Computer Systems

UD 10. LINUX - PART 1

Computer systems
CFGS DAW

Alfredo Oltra
alfredo.oltra@ceedcv.es
2022/2023

Versión:220725.1712

Licencia

Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

 Importante

 Atención

 Interesante

INDEX

1. What is Linux?....4
1.1 GNU/Linux....4
1.2 What is a distribution?....5
1.3 Where do we find Linux?....5
2. Software licenses....5
3. Installation....7
3.1 Download....7
3.2 Installation....7
3.3 The hard disk....8
3.4 The partitions....8
3.5 The boot sector....12
4. Applications....13
4.1 Basic applications....13
4.1.1 File Explorer....13
4.1.2 Text editor....13
4.1.3 The terminal....13
4.2 How to install programs....14
5. Additional material....15
6. Bibliography....15

UD010. LINUX - PART 1

1. WHAT IS LINUX?

The simplest way to define Linux is that it is a Unix-style operating system. At this point of the school year we already know that it is an operating system, but what is Unix?

Although not the first operating system, Unix is, undoubtedly, the first great operating system. Its most important features were that it is portable (can run on different computer systems) multitasking and multi-user. In addition, architecturally speaking, it was created based on concepts such as simplicity and modularity so that the code was easily maintained and extended by other programmers.

Over the years, the creator company (the Bell Labs[1]) was licensing the product to other companies that, in order to adapt it to more specific environments, were making modifications and creating different versions. Hence, products like Xenix (Microsoft), HP-UX (HP), IRIX (Silicon Graphics), SCO (Novell), AIX (IBM) … were born.

> All systems in the UNIX family are often called *IX

The problem with all of them is that they are proprietary versions so that the code is not available for study (apart from the large amount of money that a license cost). That is why, with a purely educational objective, in the late 80's, a professor at the University of Amsterdam decided to create Minix, an OS based on the UNIX philosophy but rewritten from scratch and open source. Due to its educational nature the author decided not to allow modifications which would very likely complicate the code much more.

It is at this time that a Finnish computer science student decides, based on Minix, to create a free clone that works on PC systems. This student was Linus Torvalds[2] and he called him operating system Linux.

1.1 GNU/Linux

In the previous section we commented that Linux was an operating system, but that is not exactly true. Every operating system consists of a kernel or core and a set of applications that help make possible the function of the operation system. In fact a possible classification of the different programs that accompany the kernel in an operating system could be: the shell or the terminal (which allows interaction with the user text mode), services or daemons (which are programs that run in background), a graphical server (which allows you to draw elements on screen) or a desktop (which takes advantage of the functions of the graphical server to provide graphical access to the user).

1 Founded by the inventor of the phone, Graham Bell
2 http://www.comunidadhosting.com/t/primer-mensaje-de-linus-torvals-y-comienzo-de-linux.8904/

Linux is simply the kernel of the system, kernel that to form the OS is accompanied by many GNU[3] utilities. That operating system is not called Linux, it's called GNU/Linux

Every day, everyone simplifies and calls Linux the operating system (wrongly).

1.2 What is a distribution?

The kernel and basic utilities make up the core operating system, but we know that today an operating system is gone with by a lot of software not necessary for the computer itself (drawing programs, text editors, etc.). The fact that the kernel and the basic utilities are freely distributed allows anyone to take these elements and accompany them with other utilities (whether basic or not) according to the way they likes it or their needs. This is how distributions of Linux or, more commonly, *distros* arise.

There are hundreds of them in the market, but to mention some of the best known: Ubuntu (and its different "flavors" like LUbuntu, KUbuntu, Ubuntu Server …), Open Suse, Arch, Fedora, Debian, Red Hat, Mint, Lliurex.

We will use LUbuntu throughout this course, a distribution based on Ubuntu that aims to be lightweight.

1.3 Where do we find Linux?

One of the most widespread ideas is that Unix is an operating system that is only used in academic environments or of a high technical level, and nothing is further from reality. The systems based on free versions of Unix are implanted in many computer systems. Although possibly in desktop environments Microsoft systems are still the ones that dominate most of the market, in mobile devices as in servers the reality is quite different.

For example, the two mobile operating systems par excellence (Android and iOS), are systems derived from Linux or freeBSD[4] (as well as systems such as MacOS).

2. SOFTWARE LICENSES

Several times throughout the previous sections we have commented that one of the fundamental features of Linux is the fact that it is free. But what does it mean to be free?

Because of the use of the same word in English, many people think that being free (at liberty) implies being free (not charge). The price of the software has nothing to do with the type of license to which you submit.

3 GNU is a pre-Linux project whose goal is to create a free operating system. The project has developed a lot of the utilities that go with the kernel, but to this day, they have not achieved to have a stable enough kernel.
4 Freebsd is a free version of BSD, one of the variations of Unix that was licensed to the University of Berckley

> The software license is a contract that defines all the rules that govern the use of a particular program. The contract is made between the owner of the program and the user of the application.

Its clauses determine issues such as the term of assignment of rights, the geographical scope of validity of the contract, limits of responsibility for failures, number of allowable copies, possibility or not of transferring the software to third parties, etc.

There are several types of licenses, but in general all could be included in one of these types:

- Proprietary license. Use of software on one or more machines for a certain cost. It is not usually included the program code, only the executable file.
- Shareware license. Use of software with limited functionalities for a time (or even of a definitive way) and need of payment to extend them.
- Freeware license. Unlimited use and copying at zero cost.
- Free software. It allows the use, copy, modification and free distribution with access to the source code.

Within the latter we can find two types:

- Permissive licenses: those that give the user total freedom so that with the modified code he can do what he wants. That is, although the original is free code, the modification can be private. A license of this type is BSD.
- Non-permissive licenses: those that require that the copy, modification and subsequent distribution is carried out under the same conditions as the original. That is, if the original is free code, the copy must remain so. A license of this type is the GPL, which is used by GNU products.

3. INSTALLATION

3.1 Download

The first step is to get and to burn and ISO of *LUbuntu*. You can download it from:

http://lubuntu.me/

After the download, the next step is to burn the ISO image to a CD/DVD (with a burning program like *Nero*, *Burn*, *Toast*, *ISO Burner*, etc.) or in a USB stick (with *UltraISO*, *Rufus*...)

3.2 Installation

Restart the computer with the burned DVD with *LUbuntu* in the DVD reader or if, you use a USB, with the stick connected. If we let the system start up, it will end up starting a version of *LUbuntu* in graphic mode with options to install it in a hard disk.

> To boot from DVD or USB it is necessary to have this option activated and/or modify the system boot order. These options can be configured from SETUP

We choose the *Install LUbuntu* Option

Figure 1. LUbuntu install menu

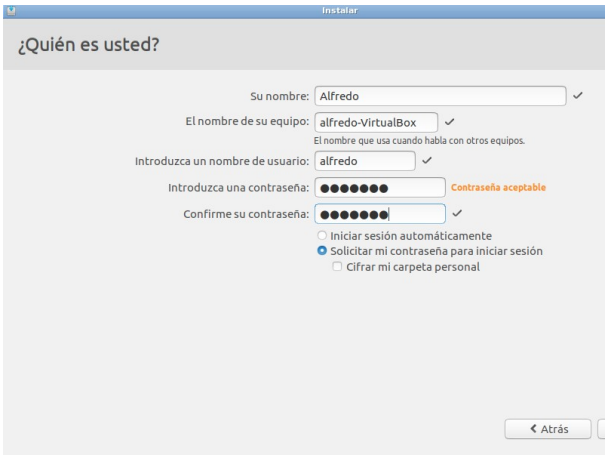

Figure 2. User data

Today the installation of Linux is very simple. In fact in the vast majority of cases accepting the default options and assigning a user (figure 2), the system is installed correctly.

But in order to make the installation more flexible and better adapted to the needs of the user, it is worthwhile to go into detail in the *installation types* window. From it, we will be able to allocate the disk in which we are going to make the installation as well as to realize an optimal distribution of the partitions to create.

In this window we have to choose the last item: *more options* (*figure 3*). This option open the window showed in the figure 4.

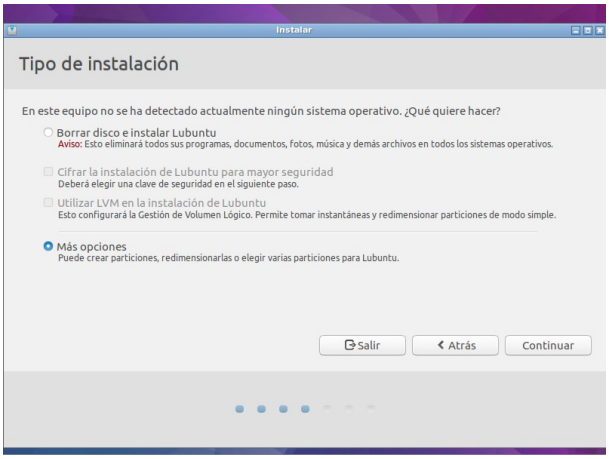

Figure 3. Type of installation

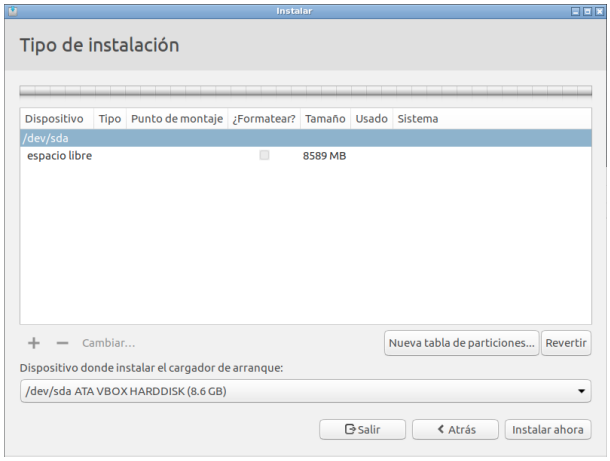

Figure 4. Selecting hard disk

3.3 The hard disk

The first step is to choose on which disk we want to perform the installation.

In Linux the devices are simulated using files that are in the *dev* directory. There are many types[5], but in terms of connected elements in the PATA or SATA connectors, the way to call them is: type + order + partition. The types are *hd* for PATA devices and *sd* for SATA devices. The order refers to the port number element in which they are connected. To do this, letters are used starting by the *a*. The partition are numbered with numbers, starting for the number 1. For instance, *sdb1* refers the first partition in a device connected in the second SATA port.

In the example you can see (figure 3) that the computer has only a SATA (*sda*) hard disk connected, which is therefore the only one that can be used to perform the installation. In case there were more (*sdb*, *sdc*, *sde* ...), these would appear in the list so that we could choose in which one of them we want to install.

3.4 The partitions

Usually we have associated the idea that a program is installed on a hard disk. In fact in the previous section we have hinted that it is the first step to perform the installation of an OS. However, an operating system is special software and can be spread over different partitions of a hard disk or even in different hard disks.

In fact this possibility is an obligation for any user with average knowledge. Apart from the security provided by the fact to separate contents (limited if the installation is performed on different partitions of a single hard disk, since if the hard disk fails all partitions fail), has the great advantage that the re-installation of the system in case of error (for example by restoring an image) is very fast and not dramatic.

How can we distribute the partitions? What can we include in each of them?

Linux always creates by default at least 2 partitions: one for the system and another, called swap, for use the virtual memory. At the moment that we decide to perform the custom installation we must remember that we must create a swap partition. Its recommended size is double of the

5 We will study several of them in the next lessons

memory RAM of the machine, although thinking about possible extensions of the same and taking into account that the capacity of hard drives today is great, a good option is assign four times more.

To see how we can distribute the rest of the system it is necessary to understand how the Linux file system works. Unlike Microsoft systems, in which each partition is assigned a drive (each of which is called using a letter of alphabet, a: b: c: d: e: ...) in Unix systems the user has a single unit called `/` (*slash* or root directory) from which to connect (to mount) in different directories the information of all the partitions of hard disks existing in the computer,

Obviously all this information is organized in such a way as to make it simple its location.

For example:

`/boot`: this folder has files for the boot management.

`/root`: It contains system administrator (superuser) programs.

`/var`: It contains variable files, such as registers, databases, mail queues...

`/tmp`: temporary files.

`/usr`: programs and system data that can be shared by multiple users.

`/home`: user's folders.

`/dev`: files that encapsulate different physical devices connected to the system. For example `/dev/lp` is a printer.

`/opt`: third-party applications (similar to Windows Programs files)

`/lib`: libraries common to all applications

`/etc`: applications configuration files.

`/mnt`: in this folder are located directories by each of the rest of the partitions of all the hard disks in the computer. For example, if I had a hard drive with a partition with W*indows*, its location would be `/mnt/windows` (the name of the folder could be any)

`/media`: similar to the previous one but with devices that are mounted live (*usb* or *SD card*)

The distribution of partitions may be made following this structure. So for example if to one partition is assigned `/`, in that partition will install the entire system (which is what is done in a default installation).

A more consistent distribution is to create a partition for the whole system and another where the user's folders are stored (`/home`). In this way the system is separated from the data facilitating the re-installation. Another interesting option is, in addition to these two, create a partition where to store `/var`, thinking about separating other data types such as databases.

> The indicated folders are just a few of them. And not all of them allow to be isolated in a partition. For example `/mnt` and `/media` does not make any sense that they are in a separate partition since they are already in themselves a site where other partitions are going to be incorporated.

Each of these partitions can be on the hard drive that we want. In our case we will use a single hard disk and will create 3 partitions: `/`, `/home` and, of course, `swap`.

To do this, simply select the empty space on the hard drive where you want to create the partition and press the + button. This opens a window where you can assign a size (in MB), the type, file system type (usually *ext4*) and the mount point you want to include in that partition.

> The act of including a partition in our general file system is called *mount*. So for example, when we connect a USB, the system what it does is that it automatically mounts that unit in the `/media` folder.

Although the type of partition is a topic that today is not as important as a few years ago, it is at least interesting to know that there are two types of partitions: the primary and the extended. A hard disk can only contain 4 primary partitions. Formerly it was more than enough, since the size of the hard drives was not enough for much more, but with the increasing capacity there arose the need to create more partitions.

Faced with the impossibility of increasing the number of primaries was created the concept of partition extended in such a way that a hard disk can create a maximum 3 primary and one extended. The great advantage of the extended one is that inside it could exist as many as partitions as desired. These partitions included within the extended partition are called logical partitions. For practical purposes the type of partition does not have much importance although it is usually advisable that the main partitions and the boot system are in primary partitions.

In the rest of the figures you can see how the new partitions are distributed on our hard disk.

> Take note of the file system to swap partition.

> Obviously we have to note that when we accept the distribution, all the information that has previously stored on that hard disk will be destroyed.

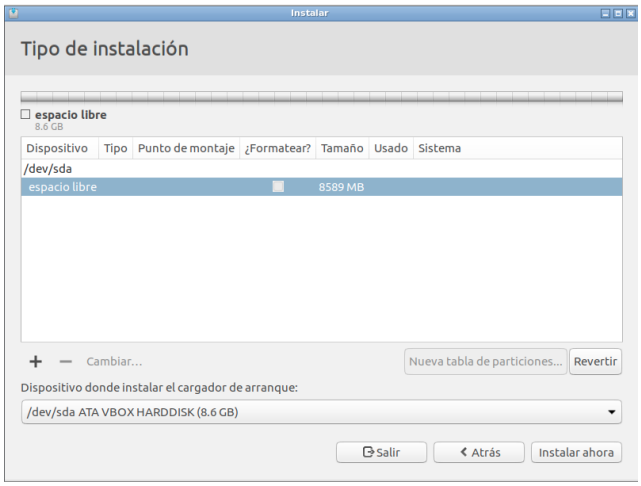

Figure 5. Selecting empty space

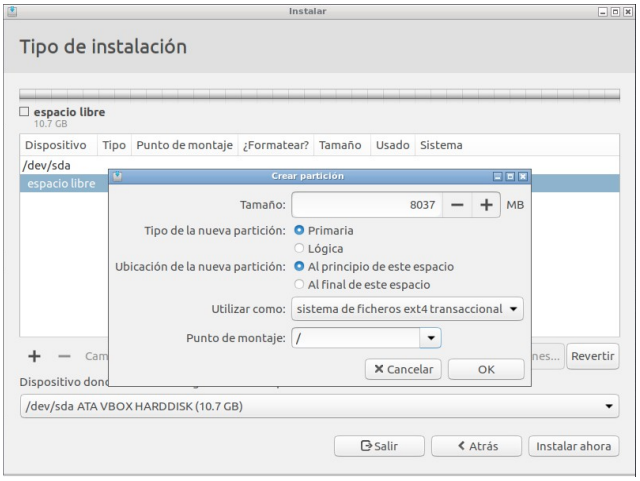

Figure 6. Config / partition

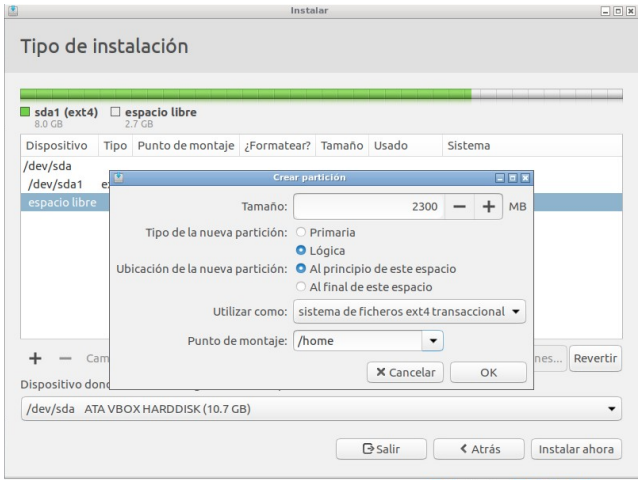

Figure 7. Config `/home` partition

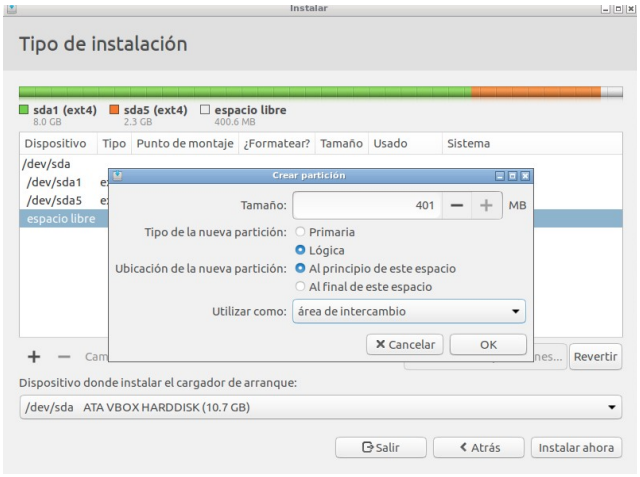

Figure 8. Config swap partition

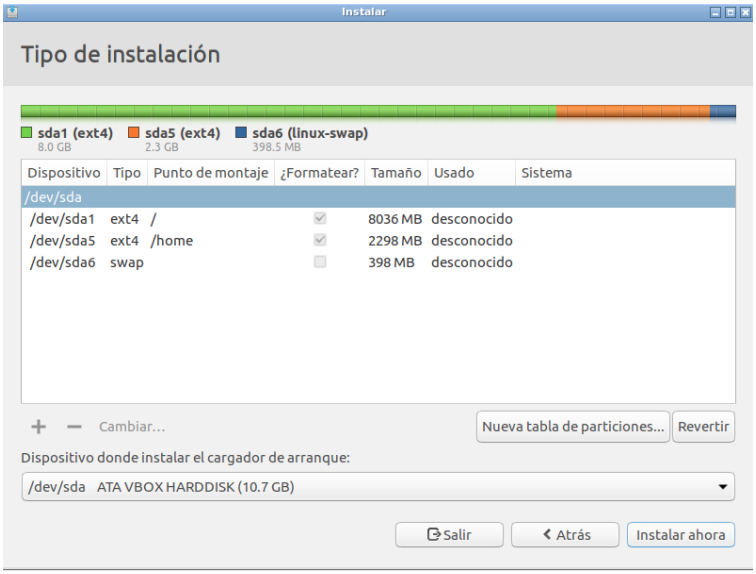

Figure 9. Final config

3.5 The boot sector

The last step is to select on which hard drive we want to install the bootloader. As we saw in previous lessons, in the BIOS there is configured an order in which to search, in the different storage systems, a sector (MBR) that allows to run the load of an operating system. The BIOS program in charge of that task, looks for that list of devices until finding one that allows the boot. That is why each operating system has to install and configure that sector, regardless of the installation of the operating system itself. This is easily achieved when there is only one operating system on the computer, but what happens when more than one is going to be installed?

With the first system to install there will be no problem, but when we install the rest, each one will install the boot sector overwriting the previous one. In this way, even if the operating systems are installed, you will only be able to boot the one that was last installed. Fortunately, Linux

distributions are usually very respectful in that aspect, so if when they install their boot sector, they detect the existence of another, do not overwrite it, but modify a small program (usually called grub) to display a menu that allows to select the OS from that which you want to boot.

> However, if the system installed later is one of the Microsoft family, it only takes into account systems that belong to the same family. So if you want to have multiple operating systems on the same computer, install those from the Microsoft family first.

> If you have problems with the boot sector remember that operating system is there, the only problem is accessing it. For that there are several utilities to make a backup, restore or repair it in case of problems

4. APPLICATIONS

4.1 Basic applications

Although each distribution includes the applications they consider to get their goals, there are three that are very common and help you in almost any desktop distribution.

4.1.1 File Explorer

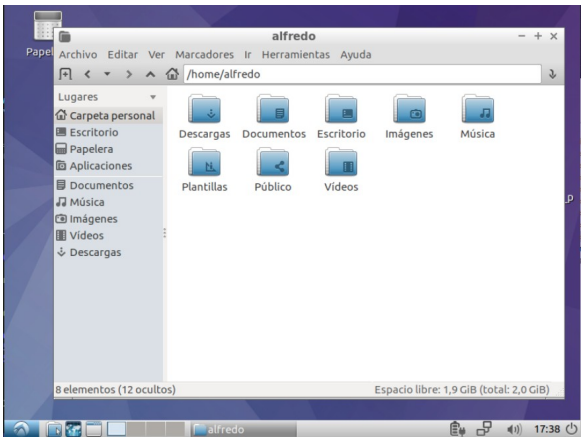

Figure 10. File explorer

It comes to be the *Windows File Explorer* or the *MacOs Finder*. You can run it from Accessories or search the system search bar, but the most common way is to run it from the icon that exists in the toolbar.

The great advantage of Linux is its heterogeneity, which does not exist to single file explorer. Depending on the desktop you installed the options may vary. One of the best known is *Nautilus*. In *LUbuntu* we can use *PCManFM*

4.1.2 Text editor

The Linux notepad. As always the options are multiple, but the most used and comes standard on most of the desktops is *gedit*. In *LUbuntu* we can use *leafpad.* It can be accessed from *Accessories.*

4.1.3 The terminal

Perhaps the most hated app by the basic users, but the best friend of those who want to get the most out of the system. Through it we can iterate with the operating system in text mode. We will develop all our work from it. We can access it from *System Tools/LXTerminal.*

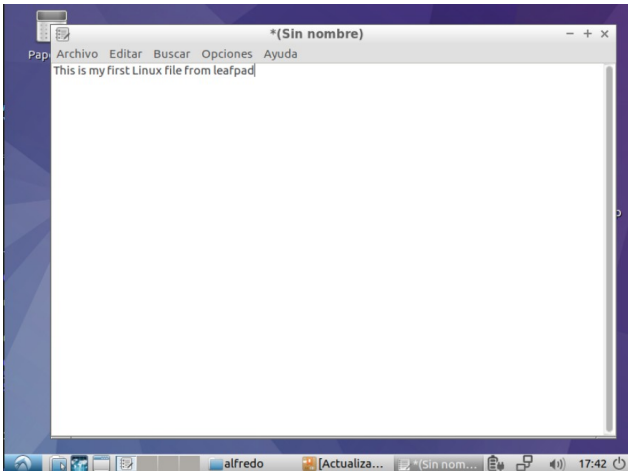

Figure 11. Edit text

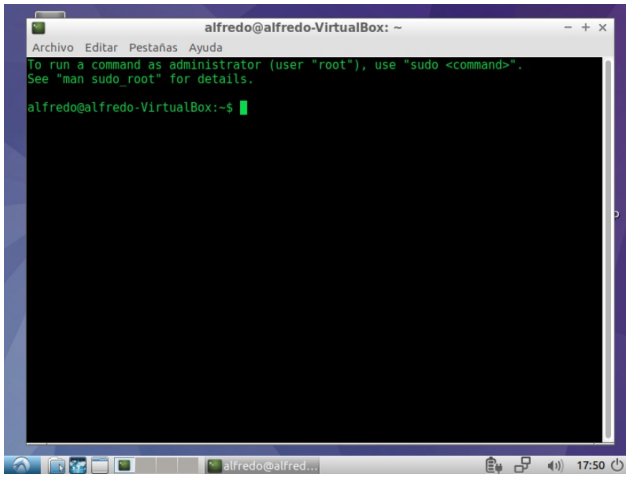

Figure 12. Terminal

4.2 How to install programs

There are three fundamental ways to install programs on Linux:

- From a package
- From a compressed file[6]
- From the source code

Obviously, this last option is the most complex and requires the knowledge necessary to compile the application, knowledge that is beyond the scope of this module.

The simplest way is to use packages, something similar to Windows *msi*. These files include all the information for the installation and the configuration of the application, notifying to the system of the dependencies mandatory to be able to execute it correctly.

> The problem is that there is no single package format. In the market there are two formats, *".deb"* (used by distributions like *Ubuntu* or *Debian*) and *".rpm"* (used by distributions like *OpenSuse* or *Fedora*). That makes the developers must generate the two types of packages to distribute their applications.

Actually the installation of these packages is very simple, they are simply downloaded, clicked on them and the installation program is automatically launched.

The problem is that *Linux* systems are very open and very dynamic systems. The versions of the applications happen really quickly and the places from which to download are very varied. This is why it is convenient to use the names of application managers, an idea existing in *Linux* for many years and that comes as something like today's application stores (*App Store, Google Play*, etc.).

6 We will study this method in next lessons

Of course, each type of package is associated with its ow manager (working in terminal mode). Today the most common format is the store (in graphical mode), but possibly the most versatile is the classic (the ancestor of the store).

For example, in distributions with "*.deb"* packages, the manager is a console command called "*apt-get". I*ts classic desktop version is *Synaptic*, and the storage mode is called "*Application Center"*.

In case of knowing the name of the package the simplest method is the first one. For example, if we want to install the VLC player, the easiest way is to open the terminal and write:

```
sudo apt-get install vlc
```

Where:

- `sudo`: allows us to execute applications in superuser mode. Obviously the installation of programs is not something that can do any user, so to be able to do this it is necessary to ask for superuser credentials.
- `apt-get`: name of the package manager application.
- `install`: option of the manager program that allows the installation. Obviously there are others for removal, updating, etc.
- `vlc`: package name

If we run, the system prompts us for the password and proceeds to the installation.

> Linux is case sensitive, so it is not the same to write in uppercase than in lower case.

If we do not know the exact name of the package, it is better to use *synaptic*. We can access from *system tools* menu. The application has a search engine to locate the package that interests us and then select it. The dependencies necessary for installation will be automatically selected.

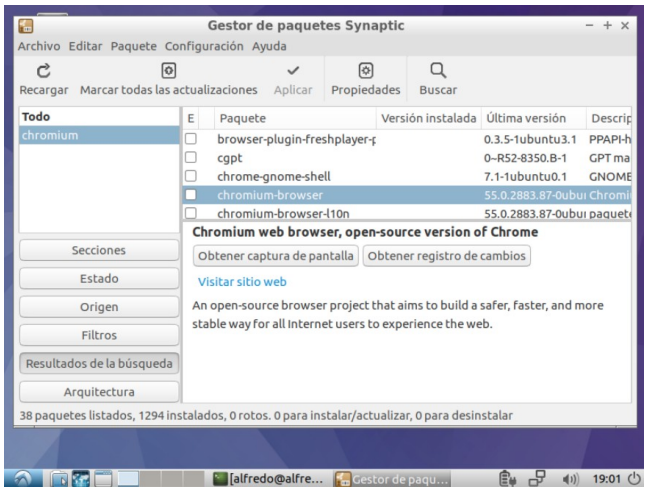

Figure 13. Synaptic

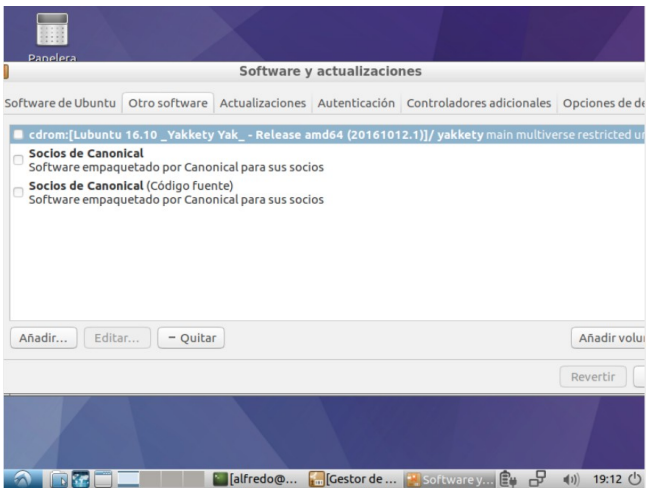

Figure 14. Repositories

These managers are configurable to be able to support any data source. In general it is configured against the servers of the distribution that is responsible for keeping them as up to date as possible.

Even so, they do not contain all the programs nor the last versions, so in many cases it is usually convenient to add new sources that allow be as up to date as possible.

You can manage those fonts from *Synaptic* in the *Settings/Repositories* menu in the other fonts tab (figure 14).

5. ADDITIONAL MATERIAL

[1] Glossary.

[2] Exercises

6. BIBLIOGRAPHY

[1] Organización de los directorios en Linux
http://www.linux-es.org/node/112