Linux Practical Manual

with Exercises

Luis José Sánchez González

IES CAMPANILLAS. MÁLAGA. SPAIN
CORK INSTITUTE OF TECHNOLOGY. CORK. IRELAND

Copyright (C) 2009 – 2014 Luis José Sánchez González.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License" (Appendix A).

INDEX

TARGET AUDIENCE FOR THIS MANUAL

This manual is a PRACTICAL text for all those who want to learn to manage Linux from the terminal, using commands.

It is suitable for students in Computing at all levels: Degree in Computer Science, Engineering or Vocational Training. However, it may be useful for anyone who is simply curious or wants to learn to perform certain tasks with Linux beyond the graphical environment.

HOW TO CONTACT THE AUTHOR

You can reach **Luis José Sánchez González** by email at **luisjoseprofe@gmail.com**

Any suggestion will be welcome.

Both suggestions and notifications of possible errors in the content will be taken into account for future versions of this manual.

VERSIONS

The original manual was written in Spanish. The last updated was made in Cork (Ireland), the 12th May 2009.
This English version was translated from Spanish by the author and revised by Paul Walsh, lecturer in Cork Institute of Technology (thanks a lot Paul!). The last update was made in Málaga (Spain), the 31st July 2014.

CHAPTER 1

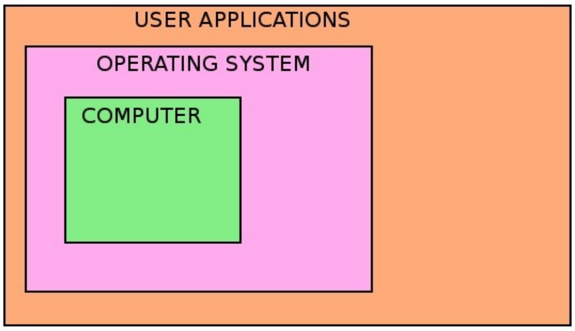
LINUX BASIC CONCEPTS

1.1 WHAT IS LINUX?

To understand what Linux is we first need to know what an operating system (you can use the acronym OS) is. We can give a simple definition of this concept:

> *"An operating system is a program that allows the user to interact with the computer and its components (monitor, hard disk, printer, etc) and help him in the completion of basic tasks such as copying or moving files from one location to another, editing text files, establishing a connection to the internet or doing backups".*

The operating system is the first program the computer executes when it is switched on.
At a higher level, we have programs that allow users to perform specific tasks. These programs are called **user applications**, or just applications. We can find many examples in the daily work with computers: accounting programs like Peachtree, word processors like OpenOffice.org Writer or MS Word, image manipulation programs like The Gimp or Adobe Photoshop.



GNU/Linux (or simply Linux) is one of the many existing O.S. but there are some features that make it special:

- **Free**: You can download it from the internet, you can copy and distribute it and you won't be a delinquent doing that. The terms under which this software can be copied and distributed are determined by the GNU License (www.gnu.org)

- **Done by volunteers**: Linux was not created for profit but to suit the needs of some users. Today it works in the same way. When someone needs a specific program, he simply creates it and puts it in the internet so that the community can use and modify it to suit their own purposes.

- **Multiuser**: Several users can connect and use a computer at the same time.

- **Multitask**: Several programs can work at the same time in a single machine..

- **Multiplatform**: There are Linux versions for many different platforms: all the PCs based on Intel or AMD processors, Digital/Compaq computers with Alpha processors, Apple computers, netbooks like Asus Eee and even mobile devices like Sharp Zaurus.

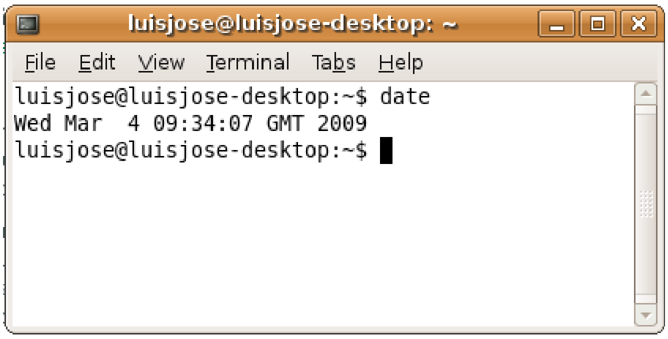
- **Stable**: Linux operating system is very mature and has been tested for a long time. There are many servers that have been running Linux for many years, 24 hours a day without a single crash.

- **Efficient**: Linux leverages the hardware resources. Even an old Pentium might work well with Linux and can be used as a mail server or a firewall.

- **There are loads of free programs**: There are many programs for Linux: word processors, image manipulation programs, 3D design programs, any kind of servers... and everything free, downloadable from the internet and installable from within the Linux environment.

1.2 WHY WORK IN CONSOLE MODE?

There are many names for it, we can say "working on a terminal", "using the command line", "working on the console", even some people prefer to say "working in text mode".
All these expressions refer to a way of working where, in order to perform a certain task, you must type in commands.


luisjose@luisjose-desktop: ~
File Edit View Terminal Tabs Help

```
luisjose@luisjose-desktop:~$ date
Wed Mar  4 09:34:07 GMT 2009
luisjose@luisjose-desktop:~$ 
```

For example, if you type the command **date** from a terminal window, the date and time of the current system is displayed. One could have found out the date by clicking the mouse on the system clock in the upper right corner of the screen, but that is true only if the system clock is there in the upper right corner and if this program is able to show a calendar.
Summarizing, you can obtain the time and date typing **date** in a terminal, and that works in the same way

in a big server machine in a space agency or in a domestic PC regardless of the version of Linux used. Incontrast, performing a task in a graphical environment can be very different even in similar computers with similar versions of Linux because these environments are very customizable, and sometimes you don't have the same menus or the same programs installed.

1.3 DIFFERENCES BETWEEN LINUX AND UNIX

At command line level, there are almost no differences. You can use the same commands both in Linux distributions and in Unix systems.
Even at the graphic environment level there are not many differences because both systems use X-Windows.
The main difference is that Linux is free while most Unix is not (sometimes they cost a lot).
Another important difference is the fact that there are versions of Linux for almost any platform, while Unix is usually very hardware-dependent.

1.4 DIFERENCES BETWEEN LINUX AND WINDOWS

The main difference is (again) the fact that Linux is free, while Windows is proprietary software and it costs money. In the same way, most Linux applications are free while those for Windows are not, although sometimes it's easy to find evaluation programs (shareware) for Windows that let you test them for a while for free.

1.5 USING LINUX WHITHOUT INSTALLING ANYTHING

You can execute Linux even while having another operating system installed on your computer, without installing anything on your hard disk. In fact, almost all the exercises contained in this manual could be done without installing Linux.

Here are some of the ways you can use Linux without installing it:

- **Live-CD**: A live-CD is a disk that allows you to boot Linux and execute programs from the disk. Almost all installation disks in modern Linux distributions can work as live-CD. This gives the opportunity to test first, and install after (if you like how it works).

- **Memory stick (*pen drive*)**: There are Linux versions especially light in terms of hardware requirements that can boot from a memory stick. You can find more information (detailed instructions about the installation process and customization) at this web-site: http://www.pendrivelinux.com/

- **Telnet**: This is a program that allows a user to establish a connection with another computer (a host) and work on the remote computer as if he was sitting in front of it. If that remote computer has Linux installed, you can work under Linux, even if your local machine has a different operating system.

1.6 WHAT IS A LINUX DISTRIBUTION?

A distribution of Linux is the operating system itself, which is usually called the **kernel**, together with an installation **program** and a **set of applications**, typically general purpose ones.

Distribution = Linux kernel + Installation Program + Applications

The first distributions were difficult to install but nowadays you just need to insert the CD and proceed by simply clicking on "next".

The following are some Linux distributions:

- **Ubuntu**: It is focused especially on desktop computers but also provides support for servers. It is based on Debian and its main features are ease of use and installation. A version is published every 6 months, one in April and another in October each year.

  Ubuntu is sponsored by Canonical, a private company founded and funded by the South African entrepreneur Mark Shuttleworth.The slogan of Ubuntu says "Linux for human beings." which summarizes its main goals: to make a Linux operating system more accessible and easy to use.

- **openSUSE**: This is a distribution sponsored by the companies Novell and AMD. The openSUSE project aims to make a distribution very easy to use and very easy to get, through Internet downloads and through physical retail outlets.

- **Mint**: Linux Mint is an Ubuntu-based distribution whose goal is to provide a more complete out-of-the-box experience by including browser plugins, media codecs, support for DVD playback, Java and other components. It is compatible with Ubuntu software repositories.

- **Fedora**: This comes from another famous distribution called Red Hat which included both free and proprietary software. The aim of Fedora is to build a complete, general purpose system, based exclusively on free software.

- **Debian**: It was Ian Murdock, in 1993, who started the Debian project and it was initially supported by the Free Software Foundation. It is perhaps the distribution that managed to maintain better the original GNU/Linux philosophy.
  For its stability and performance, it is often used in servers whose mission is critical.

- **Mandriva**: This is the successor of the popular Mandrake distribution, and it includes the KDE graphical environment. There is a wide range of versions of Mandriva: a completely free one, another one for booting from a memory pen, one for servers with multiple applications... The company in charge of this distribution (also named Mandriva) also offers technical support..

- **MEPIS**: The first version was written by Warren Woodford. MEPIS Linux is a Debian-based desktop Linux distribution designed for both personal and business purposes. It includes features such as a live installation and recovery CD, automatic hardware configuration, NTFS partition resizing, ACPI power management, WiFi support, anti-aliased TrueType fonts, a personal firewall, KDE, and much more.
  Their graphic environment is KDE and is suitable to those who use the computer as a workstation, for office tasks more than using it as a server. This distribution includes a good installation program with the possibility of repartitioning NTFS and recovering the operating system.

- **Sabayon**: This is a Gentoo based distribution, created and maintained by the Italian Fabio Erculiani.
  It emphasizes multimedia, graphics acceleration and the ability to run Windows programs with the Wine emulator. In this distribution you can find some 3D games like Battle of Wesnoth, Nexuiz and Warsow.

- **Slackware**: Created by Patrick Volkerding, Slakware is one of the oldest and most appreciated distributions. It tries to maintain a tradition of being easy to use and very stable. It includes ready-to-use servers out of the box: web, ftp and email.

1.7 HOW DID EVERYTHING START? A BRIEF HISTORY OF LINUX

The first Linux version was written by a Finish student called Linus Torvalds.
Linus was enrolled in the University of Helsinki in 1988 where he studied Computing Science. After buying a PC (based on Intel 386), he began using Minix, an operating system created by Andrew Tannenbaum for educational purposes. Linus was not too happy with this system. Lamenting the instability of the terminal emulator, which was used to connect to the computers of the university, Linus decided to make his own version of the terminal emulator program, independent of Minix. This was the first step taken in the creation of Linux.

Linus soon finished his terminal emulation program and thought it would be good to create other programs, e.g. A program to transfer files from one location to another.

In August 1991, Linus sent an email, which is already part of the history, to Usenet (a network of discussion like today forums), saying that he was working on this project.

Below is the original message:

```
From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: What would you like to see most in minix?
Summary: small poll for my new operating system
Message-ID: <1991Aug25.205708.9541@klaava.Helsinki.FI>
Date: 25 Aug 91 20:57:08 GMT
Organization: University of Helsinki


Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and
professional like gnu) for 386(486) AT clones.  This has been brewing
since april, and is starting to get ready.  I'd like any feedback on
things people like/dislike in minix, as my OS resembles it somewhat
(same physical layout of the file-system (due to practical reasons)
among other things).

I've currently ported bash(1.08) and gcc(1.40), and things seem to work.
This implies that I'll get something practical within a few months, and
I'd like to know what features most people would want.  Any suggestions
are welcome, but I won't promise I'll implement them :-)

               Linus (torvalds@kruuna.helsinki.fi)

PS.  Yes - it's free of any minix code, and it has a multi-threaded fs.
It is NOT protable (uses 386 task switching etc), and it probably never
will support anything other than AT-harddisks, as that's all I have :-(.
```

Linus released the first version of Linux, that is 0.01, in September 1991.

1.8 TUX, THE LINUX MASCOT

The mascot of the Linux operating system is a penguin called Tux.

There are many different versions of the origin of the term. The most widely accepted is that which states that it comes from the English word "**tux**edo", because it is the first thing that comes to mind when many people see a penguin.

Although some say it could also come from **T**orvalds **U**ni**x**.

The mascot was chosen by Torvalds himself inspired by a photo he found on the internet..

Tux is the main character in many Linux games like "Tux Racer", "Tux on the Run", "Super Tuxedo T. Penguin: A Quest for Herring", "Chromium B.S.U." or "Pingus".

SUMMARY OF CHAPTER 1

- An **operating system** is a program that allow the user to interact with the computer and its components and help him in the completion of basic tasks.

- Work with commands, in a terminal window, allows you to perform tasks in the same way in any Linux or Unix system.

- **Linux** is a free operating system and in most cases it costs nothing. It is written by volunteers. It is multiuser, multitasking and multiplatform. It is very stable and gets the best from a machine with few resources. Most Linux programs are also free.

- The main difference between Linux and **Unix** is that Linux is free and multiplatform while Unix use to be proprietary and very hardware dependent. **Windows** is also a proprietary operating system and the applications for this OS used to be also proprietary.

- You can use Linux without installing anything on the computer using one of these methods: live-CD, memory stick, telnet.

- A **distribution** consists of the operating system itself with the installation program and a set of applications. Some of the most important distributions are Ubuntu, openSUSE, Mint, Fedora, Debian and Mandriva.

- The first Linux version was created by Linus Torvalds in 1991 with the aim of improving MINIX, a UNIX like operating system used in the university.

- The Linux mascot is a penguin called **Tux**.

EXERCISES OF CHAPTER 1. BASIC CONCEPTS

The answers to the exercises can be found at the end.
Exercises are rated according to their level of difficulty:

🐧	Easy
🐧🐧	Medium
🐧🐧🐧	Dificult.

1. 🐧 An Operating System (OS) is a) a program that allows the user to do specific tasks, b) a word processor, c) a program that allows the user to interact with the computer and its componets or d) none of the previous answers are correct.

2. 🐧 A Linux distribution is a) the kernel of the OS, a program for the installation and a choice of different applications, b) the kernel of the OS, a graphic enviroment and a choice of different applications or c) the two previous answers are correct.

3. 🐧 Name a) any Linux distribution good in graphics and 3D games and b) any Linux distribution with preinstalled servers.

4. 🐧🐧 a) Which are the top 10 most popular Linux distributions in the last month? b) and in the last year?

5. 🐧 a) Who programmed the first Linux version?, b) with what goal?

6. 🐧 In order to execute Linux commands in a remote machine you need to install Linux in your local machine beforehand. a) True b) False.

7. 🐧 a) Was the first Linux version hardware-dependent or could it work on any machine? b) Is it the same nowadays?

8. 🐧 The Linux mascot is called a) Tuxedo, b) Tux, c) Pingu, or d) all of the above.

9. 🐧🐧 Who was the artist responsible for drawing the first version of the Linux mascot?

10. 🐧🐧 Name three light Linux distributions (with light hardware requirements) that can be installed on a memory stick.

CHAPTER 2

FILES AND DIRECTORIES (PART I)

2.1 LOGIN TO THE SYSTEM

To use Linux, the first thing to do is to identify yourself with a **user name** and **password**.

The user name cannot contain special characters such as punctuation (, ; :), slash (/), etc. The password should be reasonably long and difficult to guess. It is not a good idea to use your name, surname, telephone number, credit card number or the name of your pet as a password. If the password is short or easy to guess, someone can enter the system and delete or modify important information.

```
Ubuntu 8.04.2 ubuntu-desktop tty1

ubuntu-desktop login: luisjose
Password:
Linux ubuntu-desktop 2.6.24-23-generic #1 SMP Mon Jan 26 00:13:11 UTC 2009 i686

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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
Last login: Thu Mar  5 16:27:09 2009
luisjose@ubuntu-desktop:~$
```

The password is not shown on screen while typing. Be careful with upper and lower case, if the system says that the password is incorrect check that the “Caps Lock” is not activated.
After entering the user name and password, if the login process is done correctly, the system displays the prompt in the following format:

```
user_name@name_of_the_machine:~$
```

In this particular case, the user name is “`luisjose`”, the name of the machine is “`ubuntu-desktop`” and the character “`$`” indicates that the user connected is just a “normal” user. When a user has root privileges (superuser) the character “#” appears as we'll see later.

Linux is ready to execute commands! You can try the command “`date`”, seen as an example in the previous chapter.

```
luisjose@ubuntu-desktop:~$ date
Thu Mar  5 16:55:13 GMT 2009
luisjose@ubuntu-desktop:~$
```

2.2 DIRECTORY TREE

Imagine for a moment a lot of papers piled on the office desk: phone bills, invoices, notes taken in meetings, a bill of the car service... Finding a document in such a mess can be a nightmare if all these papers are mixed and without any order.
The solution to this mess is easy: you can use **folders**.
If you label the folders and put each paper in the correct one, everything will be perfectly arranged and neat. There may be a folder where you store the orders to suppliers, another to store all the notes taken at meetings, another one for bills ...
Moreover, there may be some sub-folders within folders. For example, the folder labeled "Bills" can have sub-folders labeled “Phone”, “Electricity”, “Car” ...

In a computer, the information is stored in the same way. Working in the graphical environment, we can talk about “folders” but working on the command line in a terminal window, we often call them “directories”, but conceptually they are exactly the same.

Below is a table with the most important directories of a Linux system:

/ root directory	**/bin**	Contains basic programs of the system.
	/boot	Contains the files needed to boot up the system.
	/dev	Files for devices: sound, printer, hard disk, cd/dvd unit, video, etc.
	/etc	Contains configuration files.
	/home	Home directory for users. Each user has his own directory inside /home/.
	/lib	Contains the shared libraries and the kernel modules
	/media	The devices are mounted inside this folder: CD-ROM, memory stick, removable hard disk, etc
	/opt	Directory reserved to install applications.
	/sbin	Contains programs for the superuser.
	/srv	Contains data for services provided by the system.
	/tmp	Temporary files directory.
	/usr	Contains most of the system files, applications, libraries, manuals, games, etc. It is a space shared by all users.
	/var	Contains administrative records and data that change frequently: error logging, databases, print queues, etc..
	/root	Home directory for the system administrator (root user).
	/proc	Data for the kernel and information about processes are stored here.

2.3 CONTENT OF A DIRECTORY. CHANGE DIRECTORY (pwd, ls, cd, mkdir)

2.3.1 pwd

The command `pwd` shows the current working directory, in other words, it tells the user where he is inside the directory tree. It is very useful when we are lost!

```
luisjose@ubuntu-desktop:~$ pwd
/home/luisjose
```

2.3.2 ls

The command `ls` shows the content of the current directory. By default, hidden directories are not shown. This is perhaps the most widely used command.

```
luisjose@ubuntu-desktop:~$ ls
Desktop  Documents  Examples  Music  Pictures  Public  Templates  Videos
```

You can add some options to `ls`, for example

```
ls -a
```

shows all the files, including those hidden (i.e. Where the name starts with a dot),

```
ls -l
```

shows a detailed listing, with the last modification date of every file, size, permissions, etc.,

```
ls -h
```

shows the size of the file in a readable format, that is bytes, Kb, Mb, etc.

You can see detailed information about all the command line options, both for `ls` and for any other command in the manual pages, using `man` followed by the command about which you want to obtain information:

```
luisjose@ubuntu-desktop:~$ man ls
```

This will give you detailed information about the command `ls`. To exit the manual pages just press "q".

2.3.3 cd

The command `cd` (change directory) allows you to change the current directory. Used alone, without any argument, takes you to your home directory. If used followed by a **path**, it changes to the directory indicated by the path.

```
luisjose@ubuntu-desktop:~$ pwd
/home/luisjose
luisjose@ubuntu-desktop:~$ cd /etc
luisjose@ubuntu-desktop:/etc$ pwd
/etc
```

In this case, the user was firstly in his home directory, and then "jumped" to the directory /etc. If you are curious about what is inside that directory, you can change the directory in the same way and type the command `ls`.

Paths can be **absolute** or **relative** ones. A path is absolute when it starts with the character "/" and relative when it starts with any other character.

In the previous example we have used an absolute path, that is, /etc. Lets see how to change to other locations using absolute paths:

```
luisjose@ubuntu-desktop:/$ cd /usr/local/
luisjose@ubuntu-desktop:/usr/local$ ls
bin  etc  games  include  lib  man  sbin  share  src
luisjose@ubuntu-desktop:/usr/local$ cd /var/spool/
luisjose@ubuntu-desktop:/var/spool$ ls
anacron  cron  cups  cups-pdf  mail  openoffice
```

We can say a relative path is a partial path. The effective path applied is the concatenation of the current path and the relative path written. Here is an example:

```
luisjose@ubuntu-desktop:/var/spool$ cd
luisjose@ubuntu-desktop:~$ pwd
/home/luisjose
luisjose@ubuntu-desktop:~$ cd Music
luisjose@ubuntu-desktop:~/Music$ pwd
/home/luisjose/Music
```

Remember that the command `cd` without any argument, take us to our home directory.

In this case,

```
cd Music
```

is equivalent to

```
cd /home/luisjose/Music
```

because the current path (`/home/luisjose`) is concatenated with the relative path written (`Music`)

Paths, both absolute and relative ones can be used with most commands, not only with `cd`.

For example, we can use a path as an argument for `ls`.

```
luisjose@ubuntu-desktop:~/Music$ ls /boot/grub/
default        installed-version  minix_stage1_5     xfs_stage1_5
device.map     jfs_stage1_5       reiserfs_stage1_5
e2fs_stage1_5  menu.lst           stage1
fat_stage1_5   menu.lst~          stage2
```

Two dots (..) refers to the directory just one level up.

```
luisjose@ubuntu-desktop:~/Music$ ls ..
Desktop  Documents  Examples  Music  Pictures  Public  Templates  Videos
```

`ls ..` shows the content of the directory `/home/luisjose` which is the directory one level up from `/home/luisjose/Music`

```
luisjose@ubuntu-desktop:~/Music$ cd ..
luisjose@ubuntu-desktop:~$ pwd
/home/luisjose
```

`cd ..` “jumps” up a step in the directory tree.

2.3.4 mkdir

You can create directories with the command `mkdir`. For example, consider a directory structure where a student can store information about his subjects according to the following scheme:

/home/luisjose	/Documents			
	/Desktop			
	/Pictures			
	/Music			
	/mathematics	**/course_01**	**/algebra**	**/old_exams** **/notes**
			/analysis	
			/physics	**/exercises_books** **/videos**
			/computing	**/pascal_compiler**
	/Video			

To create this structure the student has to do the following:

```
~$ mkdir mathematics
~$ cd mathematics/
~/mathematics$ mkdir course_01
~/mathematics$ cd course_01/
~/mathematics/course_01$ mkdir algebra analysis physics computing
~/mathematics/course_01$ ls
algebra  analysis  physics  computing
~/mathematics/course_01$ cd algebra/
~/mathematics/course_01/algebra$ mkdir old_exams notes
~/mathematics/course_01/algebra$ cd ..
~/mathematics/course_01$ cd physics
~/mathematics/course_01/physics$ mkdir exercises_books
~/mathematics/course_01/physics$ mkdir videos
~/mathematics/course_01/physics$ cd ..
~/mathematics/course_01$ cd computing/
~/mathematics/course_01/computing$ mkdir pascal_compiler
```

Notice we haven't used the full prompt (with the user name and machine) this time. We won't use it any more in this manual unless necessary.

2.4 VISUALIZING FILES (cat, more, less, head, tail)

The commands `cat`, `more` and `less` are used to show the contents of text files. The difference lies in how they display the content. You have to specify the name of the file as an argument. You can also write the path, if the file you want to show is not in the current directory.

The command `cat` shows the contents of a file and, when it ends, the user is again at the command line prompt.
For example,

```
~$ cat /var/log/dmesg
```

shows the content of the file `dmesg` which is inside the directory `/var/log`. If you try to do it by yourself, you will notice that is impossible to see all the text in this file, because everything passes on the screen very quickly. That's why `cat` is most used to visualize the content of little files.

The command `more` does the same as `cat`, but now the content of the file is shown screen by screen, that is, the text is written until the screen is full and then the system waits for the user to press <space> to continue by showing the next text page:

```
~$ more /var/log/dmesg
```

The command `less` is the most versatile of the three, since it allows you to move forward and backward within the file, using the cursor keys or "PageUp" and "PageDown" keys:

```
:~$ less /var/log/dmesg
```

At any time, visualization can be interrupted to return to the prompt by pressing “q” key.

The commands `head` and `tail` allow you to partially show the content of a file. As its name suggests, head displays the first lines of the file and tail shows the last lines.

Some examples:

```
~$ head /boot/grub/menu.lst
# menu.lst - See: grub(8), info grub, update-grub(8)
#            grub-install(8), grub-floppy(8),
#            grub-md5-crypt, /usr/share/doc/grub
#            and /usr/share/doc/grub-doc/.

## default num
# Set the default entry to the entry number NUM. Numbering starts from 0, and
# the entry number 0 is the default if the command is not used.
#
# You can specify 'saved' instead of a number. In this case, the default entry

~$ tail /boot/grub/menu.lst
root         (hd0,0)
kernel       /boot/vmlinuz-2.6.24-19-generic   root=UUID=409e68a1-6123-476f-abf7-
042854b68f3c ro single
initrd       /boot/initrd.img-2.6.24-19-generic

title        Ubuntu 8.04.2, memtest86+
root         (hd0,0)
kernel       /boot/memtest86+.bin
quiet

### END DEBIAN AUTOMAGIC KERNELS LIST
```

By default, both `head` and `tail` show 10 lines, but this behaviour can be changed with the command line option -n.

```
~$ tail -n4 /boot/grub/menu.lst
kernel       /boot/memtest86+.bin
quiet

### END DEBIAN AUTOMAGIC KERNELS LIST
```

In this case, only 4 lines have been shown.

2.5 EDITING FILES (touch, vi, ee, mcedit)

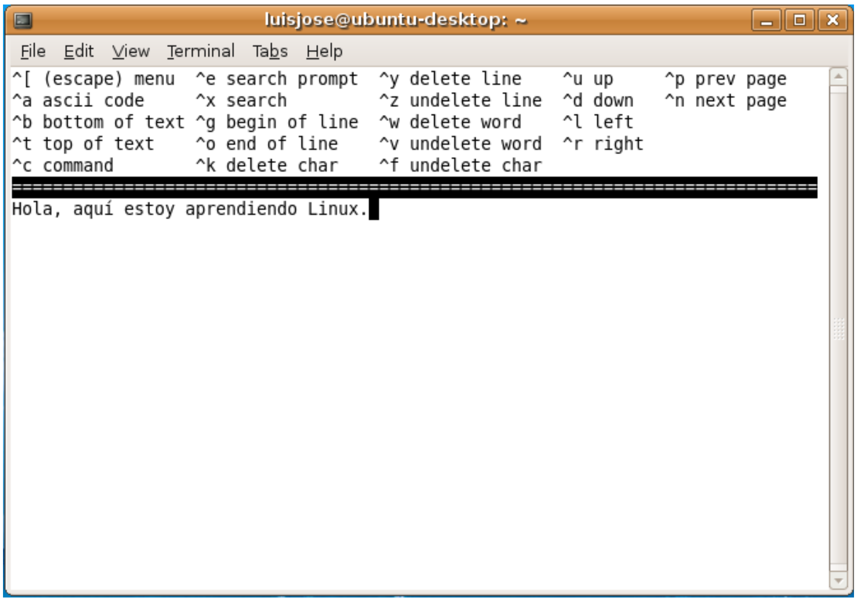
The command `touch` allows you to create an empty file. You can do the same with any text editor, but doing it with `touch` is especially easy and fast.

```
~$ ls
Desktop  Documents  Examples  Music  Pictures  Public  Templates  Videos
~$ touch prueba.txt
~$ ls
Desktop    Examples   Pictures    Public     Videos
Documents  Music      prueba.txt  Templates
~$ cat prueba.txt
~$
```

You can see how we have created the file `prueba.txt`. If we try to visualize its content, with the command `cat`, there is nothing on the screen, therefore it's empty.

The program `ee` is a rudimentary editor, but at the same time, effective. We can edit the previous file and write a sentence:

```
~$ee prueba.txt
```



```
^[ (escape) menu  ^e search prompt  ^y delete line    ^u up     ^p prev page
^a ascii code     ^x search         ^z undelete line  ^d down   ^n next page
^b bottom of text ^g begin of line  ^w delete word    ^l left
^t top of text    ^o end of line    ^v undelete word  ^r right
^c command        ^k delete char    ^f undelete char
==============================================================================
Hola, aquí estoy aprendiendo Linux.
```

Pressing ESC, the user can exit to the main menu and save the file. We can see now the content of the file:

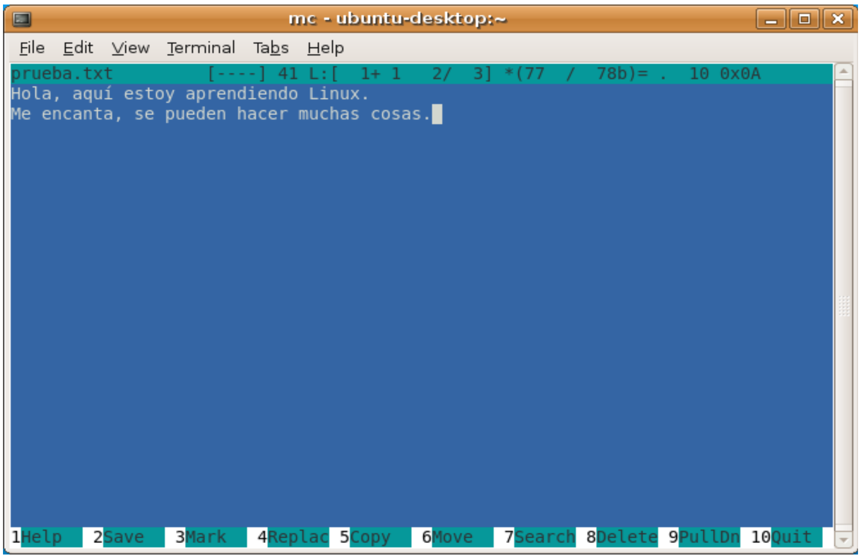
```
~$ cat prueba.txt
Hola, aquí estoy aprendiendo Linux.
```

The program `nano` is another simple text editor. If you are curious, you can test it and compare it with `ee`. If some of these programs are not installed on the machine, you can install them by typing `sudo apt-get install` followed by the name of the program you want to install. For example, if we want to install `ee`:

```
~$ sudo apt-get install ee
```

The program `mcedit` is a text editor that is a bit more sofisticated than `ee` or `nano` (at least apparently) and is a part of `mc` (Midnight Commander), which is software that is very similar to the famous Norton Commander for MS-DOS. Now we are going to modify the file `prueba.txt`, but before that, we have to install `mc`, because it is not installed by default:

```
~$ sudo apt-get install mc
~$ mcedit prueba.txt
```


mc - ubuntu-desktop:~
File Edit View Terminal Tabs Help
1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn 10Quit

```
prueba.txt          [----] 41 L:[  1+ 1   2/  3] *(77  /  78b)= .  10 0x0A
Hola, aquí estoy aprendiendo Linux.
Me encanta, se pueden hacer muchas cosas.
```

By typing F2, we can save changes and by typing ESC key twice (or pressing F10 key) we can exit the program.

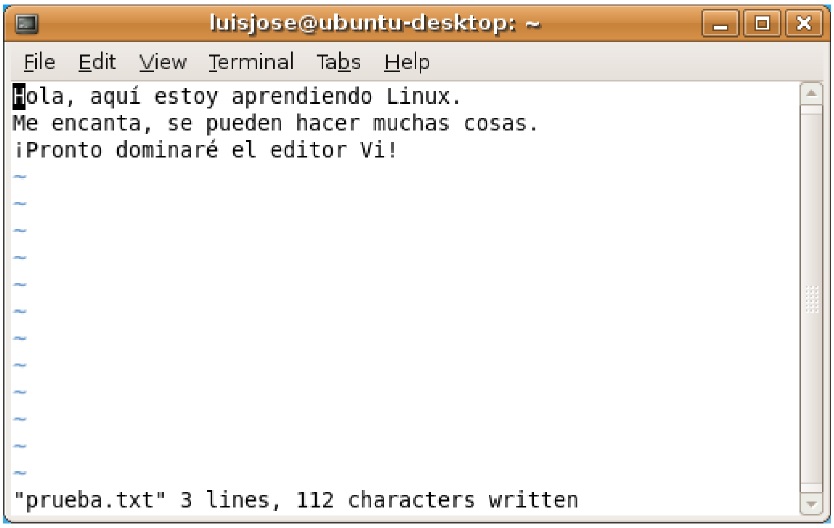
We can see that all the changes in the file have been saved properly:

```
~$ cat prueba.txt
Hola, aquí estoy aprendiendo Linux.
Me encanta, se pueden hacer muchas cosas.
```

We have left for the last the Linux editor par excellence, `vi`. At first sight it seems the most difficult to use, which is true, and it seems also that it has just a few options and capabilities but, on the contrary, it is a very powerful program. Any self-respecting Linux user must know how to use this program.
We'll add a new text line to the file `prueba.txt`. To do so, we'll follow these steps:

```
~$ vi prueba.txt
```

- Press "i" to enter "editing" mode.
- Write the text.
- Press ESC key to exit from "editing" mode.
- Type ":" + "w" + ENTER to save the changes.
- Type ":" + "q" + ENTER to exit the program.



```
Hola, aquí estoy aprendiendo Linux.
Me encanta, se pueden hacer muchas cosas.
¡Pronto dominaré el editor Vi!
~
~
~
~
~
~
~
~
~
~
~
~
"prueba.txt" 3 lines, 112 characters written
```

We can check once again that everything is properly saved:

```
~$ cat prueba.txt
Hola, aquí estoy aprendiendo Linux.
Me encanta, se pueden hacer muchas cosas.
¡Pronto dominaré el editor Vi!
```

It is highly advisable to do the tutorial called `vimtutor`.

SUMMARY OF CHAPTER 2

- Any user needs a **user name** and a **password** to enter the system.

- The information is stored in files inside **directories** and **subdirectories** (folders and sub-folders).

- There are several predefined directories like /bin, /dev, /home, /etc, /var, etc. in all Linux systems.

- There are **absolute paths**, starting with the character "/", which define a complete path and **relative paths**, not starting with "/", which a path that is the concatenation of the current directory with the relative path.

- These are the commands studied in this chapter:

Command	*Action*	*Example*
pwd	shows the current directory	pwd
ls	lists files and folders	ls -l
cd	changes to a certain directory	cd mp3/wim_mertens
mkdir	creates one or several directories	mkdir letters bills
cat	visualizes the contents of a file	cat /var/log/dmesg
more	visualizes the contents of a file screen by screen	more /var/log/dmesg
less	visualizes the contents of a file screen by screen, and it lets you go up and down through the text	less /var/log/dmesg
head	shows the first lines of a file	head -n5 /var/log/dmesg
tail	shows the last lines of a file	tail /var/log/dmesg
touch	creates an empty file	touch phones.txt
ee	very simple text editor	ee phones.txt
mcedit	text editor, part of the software called Midnight Commander	mcedit phones.txt
vi	very powerful text editor	vi phones.txt
apt-get	install and uninstall programs	apt-get install mc
man	shows help about a certain command	man ls

EXERCISES OF CHAPTER 2. FILES AND DIRECTORIES (PART I)

The answers to the exercises can be found at the end.
Exercises are rated according to their level of difficulty:

🐧	Easy
🐧🐧	Medium
🐧🐧🐧	Dificult.

1. 🐧 In which directory are the configuration files of the system?

2. 🐧 To login to a Linux system requires a) user name, password and IP address, b) username and password c) only a password.

   In the following exercises you should use the required Linux commands to perform the operations described.

3. 🐧 Display the content of the current directory.

4. 🐧 Display the contents of the directory which is just at an upper level.

5. 🐧 🐧 In wich day of the week you were born?, use the `cal` command to find it out.

6. 🐧 Display the files in the `/bin` directory.

7. 🐧 Assuming you are in your home directory (`/home/username`), display the contents of the /`usr/bin` directory a) with just one command line, b) moving step by step through the directory tree and c) with two command lines.

8. 🐧🐧 Display all the files in the `/etc` directory and all the files in the subdirectories recursively (with just a one-line command).

9. 🐧🐧 Display all the files in the `/usr/bin/X11` directory ordered by size (from highest to lowest). Only the name of each file must be shown, without any other additional information.

10. 🐧🐧 Show all the files in the `/etc` directory ordered by size (from highest to lowest) together with other characteristics, i.e. permissions, size, date of last modification, etc. The size of each file should appear in a "readable" format, expressed in Kb, Mb, etc.

11. 🐧🐧🐧 Show all the files in the `/bin` directory ordered by size (from lowest to highest). Only the name and the size of each file must be shown, without any other additional

information. The size of each file must be shown in a “readable” format, expressed in Kb, Mb, etc.

12. 🐧 Show the contents of the root directory using an absolute path as an argument of the `ls` command.

13. 🐧🐧 Show the contents of the root directory using a relative path as an argument of the `ls` command. Assume that the current directory is /home/elena/documents.

14. 🐧 Create the directory `expenses` inside the home directory.

15. 🐧 What happens if you try to create a directory inside `/etc`?

16. 🐧 Display the contents of the `/etc/fstab` file.
.

17. 🐧 Display the first 10 lines of the `/etc/bash.bashrc` file.

18. 🐧 Create the following directory structure inside your home directory:

```
                    multimedia
                        |
   -----------------------------------------------
   |             |              |              |
 music       pictures         video          slides
                 |
          --------------
          |            |
      personal       other
```

19. 🐧 Create an empty file inside the `music` directory named `favourite_styles.txt`

20. 🐧 Use your favourite text editor to edit `favourite_styles.txt`, then type in the music styles you like most. Save the changes and exit.

21. 🐧 Display the contents of the file `favourite_styles.txt`.

22. 🐧 Show the first 3 lines of the file `favourite_styles.txt`.

23. 🐧 Show the last line of the file `favourite_styles.txt`.

24. 🐧🐧🐧 Show the content of the file `favourite_styles.txt` excluding the first line`.` Assume that the exact number of lines in the file is unknown.

CHAPTER 3

FILES AND DIRECTORIES (PART II)

3.1 WILDCARDS

Very often, it is necessary to perform actions on multiple files or directories at once.
For example:

```
$ cat doc1 doc2 doc3 doc4 doc5 doc6
```

You can use **wildcard** patterns to avoid writing the name of every single file.
To list all the files with names starting with “doc” followed by a number from 1 to 6, you can use a wildcar pattern like this:

```
$ cat doc[1-6]
```

If you simply want to list all the files with names starting with “doc”, you can type this:

```
$ cat doc*
```

where the character “*” matches any combination of characters, even the empty string. If a file named just doc exists in the current directory, it will be also shown.
The character “*” can be placed at any position. For example, to show all the files with names starting with the letter “a” and ending with the letter “s” in the directory /usr/bin you can type this:

```
$ ls /usr/bin/a*s
```

The symbol “?” matches any single character. For example, the next command lists all files in the directory /usr/bin with names starting with “g”, followed by any single character, followed by “o” and ending with any combination of characters including the empty string.

```
$ ls /usr/bin/g?o*
```

At the beginning of this chapter, we had an example of a wildcard string that included square brackets. The bracket wildcard is more or less like “?” but it allows one to be more specific. You can place any characters you'd like to match inside the brackets. You can also specify a range. For example [adfg] match any of the characters a, d, f or g. The pattern [Hh]ello matches both “Hello” and “hello”. [a-z]* matches any string starting with a lower case letter.

3.2 COPYING AND DELETING FILES (cp, mv, rm)

3.2.1 cp

The command **cp** allows you to copy files. You can copy a single file or many. You can copy both files and directories and, of course, you can use wildcard patterns.
There are three items involved in the copy process: what is copied, the source path and the target path. It is worth recalling that the paths can be both absolute and relative ones. The source path is specified along with whatever you want to copy. Here is an example:

```
$ cp /etc/hosts /home/student/tests/
```

The command above copies the file “hosts”, which is located in /etc directory into the /home/student/tests/ directory.

If the source path is not specified, current directory will be taken as default.
For example:

```
$ cp *.odt texts/
```

copies all files with odt extensions from the current directory to the texts directory.
If you want the target path to be the current directory, you can use the dot “.”
For example:

```
$ cp /usr/bin/g* .
```

copies all files starting with “g” in /usr/bin directory into the current directory.

3.2.2 mv

The command **mv** can be use in two different ways, to move or to rename files. You can perform these two actions separately or you can perform them at once.

For example:

```
$ mv my_text.txt letter.txt
```

changes the name of the file `my_text.txt` to the new name `letter.txt`.

While

```
$ mv letter.txt Documents/
```

move `letter.txt` into the `Documents` directory.

As mentioned above, you can do both at once, move and rename:

```
~$ cd Documents/
~/Documents$ mkdir mail
~/Documents$ mv letter.txt mail/letter01.txt
```

In this case, file `letter.txt` has been moved into `~/Documents/mail` and it has changed its name to `letter01.txt`

3.2.3 rm

The command **`rm`** is used to delete files. You have to take into account that the files are not sent to a trash bin, so THEY CAN NOT BE RETRIEVED ONCE DELETED.

Example:

```
$ rm *.txt
```

This command deletes all files with a `txt` extension in the current directory.

3.3 COPYING AND DELETING DIRECTORIES (cp, mv, rm)

Just as you can copy, move or delete files, you can copy, move or delete directories. You have to take into account that a directory can contain many files, and even other directories and more files inside these directories. Therefore, if you want to copy a directory, with all the files and directories inside, you must indicate it with -R. This is called "a recursive copy".

Example:

```
~$ mkdir multimedia2
~$ cp multimedia/* multimedia2
cp: se omite el directorio «multimedia/pictures»
cp: se omite el directorio «multimedia/music»
cp: se omite el directorio «multimedia/slides»
cp: se omite el directorio «multimedia/video»
~$ ls multimedia2
~$
```

We have tried to copy the content of `multimedia` directory into `multimedia2`, but no file has been copied, what happened? Simply we haven't perform a recursive copy (with -R option). We tried to copy all the things in the immediate first level inside `multimedia` but not at a deeper level. Because there weren't files at that level, nothing has been copied.

Lets try now in the recursive way:

```
~$ cp -R multimedia/* multimedia2
~$ ls -R multimedia2
multimedia2:
pictures  music  slides  video

multimedia2/pictures:
misc  personal

multimedia2/pictures/misc:

multimedia2/pictures/personal:

multimedia2/music:
favorite_styles.txt

multimedia2/slides:

multimedia2/video:
```

As you can see, both the directory tree and its contents have been copied.

The command `mv` works like `cp`, just move instead of copy. When it is used to rename directories, it works exactly in the same way as it works with files.

Example:

```
~$ mv multimedia2 multimedia_copy
```

This change the name of `multimedia2` directory to the new name `multimedia_copy`. You can check by yourself that the content of that directory remains the same.

You can use `rm` to delete directories.

```
~$ rm multimedia_copy/
rm: no se puede borrar «multimedia_copy/»: Es un directorio
```

We got an error, what happened? If you are clever enough, you could solve the problem...
...Thats it, you have to delete recursively:

```
~$ rm -Rf multimedia_copy/
```

We wrote `-R` option, and also added `-f` option which avoids confirmation checks for each file to be deleted.

SUMMARY OF CHAPTER 3

- Use of wildcards:

Example	*Matching*
`*`	Any string.
`*f*`	Any string containing the character “f”.
`z*`	Any string starting with “z”.
`a?`	A string with two characters, the fist one must be the letter “a” and the second one can be any character.
`[Dd]ocument`	Matches “Document” and “document.”
`A[a-z][0-6]`	A three character string. The first character is “A”, the second one is any lower case and letter and the third one is a digit from 0 to 6.

- We have seen the following commands in this chapter:

Command	*Action*	*Example*
cp	copies files and directories	cp *.txt mail/
mv	moves or rename files and directories	mv words.txt text.txt
rm	deletes files and directories	rm -R things/rubbish
rmdir	deletes directories	rmdir old

EXERCISES OF CHAPTER 3

The answers to the exercises can be found at the end.
Exercises are rated according to their level of difficulty:

🐧	Easy
🐧🐧	Medium
🐧🐧🐧	Dificult.

In the following exercises, you should use the required Linux commands to perform the operations described.

1. 🐧 Show all the `jpg` pictures in the current directory.

2. 🐧 Display all the files in the directory `/usr/bin` starting with letter “`j`”.

3. 🐧🐧 Show all the files in the directory `/usr/bin` starting with the letter “`k`”, with an “`a`” in the 3rd place.

4. 🐧 Show all the files in the directory `/bin` ending with “`n`”.

5. 🐧 Display all the files in the directory `/etc` and all the files in every subdirectory recursively.

6. 🐧🐧 In your home directory, create another directory named `test`. Copy the file `gzip` from the directory `/bin` to `test`. Create a duplicate of `gzip` named `gzip2` inside `test`.

7. 🐧🐧 Change the name of the directory `test` to `test2`. Create `test3` at the same level in the directory tree as `test2` and move all the files from `test2` to `test3`. Delete `test2`.

8. 🐧🐧 Create an empty file named “*?Hello all?*”. Can you? Is it a good idea to name a file this way? Explain your answer.

9. 🐧🐧🐧 Create a directory named `multimedia_test` and copy all the content from the directory `multimedia into it`. Then, create two files in `multimedia/video/`, one named `films.txt` and another named `actors.txt`. Edit the file `films.txt` and write the title of your favourite movie. Then, create another file in `multimedia_test/video/`, also named `films.txt`. Edit this file and now write the titles of your five favourite movies. Copy of all the content of `multimedia` into `multimedia_test ensuring that only most recently modified versions of files remain.`To check that everything is ok, check if `multimedia_test/video` contains the empty file `actors.txt` and the file `films.txt` contains 5 titles and not 1.

10. 🐧🐧 Delete the directory `multimedia/pictures/others`. The system must ask for confirmation.

11. 🐧🐧 Move the file `films.txt`, which is in `multimedia/video`, to the directory above, at the same time renaming the file to `my_films.txt`.

CHAPTER 4

GROUPS, USERS AND PERMISSIONS

4.1 WHY THERE ARE GROUPS, USERS AND PERMISSIONS

We have seen that files are organized inside directories (folders) in order to have the system neat. Moreover, is easy to find files when each one is in its place

Lets go back again to our office. Each document is in the correct place, there are folders and sub-folders and everything is organized. The accountant must have access to the folders with bills and receipts, but must not have access to the information on personnel, product development or marketing. In a Linux system, files and directories work in the same way. For example, files in `/etc` directory only can be modified by the system administrator. This prevents unwanted users from entering the system and changing or damaging critical information.

4.2 WHAT IS THE SUPERUSER?

The superuser, system administrator or simply the `root`, is a special user with privileges to change the system configuration, delete and create files in any directory, create new groups and users, etc.

> **IMPORTANT:** WORKING AS SUPERUSER MAY BE DANGEROUS, THE SYSTEM CAN BE IRREVERSIBLE DAMAGED. YOU HAVE TO BE SURE OF WHAT YOU ARE DOING WHEN WORKING AS SUPERUSER.

After this note, let's do something as `root`:

```
$ touch /etc/test.txt
touch: no se puede efectuar `touch' sobre «/etc/test.txt»: Permiso denegado
$ sudo touch /etc/test.txt
$ ls /etc/pru*
/etc/test.txt
```

We tried first to create the file `test.txt` in `/etc` directory as normal user and then we got an error: “Permission denied”, which means a user without privileges can not perform that action. Then we tried as superuser. To do so, we used `sudo` command. This time we managed to do what we wanted. Notice we prompted the password. This time we managed to do what we wanted. The system asked for the password, that is important because if it did not, anybody could execute commands as administrator and could therefore damage the system.

4.3 PERMISSIONS

Information on groups, users and permissions of files can be obtained using the `ls` command with the `-l` option. Let's see the permissions of a file named `whatis` placed in `/usr/bin` directory.

```
$ ls -l /usr/bin/whatis
-rwxr-xr-x 1 root root 87792 2008-03-12 14:24 /usr/bin/whatis
```

The first column shows the **permissions**, the third shows the **user** (in this case it is the system administrator) and the fourth column shows the name of the **group** (in this case it is the same as the user).

Let's see the meaning of these characters in the first column:

-	r	w	x	r	-	x	r	-	x
File type.	Permissions for the owner of the file			Permissions for the group to which the file belongs			Permissions for the rest of users		

r	**Read** permission.
w	**Write** permission
x	**Execution** permission

These are the file types:

File types	
l	Symbolic link.
c	Special character device.
b	Special block device.
p	FIFO (data structure).
s	Socket (communications).
-	None of the previous. Can be a text file, a binary file, etc.

In our last example we have the character "-" indicating the file type, because the file (/usr/bin/whatis) is a binary file (an executable program). The permissions for the owner are indicated by `rwx`, which means permissions for reading, writing and executing the file. Permission for writing means the user can delete, rename or edit the file. Both the group and the rest of the users have the permissions indicated by `r-x`, which means they can use the file (they can read or execute) but they can not modify it.

4.4 WHO WE ARE (whoami, groups)

Before starting creating users, groups and changing permissions, we must know who we are and to which group we belong. Although, we login to the system as a certain user, we can use `su` to execute commands as if we were a different user. Of course, we must know the password of this user.

```
$ whoami
luisjose
$ su student
CPassword:
$ whoami
student
```

To return to the original user, you can just type `exit`.

```
$ whoami
student
$ exit
exit
$ whoami
luisjose
```

With the `groups` command, you can see to which group or groups you belong.

```
luisjose@luisjose-xps1330:~$ groups
luisjose adm dialout cdrom floppy audio dip video plugdev scanner lpadmin admin
netdev powerdev sambashare
```

You can specify one or more users after `groups`. That will tell us to which group/s each user belongs.

```
luisjose@luisjose-xps1330:~$ groups student root
student : student
root : root
```

4.5 GROUP MANAGEMENT (groupadd, groupdel, groupmod)

The commands `groupadd`, `groupdel` and `groupmod` allows you to create, delete and modify groups respectively.

Let's create three groups named office_malaga, office_jaen and office_madrid

```
$ groupadd office_malaga
groupadd: incapaz de bloquear el fichero de grupos
$ sudo groupadd office_malaga
$ sudo groupadd office_jaen
$ sudo groupadd office_madrit
```

If we try to create a group when logged in as user without privileges we get an error. To manage groups and users, you need to execute commands as superuser, therefore we have to type `sudo` before the command.

We have written the name of the second group incorrectly (madrit instead of madrid), no panic! We can fix the problem with `groupmod`.

```
$ sudo groupmod -n office_madrid office_madrit
```

The company decided to close the office in Jaén due to cost reduction and transfer the resources to the office in Málaga, so we don't need the group `office_jaen` anymore. You can delete it using `groupdel`.

```
$ sudo groupdel office_jaen
```

4.6 USER MANAGEMENT (adduser, userdel, usermod)

User management, just like group management, requires commands to be executed with root privileges. You can type `sudo` before each command, or you can do simply this:

```
$ sudo bash
#
```

Notice the prompt has changed. Now the “#” character is shown instead “$”. From now, all the commands will be executed with root privileges. Remember to exit the root mode type `exit` when you finish all the tasks you want to perform as root.

We want to set up two users in the `office_malaga` group and another one in `office_madrid`. There will be another user coming and going between the two offices so we'll set up this user in both groups.

```
# adduser pedro --ingroup office_malaga
# adduser ana --ingroup office_malaga
# adduser berta --ingroup office_madrid
# adduser laura --ingroup office_malaga
# adduser laura office_madrid
```

We have killed two birds with one stone. We created the users and, at the same time, they were included in their corresponding groups. These two steps can also be performed separately.
The user `laura` belongs to two groups. We created the user ana and, at the same time, this user was added to the group named `office_malaga` with the option `–ingroup`. To add an existing user to a group, you can use `adduser` without any command line option.

```
# groups ana berta laura
ana : office_malaga
berta : office_madrid
laura : office_malaga office_madrid
```

Notice we have used **`adduser`** and not `useradd`. The latter is considered a low level command and is recommended to use the first one.

When creating users, we have been asked to type the passwords. We can change them later on with the `passwd` command.

```
# passwd pedro
# passwd ana
# passwd laura
```

Don't forget to exit root mode with the `exit` command when system administrator privileges are not required.

```
# exit
```

From now on, the prompt “$” will indicate that the user is working as a normal user, and the prompt “#” will indicate that the user is “in root mode” (or that he is really the root user).

Note that a default directory is created inside `/home` directory for each user. When a user login to the system, he “lands” in that directory. This directory is commonly known as the working directory.

```
$ ls /home/
student  ana  berta  ftp  laura  luisjose  pedro
```

4.7 CHANGING GROUP AND OWNER (chown, chgrp)

Let's assume the file `report.txt` was created by the user named `pedro`. By default, the owner of a file is the same user that created the file, the user `pedro` in this case. Remember that user `pedro` belongs to the group named `office_malaga`.

```
$ su pedro
$ cd
$ pwd
/home/pedro
$ touch report.txt
$ ls -l
```

```
-rw-r--r-- 1 pedro office_malaga  0 2009-03-19 12:46 report.txt
```

We can change almost all of these things. Let's move this file to `laura`'s working directory and let's change the owner also.

```
# mv report.txt /home/laura/
# cd /home/laura/
# chown laura report.txt
# ls -l
-rw-r--r-- 1 laura office_malaga  0 2009-03-19 12:46 report.txt
```

Now the user `laura` is the owner of the file.
Both `chown` and `chgrp` can be used with the command line option -R to change the owner or the group recursively inside a directory.

4.8 CHANGING PRIVILEGES (chmod)

The `chmod` command allows you to change the permissions of one or more files. You can see the permissions using `ls -l`.

```
$ ls -l
-rw-r--r-- 1 pedro office_malaga  0 2009-03-19 15:38 hello_world.rb
$ chmod +x hello_world.rb
$ ls -l
-rwxr-xr-x 1 pedro office_malaga  0 2009-03-19 15:38 hello_world.rb
```

Execution permission has been added to the file hello_world.rb. Notice there are now three “`x's`“, one for the owner of the file, another one for all the users that belong to the same group that the file is in, and finally another one for the rest of the users. That means anybody can execute this program.

When none of the letters “u”, “g” or “o” are specified, by default, the permission is set for everyone (as in the previous example). You can also explicitly specify that you want to set the permissions for everyone (all) with the character “a”.

To illustrate this, the following table shows schematically, the parameters of the `chmod` command:

u	g	o	+ -	r	w	x
user, owner of the file	**g**roup, users which belong to the same group	**o**thers, rest of users	give permission remove permission	**r**ead	**w**rite	e**x**ecution

Let's remove the execution permission for the “rest of the users” (others) and let's give write permission to the users in the same group.

```
$ ls -l
-rwxr-xr-x 1 pedro office_malaga  0 2009-03-19 15:38 hello_world.rb
$ chmod o-x hello_world.rb
$ chmod g+w hello_world.rb
$ ls -l
-rwxrwxr-- 1 pedro office_malaga  0 2009-03-19 15:38 hello_world.rb
```

This method, that uses the characters `rwx` is called the symbolic method. We can also use another method called the numeric method.

4 2 1	**Total**
r w x	4 + 2 + 1 = **7**
r w -	4 + 2 + 0 = **6**
r - x	4 + 0 + 1 = **5**
r - -	4 + 0 + 0 = **4**
- w x	0 + 2 + 1 = **3**
- w -	0 + 2 + 0 = **2**
- - x	0 + 0 + 1 = **1**

So, this command line

```
$ chmod 755 hello_world.rb
```

is equivalent to these three lines

```
$ chmod u+rwx hello_world.rb
$ chmod g+rx-w hello_world.rb
$ chmod o+rx-w hello_world.rb
```

Either will give the following resulting permissions

```
$ ls -l
-rwxr-xr-x 1 pedro office_malaga  0 2009-03-19 15:38 hello_world.rb
```

You can change the permissions for directories in the same way as you do for files, although the meaning is a bit different. If a directory has read permission, that means you can see the content. If it has write permission, that means you can create files inside it, and finally, if it has execution permission, that means you can enter the directory using cd.

SUMMARY OF CHAPTER 4

- These are the commands used in this chapter:

Command	*Action*
ls -l	List files in the current directory in long format. You can see size, permissions, user and group to which the files belong, etc.
sudo	Enables you to execute commands as root.
su	Changes the user.
whoami	Shows the current user name.
groups	Shows the group or groups to which the current user belongs.
groupadd	Adds a new group.
groupdel	Deletes a group.
groupmod	Modifies the settings of a group.
adduser	Adds a new user.
userdel	Deletes a user.
usermod	Modifies the settings of a user.
passwd	Assigns or change the user password.
chown	Changes the owner of a file.
chgrp	Changes the group of a file.
chmod	Changes the permissions..

EXERCISES OF CHAPTER 4

The answers to the exercises can be found at the end.
Exercises are rated according to their level of difficulty:

🐧	Easy
🐧🐧	Medium
🐧🐧🐧	Dificult.

1. 🐧 Complete the following table:

`654`	
	`rwxrw-rw-`
	`rwxrwxrwx`
`520`	
`764`	
	`r--r-----`

In the following exercises, you should use the required Linux commands to perform the operations described.

2. 🐧 Create the groups `office1` and `office2`.

3. 🐧 Create the users `gearoid` and `paul`. These users must belong only to the group `office1`.

4. 🐧 Create the users `anna` and `emma`. These users must belong only to the group `office2`.

5. 🐧🐧 As user `gearoid`, create a file named `topsecret.txt` in his home directory. Only this user must have access to this file, both for reading and writing.

6. 🐧🐧 Create another file named `sales.txt`, also as user `gearoid`,. All users in the same group as `gearoid` must have access to this file, both for reading and writing. The permissions for owner and the rest of users must remain as default. Check that you can modify the file if you access it as user `paul`.

7. 🐧🐧 As user `anna`, create a file named `employees.txt`. Any user must have access to read its content and any user in the same group must have access to read or write to it.

8. 🐧🐧 Create the user `student` (if it is not yet created). Copy the file `employees.txt` to the home directory of user `student` . Change the owner and the group of this file to

`student`.

9. 🐧🐧 As user `paul`, copy a program from the directory `/usr/bin` to the home directory with a different name. For example, you can copy `xclock` with the new name `myclock`. Take a look at the permissions of this program. Check that you can execute it. Maybe you have to allow users to run programs in the graphic environment. To do that, type `xhost +` as administrador.

10. 🐧🐧 Change the permissions of `myclock` so that only the owner can execute it.

11. 🐧 Create the user `jim` asa member of the group `office2`. Create a new directory named `shared` inside his home directory.

12. 🐧🐧 Exit the graphic environment and enter again as user `jim`. Using OpenOffice.org Calc, create the files `contacts.ods`, `expenses_march.ods` and `salaries.ods`. Insert some data in each file and save everything in the directory `shared` created before.

13. 🐧🐧 Give read permission to all users for the directory `shared` and all the files inside it..

14. 🐧🐧 Remove write access from the file `contacts.ods` so that users in the same group can modify it, and the rest of user can't modify it.

15. 🐧🐧 Change the permissions in the file `expenses_march.ods` so that only the owner can modify it and all users in the same group can read it..

16. 🐧 Change the permissions in the file `salaries.ods` so that only the owner can access it, both for reading and writing.

17. 🐧 Supose a user has read access to a file, but this file is in a directory for which the user has no read access. Could this user read this file? Try it.

ANSWERS

1 BASIC CONCEPTS

1. An operating system is a program that allow the user to interact with the computer and its components, so (c) is correct.

2. A Linux distribution consists of the kernel of the OS, along with an installation program and a set of applications. Therefore, the correct answer is (a).

3. a) Sabayon b) Slackware.

4. There is a list of the most popular Linux distributions at http://distrowatch.com. At the time of writing this document, the information displayed is as follows:
   a) the most popular distros in last month are: 1 Ubuntu, 2 Debian, 3 Mint 4 openSUSE, 5 Fedora, 6 MEPIS, 7 Mandriva, 8 Arch, 9 PCLinuxOS, 10 CentOS.
   b) the most popular distros in last year are: 1 Ubuntu, 2 openSUSE, 3 Mint, 4 Fedora, 5 Debian, 6 PCLinuxOS, 7 Mandriva, 8 Sabayon, 9 Dreamlinux, 10 CentOS.

5. a) The first Linux version was programmed by Linus Torvalds, b) the goal was to improve MINIX, a UNIX-like operating system used in the university.

6. b) False. You can have any operating system on the local machine, the only thing you need is a telnet client to access the remote machine.

7. a) The first Linux version was written for AT 386/486 clones. b) Nowadays there are Linux versions for almost any hardware.

8. The Linux mascot is called Tux. Therefore, the correct answer is (b).

9. Larry Ewing.

10. You can take a look at this site: http://www.pendrivelinux.com. Some of the best known distros to use with memory sticks are Damn Small Linux (DSL), SLAX and Puppy Linux.

2 FILES AND DIRECTORIES (PART I)

1. In /etc directory.

2. b) Username and password.

3. ls

4. ls ..

5. cal may 1972 (type the correct month and year to see the calendar).

6. ls /bin

7. a) ls /usr/bin
   b) cd ..
      cd ..
      cd usr
      cd bin
      ls
   c) cd /usr/bin
      ls

8. ls -R /etc

9. ls -S /usr/bin/X11

10. ls -Slh /etc

11. ls -Sshr /bin

12. ls /

13. ls ../../..

14. mkdir expenses

15. A “normal” user can't create a directory inside /etc. you get the error “Access denied”. Only a user with admin privileges (root) can do. This prevent the users from damaging the system.

16. cat /etc/fstab

17. head /etc/bash.bashrc

18.
```
~$ mkdir multimedia
~$ cd multimedia/
~/multimedia$ mkdir music pictures video slides
~/multimedia$ cd pictures/
```

```
    ~/multimedia/pictures$ mkdir personal others

19. cd
    cd multimedia/music
    touch favourite_styles.txt

20. vi favourite_styles.txt
    i
    electronic
    ambient
    folk
    classic
    pop
    ESC:w
    :q

21. cat favourite_styles.txt

22. head -n3 favourite_styles.txt

23. tail -n1 favourite_styles.txt

24. tail -n+2 favourite_styles.txt
```

2 FILES AND DIRECTORIES (PART II)

1.
   ```
   ls *.jpg
   ```

2.
   ```
   ls /usr/bin/j*
   ```

3.
   ```
   ls /usr/bin/k?a*
   ```

4.
   ```
   ls /bin/*n
   ```

5.
   ```
   ls -R /etc
   ```

6.
   ```
   ~$ mkdir test
   ~$ cp /bin/gzip test
   ~$ cd test/
   ~/test$ cp gzip gzip2
   ```

7.
   ```
   ~/test$ cd ..
   ~$ mv test test2
   ~$ mkdir test3
   ~$ mv test2/* test3/
   ~$ rmdir test2
   ```

8. Yes, you can do it using backslash to escape special characters:
   ```
   touch \*\?Hola\ caracola\?\*
   ```
   It is not recommended to use strange characters that can be confused with wildcards. Nor is it recommended to use white spaces. Instead, you can use the underscore character “_”.

9.
   ```
   ~$ mkdir multimedia_tests
   ~$ cp -R multimedia/* multimedia_tests/
   ~$ cd multimedia /video/
   ~/multimedia/video$ touch films.txt actors.txt
   ~/multimedia/video$ vi films.txt
   ~/multimedia/video$ cd
   ~$ cd multimedia_tests/video/
   ~/multimedia_tests/video$ vi films.txt
   ~/multimedia_tests/video$ cd
   ~$ cp -Ru multimedia/* multimedia_tests/
   ```

10.
    ```
    rm -Ri multimedia/pictures/others/
    ```

11.
    ```
    ~$ cd multimedia/video/
    ~/multimedia/video$ mv films.txt ../my_films.txt
    ```

4 GROUPS, USERS AND PERMISSIONS

1.

654	**rw-r-xr--**
766	rwxrw-rw-
777	rwxrwxrwx
520	**r-x-w----**
764	**rwxrw-r--**
440	r--r-----

2.
```
# groupadd office1
# groupadd office2
```

3.
```
# adduser gearoid --ingroup office1
# adduser paul --ingroup office1
```

4.
```
# adduser anna --ingroup office2
# adduser emma --ingroup office2
```

5.
```
$ su gearoid
$ cd
$ touch top_secret.txt
$ chmod 600 top_secret.txt
```

6.
```
$ touch sales.txt
$ chmod g+rw sales.txt
```
   We can check that user paul can modify this file because he is a member of the same group:
```
$ exit
$ su paul
$ vi /home/gearoid/sales.txt
```

7.
```
$ exit
$ su anna
$ cd
$ touch employees.txt
$ chmod 664 employees.txt
```

8.
```
$ exit
$ sudo cp /home/anna/employees.txt  /home/student/
$ sudo chown student /home/student/employees.txt
$ sudo chgrp student /home/student/employees.txt
```

9.
```
$ sudo xhost +
$ su paul
$ cd
$ cp /usr/bin/xclock myclock
$ ls -l
-rwxr-xr-x 1 paul office1 32568 2009-03-23 11:18 myclock
$ ./myclock
```

10.
    ```
    $ chmod go-x myclock
    ```

11.
    ```
    $ exit
    $ sudo adduser jim --ingroup office2
    $ su jim
    $ cd
    $ mkdir shared
    ```

12. You can run Calc through Applications → Office → OpenOffice.org Cal Spreadsheet.

13.
    ```
    chmod -R a+r shared
    ```

14.
    ```
    cd shared
    chmod g+w contacts.ods
    chmod o-w contacts.ods (this is redundant)
    ```

15.
    ```
    chmod 640 expenses_march.ods
    ```

16.
    ```
    chmod 600 salaries.ods
    ```

17. No. A user cannot see inside a directory if he does not have read permission for it, even if the permissions in the files inside the directory are set to 777.

APPENDIX A. GNU FREE DOCUMENTATION LICENSE

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.

- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  YEAR  YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled "GNU
Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with … Texts." line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.