh
488 ./script.sh
489 ./ script.sh
490 exit
491 man tar
492 clear
493 man date
494 ssh ted@mail.leonardo.local
495 ssh ted@mail.leonardo.local
496 ssh tedDebian@mail.leonardo.local
497 ssh ted@mail.leonardo.local
498 ssh ted@mail.leonardo.local
499 cd ..
500 cd ..
501 cd ..
502 cd ..
503 /etc/init.d/nullmailer restart
504 echo testmail |sendmail root
505 history
root@debianTED:/#
```

```
root@debianTED:/# history
```

- *Now it's time to write a shell script and see how it works...*



If you don't want to waste time typing a frequently used sequence of commands at a command prompt, more especially if they are abnormally long or complex, you should consider making a script. Scripting is a way by which one can alleviate this necessity by automating these command sequences in order to make ones life at the shell easier and more productive. Scripting is all about making the computer, the tool, do the work.

ted@debianTED: ~

File Edit View Search Terminal Help

GNU nano 2.2.6File: script.shModified

```
#!/bin/bash script.sh

clear

sleep 2

echo "Hey!"

echo "I am "

whoami

sleep 5

echo "We are on "

date +DATE:%d%m%y

echo "and the time is "

date +TIME:%H%M

sleep 10

echo "Hm, I'm bored. Let's execute an update and an upgrade."

sleep 5

aptitude update && aptitude upgrade

sleep 10

echo "My job here is done"

sleep 3

clear

echo "Bye!"
```

[Read 39 lines]

^G Get Help

^X Exit

^O WriteOut

^J Justify

^R Read File

^W Where Is

^Y Prev Page

^V Next Page

^K Cut Text

^U UnCut Text

^C Cur Pos

^T To Spell

```
Activities  Terminal  Thu 10:48
ted@debianTED: ~/laie/bla

File Edit View Search Terminal Help

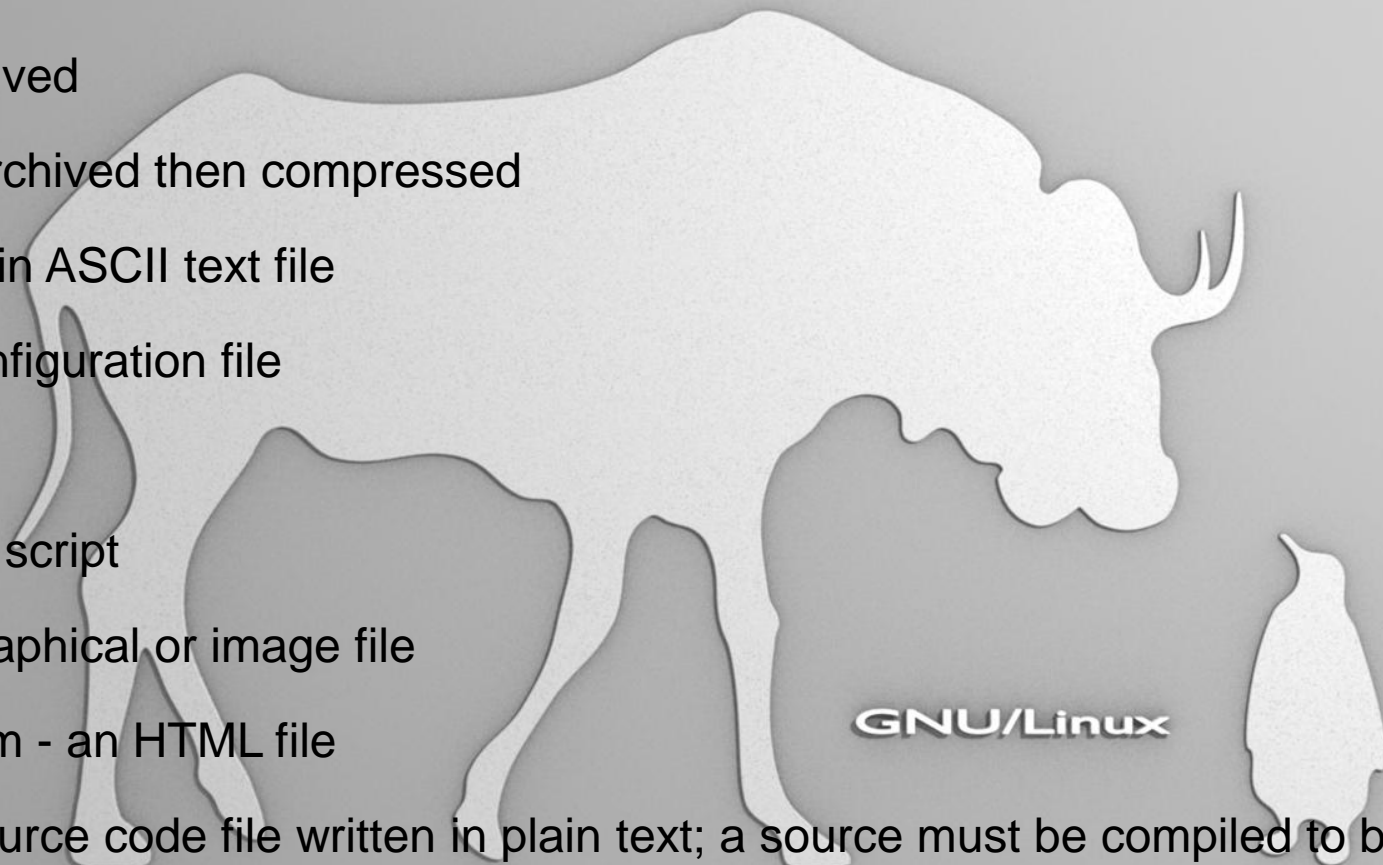
Hey!
I am
root
We are on
DATE:130912
and the time is
TIME:1047
Hm, I'm bored. Let's execute an update and an upgrade.
Get: 1 http://security.debian.org wheezy/updates InRelease [87.8 kB]
Get: 2 http://security.debian.org wheezy/updates/main Sources [937 B]
Get: 3 http://ftp.de.debian.org wheezy InRelease [207 kB]
Get: 4 http://security.debian.org wheezy/updates/main amd64 Packages [979 B]
Get: 5 http://ftp.de.debian.org wheezy/main Sources/DiffIndex [7,876 B]
Get: 6 http://ftp.de.debian.org wheezy/main amd64 Packages/DiffIndex [7,876 B]
Get: 7 http://ftp.de.debian.org wheezy/main Translation-en/DiffIndex [7,876 B]
Get: 8 http://ftp.de.debian.org wheezy/main Translation-fi/DiffIndex [7,819 B]
Get: 9 http://ftp.de.debian.org wheezy/main Translation-pt/DiffIndex [7,819 B]
.....
Ign http://security.debian.org wheezy/updates/main Translation-sk
Ign http://security.debian.org wheezy/updates/main Translation-km
Ign http://security.debian.org wheezy/updates/main Translation-nb
Ign http://security.debian.org wheezy/updates/main Translation-id
Ign http://security.debian.org wheezy/updates/main Translation-cs
Ign http://security.debian.org wheezy/updates/main Translation-ca
Ign http://security.debian.org wheezy/updates/main Translation-fr
Ign http://security.debian.org wheezy/updates/main Translation-sr
Ign http://security.debian.org wheezy/updates/main Translation-eo
Fetched 453 kB in 35s (12.8 kB/s)

No packages will be installed, upgraded, or removed.
0 packages upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B of archives. After unpacking 0 B will be used.

My job here is done
```

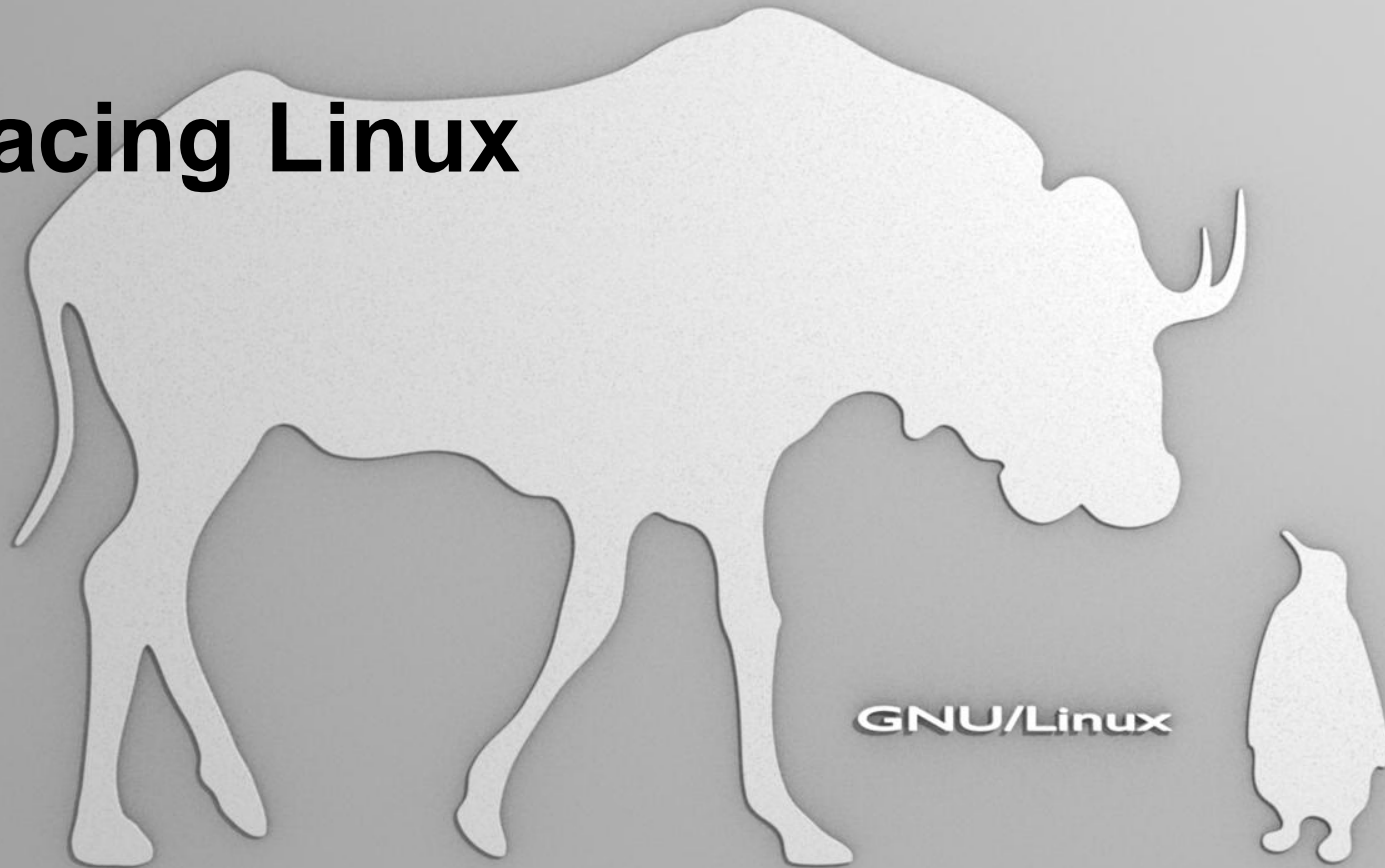
File extensions

These are suffixes used to indicate the encoding of content or usage.

- 
- .tar - archived
 - .tar.gz - archived then compressed
 - .txt - a plain ASCII text file
 - .conf - configuration file
 - .pdf
 - .sh - shell script
 - .jpg - a graphical or image file
 - .html / .htm - an HTML file
 - .src - a source code file written in plain text; a source must be compiled to be used
 - .iso - a image(copy) of a CD-ROM or DVD file
 - .odf + .odt + .odp - they are file extentions for **libreoffice** package; just like .ppt or 80
 - .docx for Mirosoft Office

Chapter VI

Facing Linux

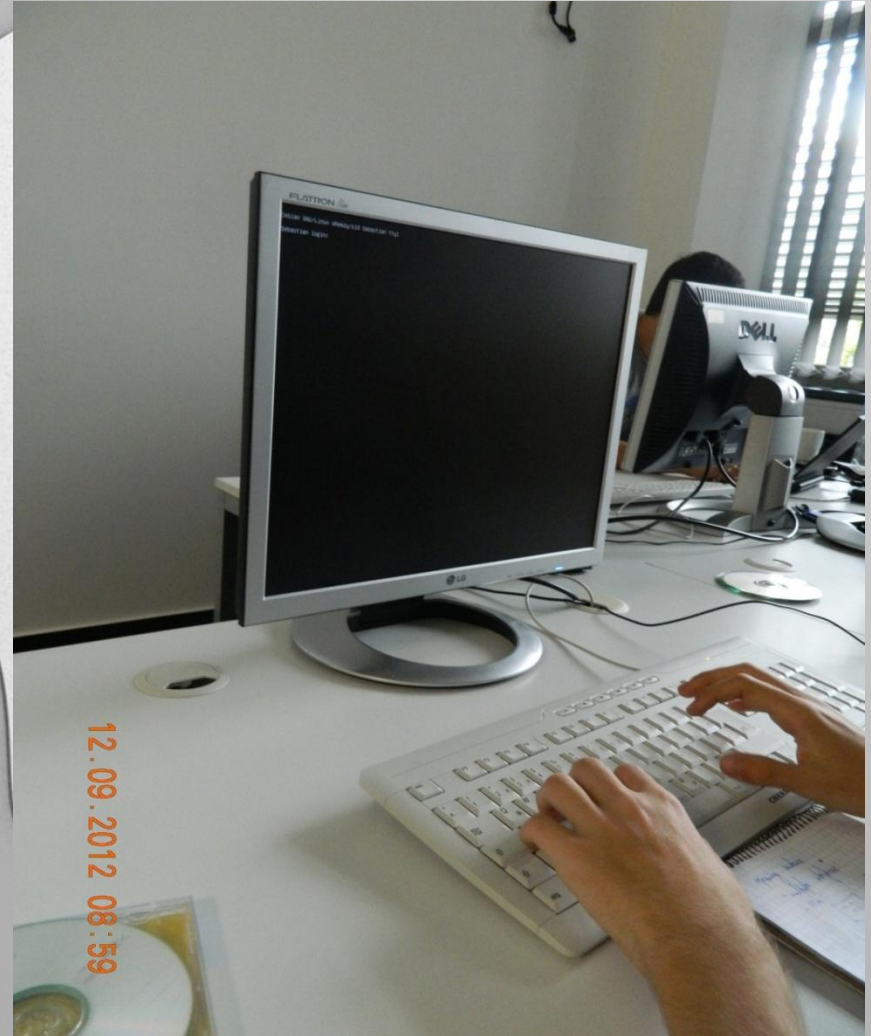


http://youtu.be/_tMv6zBOEpo



First Impressions (first week)

- When we first entered the room, there was the big screen in front of us and we thought about how simple it was just to follow the commands on it.
- Then we received the printed pages with the first installation commands and we thought it was going to be completely different from what it proved to be in the end.



And then we started to wait for all those programs to install and update and upgrade and we realized that we should not only copy those commands, but also understand them, not only type them, but also find out why they help us work with Linux.



Getting Used (second week)

During the second week, we began to understand the structure of a command and how it works. We made our first steps into the process of understand this new system of operation.

It was very useful for us that we could ask for help every time we had an error.



Starting to Enjoy (third week)



The last week of the project made us realize how useful and advantageous working with Linux really is. We should have been more open-minded in the previous days and not always consider Windows better than Linux just because we were much more familiar with the latter one. We also found out the complexity of computer usage.

While studying Linux, we discovered the way in which we can enjoy working with the computer. For example, an interesting thing we have done was getting into the other's computers with ssh command. This command helped us understand the similarities and differences between programs used with Windows (TeamViewer for example) and their equivalents in Linux (ssh).

```
File Edit View Search Terminal Help
iulia@IULIA:~$ ssh placinta@192.168.0.105
The authenticity of host '192.168.0.105 (192.168.0.105)' can't be established.
ECDSA key fingerprint is 61:f7:d9:e3:a3:53:31:96:de:6b:b1:4d:56:af:2a:29.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added '192.168.0.105' (ECDSA) to the list of known hosts.
placinta@192.168.0.105's password:
Linux Teodora 3.2.0-3-amd64 #1 SMP Mon Jul 23 02:45:17 UTC 2012 x86_64

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: Wed Sep  5 12:07:57 2012 from iulia.local
placinta@Teodora:~$
```

First!!

SUBMITTED BY BURUIANA ON MON, 09/10/2012 - 12:29

We're in Germany!

[Read more](#)[2 comments](#)[Log in](#)

or

[register](#)

to post comments

Creating Our First CMS

SUBMITTED BY IULIA ON MON, 09/10/2012 - 12:20

Really nice experience, taking into account the fact that everyone in our team has a different opinion on the site's color scheme.

[Read more](#)[2 comments](#)[Log in](#)

or

[register](#)

to post comments

Navigation

[Forums](#)[Recent content](#)

User login

Username *

Password *

- [Create new account](#)
- [Request new password](#)

[Log in](#)

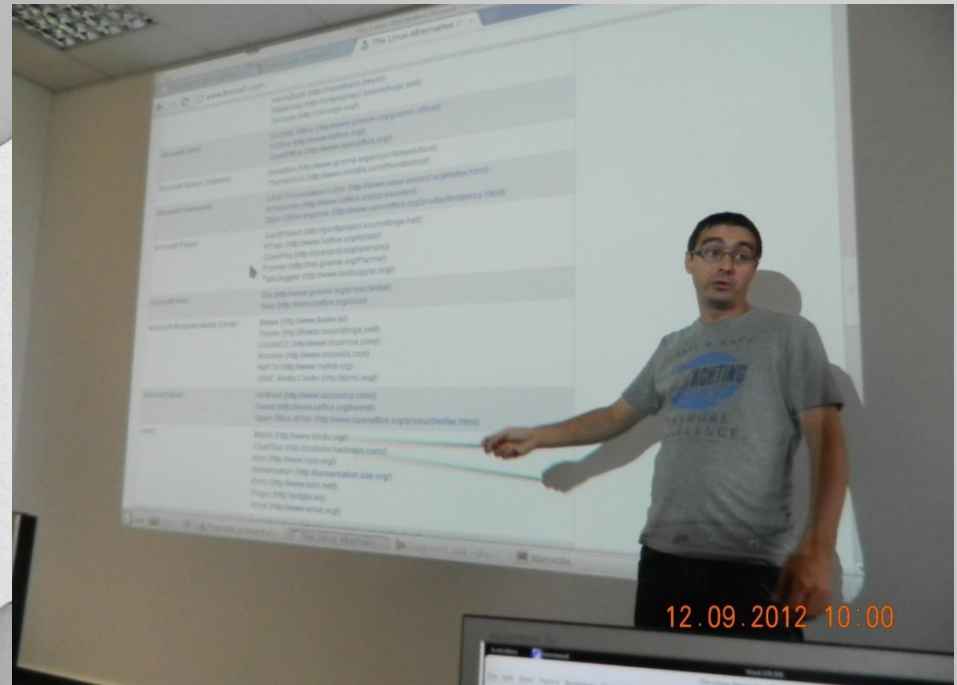
Opinions on the Project

Theodor Ivan: “I actually thought it would be much more difficult as it really was. I discovered it is quite easy because of the friendly interface and because of the fact that the commands can be easily learnt. I find it useful that now I am able to use two different operating systems!”

Bianca Miron: “In my opinion, it was a really interesting experience. Learning how to use something I’ve never seen before was hard at the beginning, but taking it step by step, it became accessible. I’ll look forward to my second meeting with Linux!”



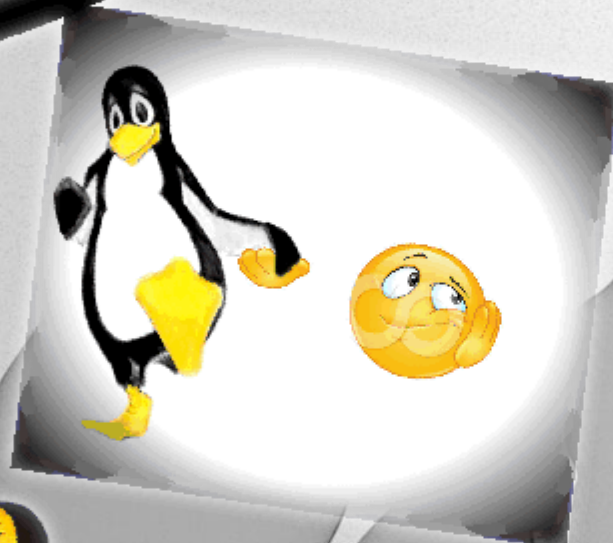
Review



We consider this project a real challenge because there are so many things to learn in such a short period of time. All in all, it was a very interesting experience which got the best out of us. We are grateful for having studied with a very good teacher who helped us understand Linux!



Week 1



Week 2



GNU/Linux

Week 3