Pre-workshop material

DBTech's Virtual Workshop on Distributed, Replicated, Embedded and Mobile Databases (Deliverable #28).

Antonio César Gómez Lora

Dpto. Lenguajes y Ciencias de la Computación

Universidad de Málaga

Introduction

In this document you can find useful information required for the DBTech's Virtual Workshop on Distributed, Embedded and Mobile Databases.

This document is divided into four different chapters. The first one enumerates all the concepts you must be familiarized with. These concepts concern topics among different areas, such as databases, networking, and virtualization.

The second chapter introduces virtualization and how this powerful tool can be used to test distributed systems.

The third chapter introduces the Operative System selected. And the fourth one describes the database software we are going to use.

Pre-requirements

This workshop is not oriented to general public. As the title describes this virtual workshop is focused on distributed, replicated, embedded and mobile databases, so you must known what a relational database is. You must also have some experience on SQL language and it is recommend that you have some experience on database, or at least have a minimum theoretical notion of, database administration.

You must have some experience with some relational database management system, especially with server based solutions, such as Oracle Database, Microsoft SQL Server, IBM DB2, Sybase Adaptive Server, PostgreSQL, MySQL, etc.

Talking about distributed environments also means talking about network environments. So some basic skills on network administration are required. This means that we assume that you known how to configure a network interface, involving the following concepts: MAC address IPv4 address, routing and gateway.

This workshop builds a virtual network using different virtual machines. But you can ignore all the virtualization part and issues in the case that you have a real network with real computers and you want to use them.

We also assume that you known basic topics on virtualization or you have seen a virtualized system before (VirtualBox, VMWare, Parallels Desktop, ...)

Virtualization

Virtualization can be defined as the capacity to abstract the functionality and behaviour or simply simulate computer devices, components and/or resources. When the abstraction includes all the components of a computer, implementing a complete environment where we can install and execute software, this virtual environment can be viewed as "virtual computer", called the guest, different from the real one that executes the virtualizer, called the host. Generally the virtualizer allows us to simulate and execute different guests concurrently.

The software than build this abstraction layer is called hypervisor, virtual machine monitor or virtualizer. There not exists a unique way to implement virtualization, so depending on what and how you abstract or simulates, and how it affects the underlying system (the host) you can find different kinds of virtualization:

- **Full or native virtualization**: the simulation of the virtual hardware does not require modification in the host operating system. Some examples: VirtualBox, VMWare, Parallels Desktop, etc.
- **Para-virtualization**: the simulation of the virtual hardware requires modifications in the host operating system. For example Xen.
- **O.S. virtualization**: Virtualization only affects some resources or components (it does not build a complete virtualized machine), for example the filesystem and network interfaces. So host and guest operating system share some hardware and even software resources, for example CPU, memory and the O.S. kernel. Parallels Virtuozzo and Open VZ implements this kind of virtualization.

When the virtualized mix full and para-virtualization it is called partial virtualization, and when the host's hardware support virtualization and the virtualizer uses this feature it is called hardware-assisted virtualization.

Virtualization allows you to build virtual computers. Virtual computers have some negative and positive aspects. Basically only two of the negative aspects are focused on virtualization: performance and timing. Accessing virtual hardware is slower than accessing a real device, so it is not a good idea to virtualize a system that requires an intensive use of a hardware device. For example, virtual machines are no good for gaming (they usually require an intensive use of the display adapter) and for big databases (they require an intensive use of the storage devices). Virtual machines also does not have a stronger control over their CPUs and the system clock so they are not good solutions for system where time is an impact factor, for example, complex real-time systems.

Among the advantages of virtualization in enterprise environments: it increases the real servers (hosts) use-index, decreases the use/cost ratio and decreases the energy consumption and heat generation (reducing the cooling costs). Also as virtualization abstract the real

hardware layer it allows hardware changes and even improve dramatically the hardware migration costs.

Among the advantages of virtualization in personal, teaching, researching and development environments: it allows us to use different O.S., to have portable virtual machines, to make specialized machines and distribute them, to test potential dangerous code or access risky areas (i.e. test virus infection and cleaning, run suspicious programs), to test performance when hardware changes (you can easily test a complete system with different hardware configurations, for example changing the assigned amount of RAM and CPU), to test network environments and distributed systems.

There exist many commercial and free products for virtualization. We have selected Oracle's VirtualBox because although it has an Open Source version (VirtualBox-OSE), the commercial version is also free of charge for personal use, evaluation or educational use (being free for teachers and students in schools, colleges and universities). VirtualBox and VirtualBox-OSE are easy to use full virtualizers with a desktop-oriented graphical interface, and it does not limit the amount of hardware resource you can assign to a guest system. Some other good candidates are: Microsoft Virtual-PC if you work only with Windows based systems; VMWare Workstation if you have the license (it is not free); VMWare server if you do not want to assign more than 2 CPUs to a guest system.

VirtualBox

This manual is based on the last version available for VirtualBox (3.2.10). You can download virtual from its web site. There exist different versions based on the host O.S., so you can find versions for Windows, OS X, Linux and Solaris as you can see in Fig. 1.

Fig. 1. VirtualBox download page.

VirtualBox initial screen allows us to manage our guest systems.

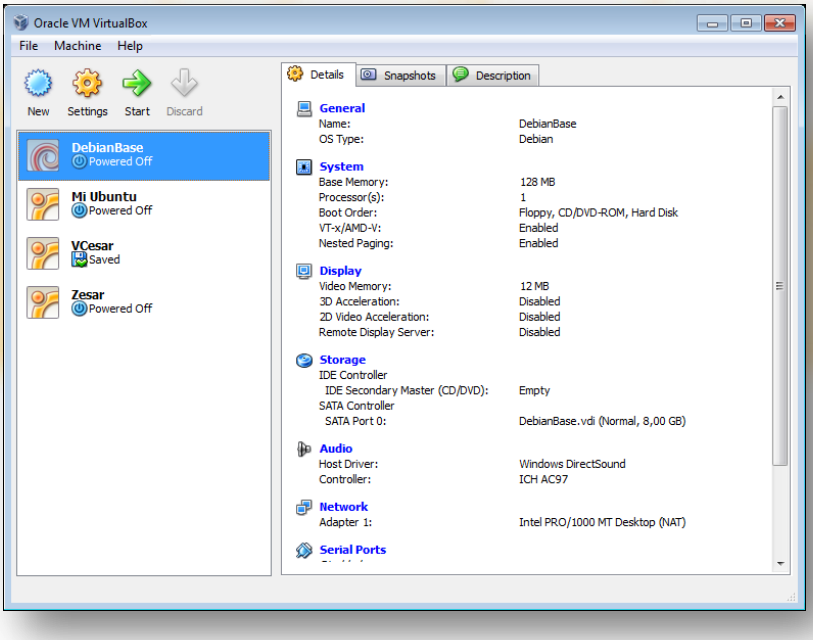

Fig. 2. VirtualBox GUI.

As you can see in the Fig. 2, this example VirtualBox has four virtual machines created (DebianBase, Mi Ubuntu, VCesar and Zesar). None of them is running, three are powered off and one (VCesar) is not running but it has its execution state saved.

Using the button New or the menu entry Machine->New (Fig. 3) we start the new virtual machine wizard.

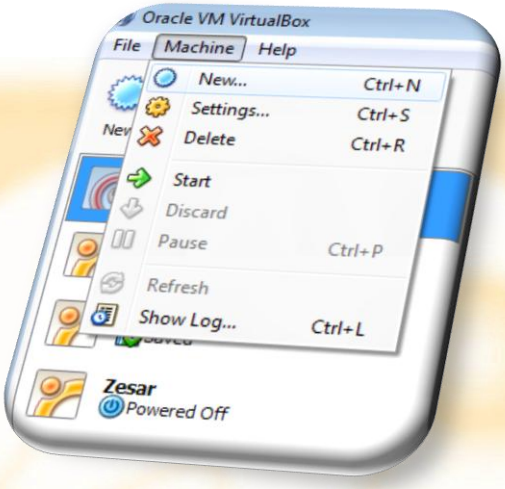

Fig. 3. New machine menu entry.

The wizard welcome screen can be seen in Fig. 4.

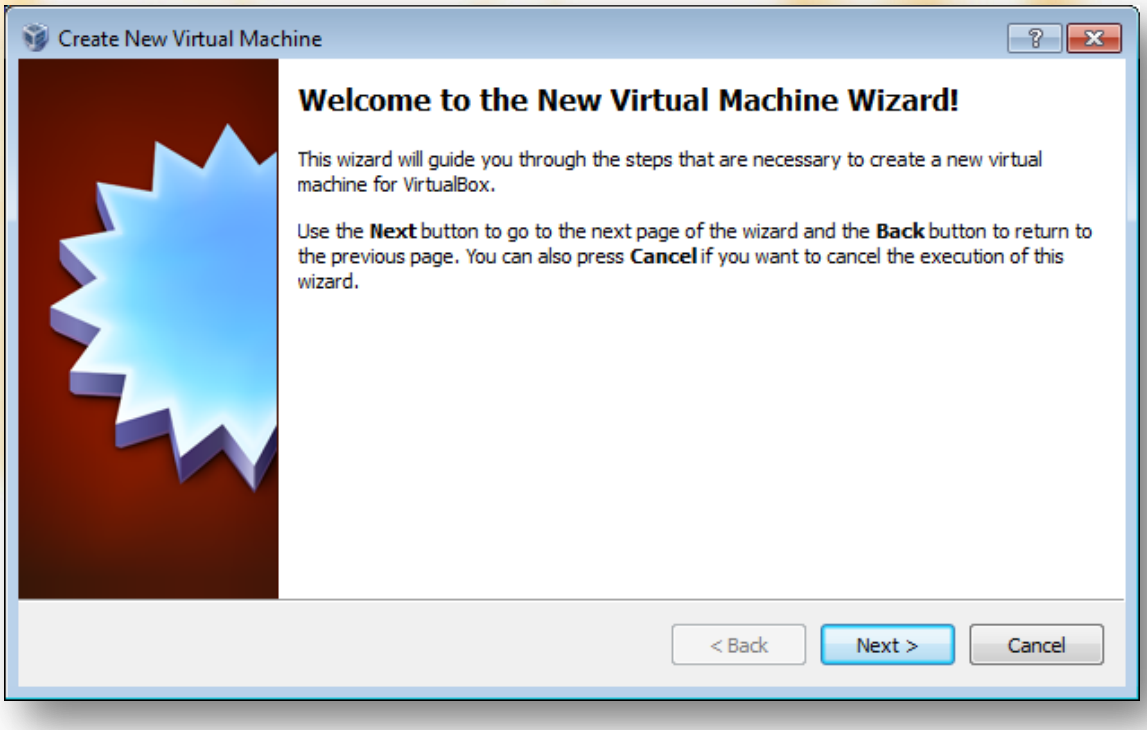

Fig. 4. New Virtual Machine welcome window.

We must introduce a name for the new machine and specify the guest operating system and version we plan to install (Fig. 5).

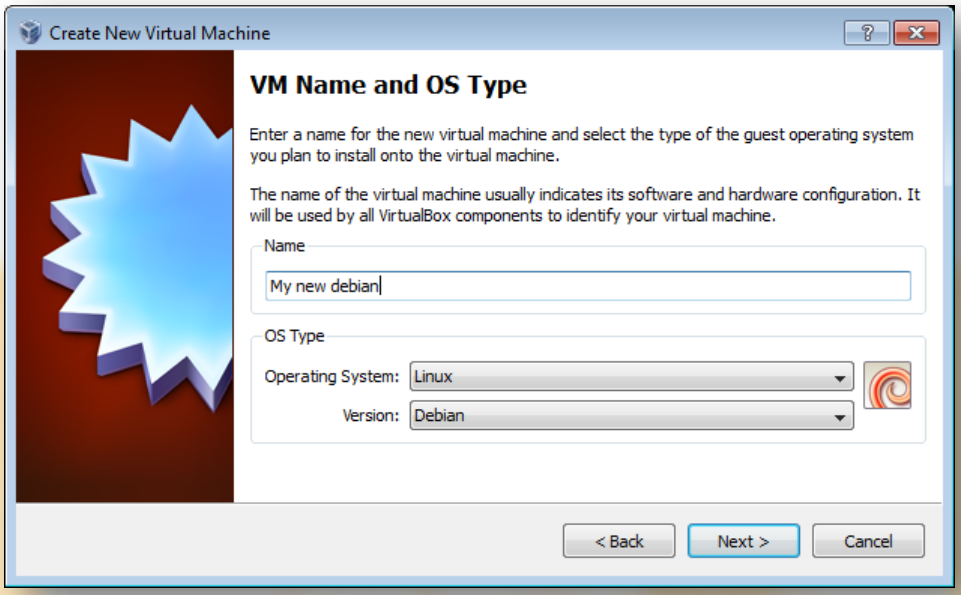

Fig. 5. VM Name and OS Type.

Next we must specify the amount of our RAM memory that we want to assign to the new virtual machine. Based on selected OS Type and version VirtualBox recommends a base memory size, but we can increase or decrease this value. The maximum size corresponds with our RAM size. In Fig. 6 although VirtualBox recommends 384 MB for Debian Linux, we reduce the size to 128 MB, because we are going to install none X window manager.

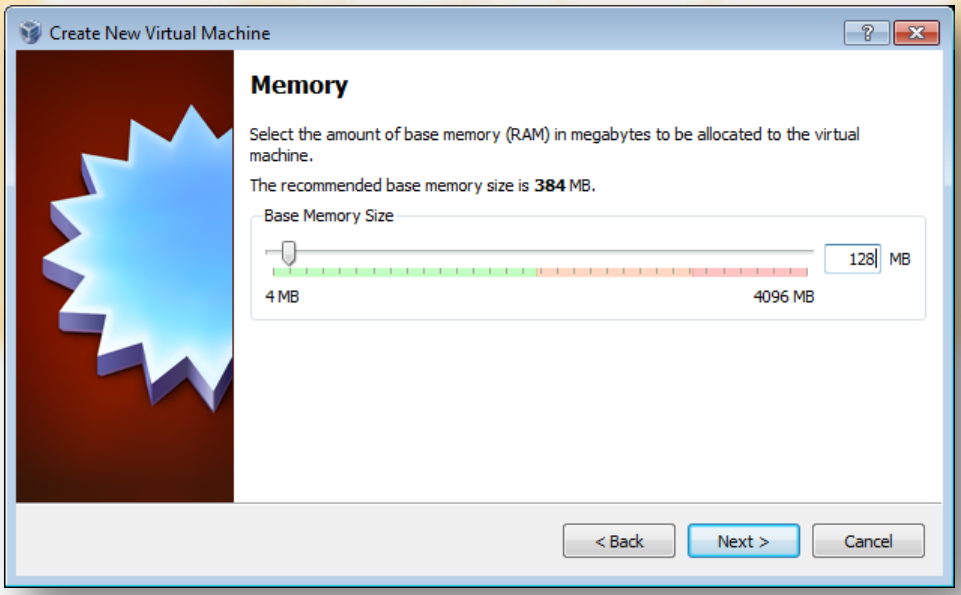

Fig. 6. Memory.

Then we must specify if we want to create a new virtual hard disk for this machine or use an existing one (Fig. 7). Let's select a new one.

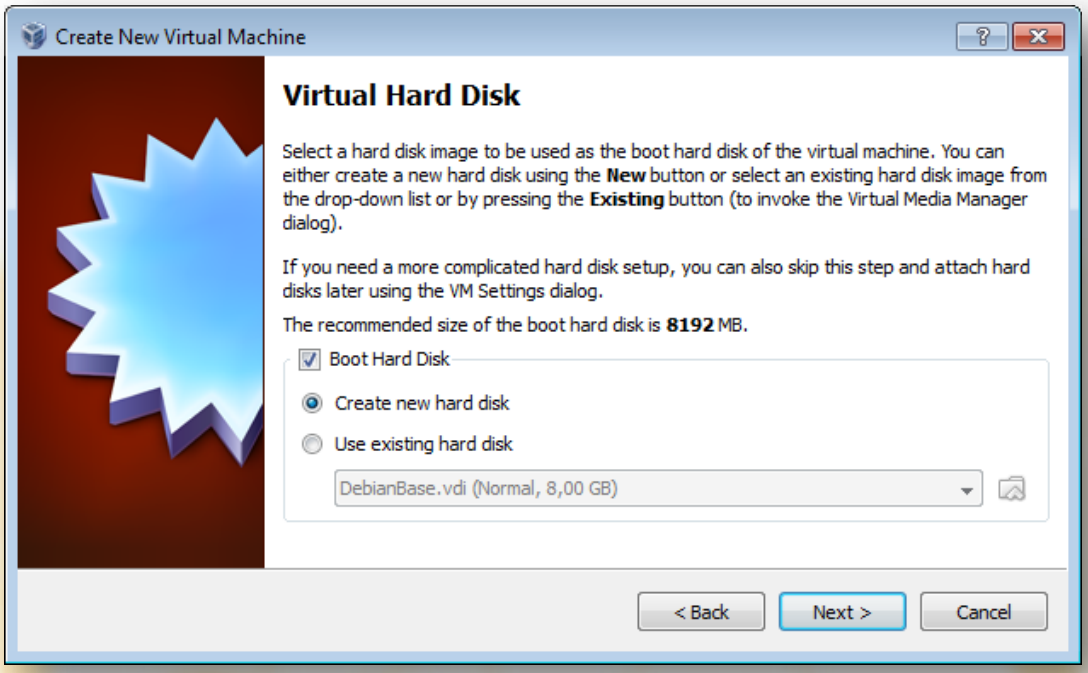

Fig. 7. Virtual Hard Disk.

Now starts create new virtual disk wizard (Fig. 8).

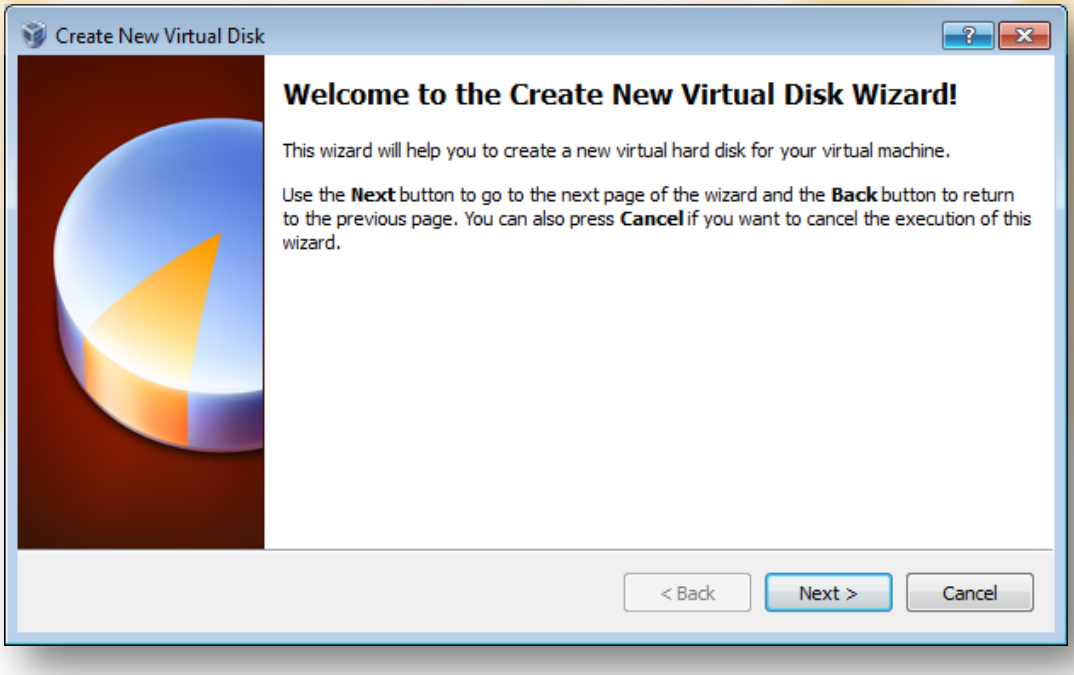

Fig. 8. Create new virtual disk wizard welcome.

We must indicate (Fig. 9) the type of disk storage (fixed-size or dynamically expanding). Fixed-size storage will create a file in our file system with the same size that we specify for this hard disk. Dynamically expanding, the default option, create a file that grows with the occupation of the virtual disk, being the max file size the same size that we specify for the hard disk.

Fig. 9. Hard disk storage type.

As we have mentioned the virtual disk will be stored as a single file in our host file system. So we must specify a file name, path and maximum size (Fig. 10).

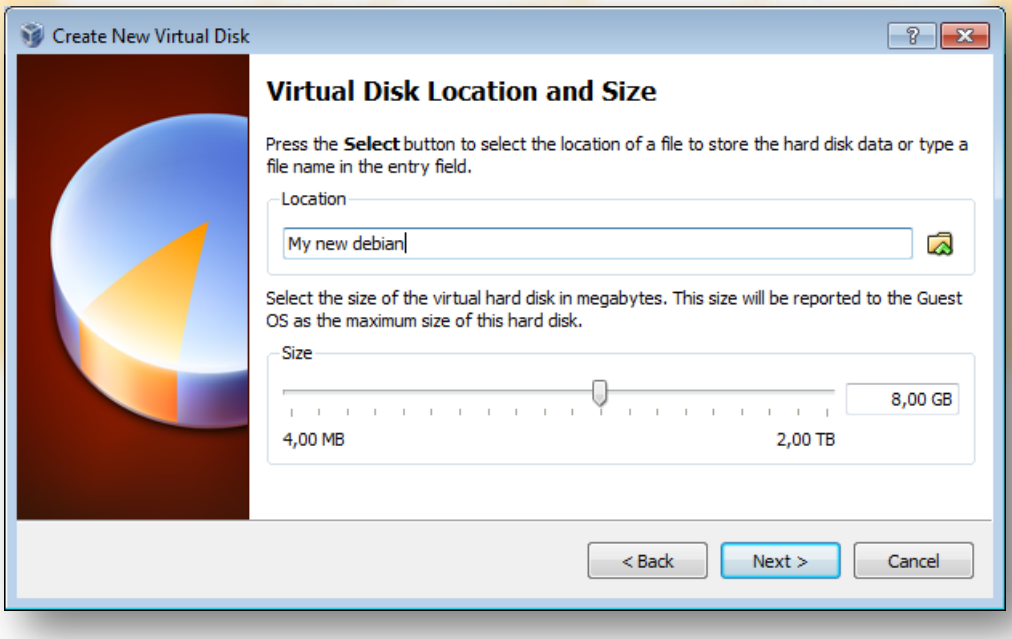

Fig. 10. Virtual disk location and size.

Finally VirtualBox shows a summary window (Fig. 11).

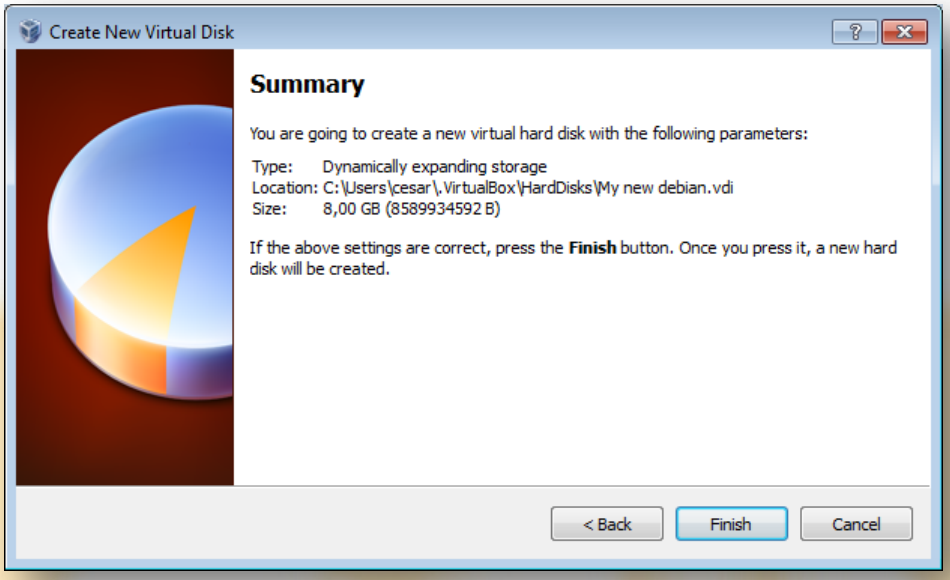

Fig. 11. Hard disk summary.

Now the machine has been created with a default hardware configuration. In Fig. 12 we can see how the new machine has been properly created and VirtualBox shows its hardware details.

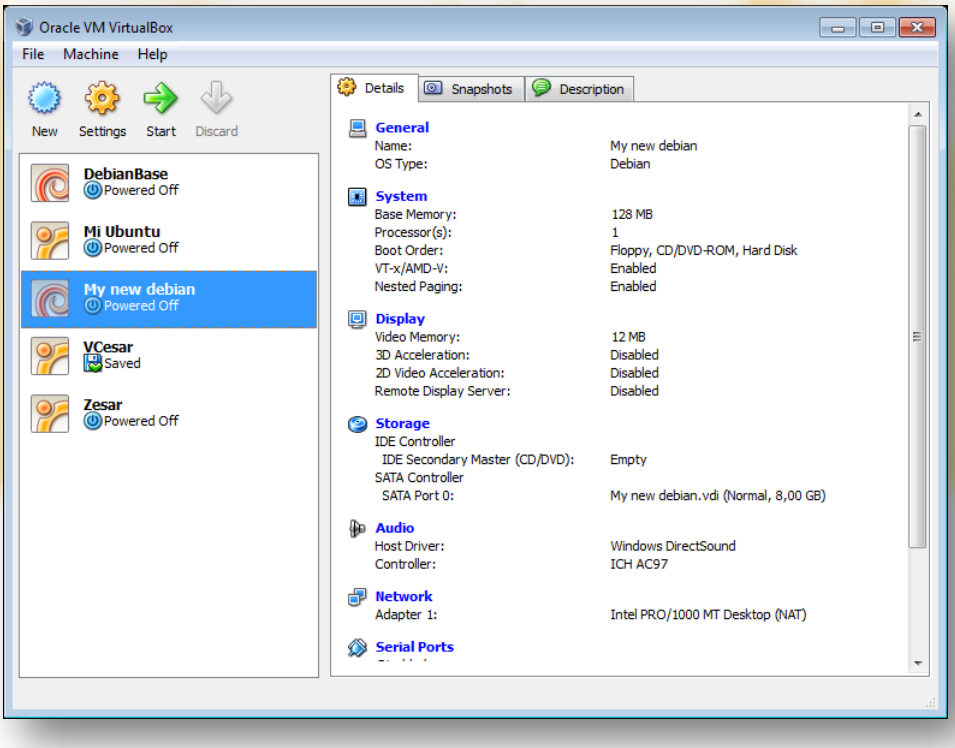

Fig. 12. Machine created.

Version to be presented on
DBTech's Helsinki Meeting, Nov'10
Please feedback any comment to: cesar@lcc.uma.es

We can change the hardware configuration changing the machine settings. We can use the Settings button, the Machine->Settings menu entry or the Settings entry in the contextual menu (Fig. 13).

Fig. 13. Settings meny entry.

The settings window (Fig. 14) allow us to change some aspects of the virtual hardware presented to this machine. We can change virtual hardware such as motherboard characteristics, CPU and RAM resources, display adapter settings, storage units and controllers, audio, network interfaces, serial ports, access to usb devices and shared folders between host and guest operating system (this last aspect requires proper configuration in both host and guest machines).

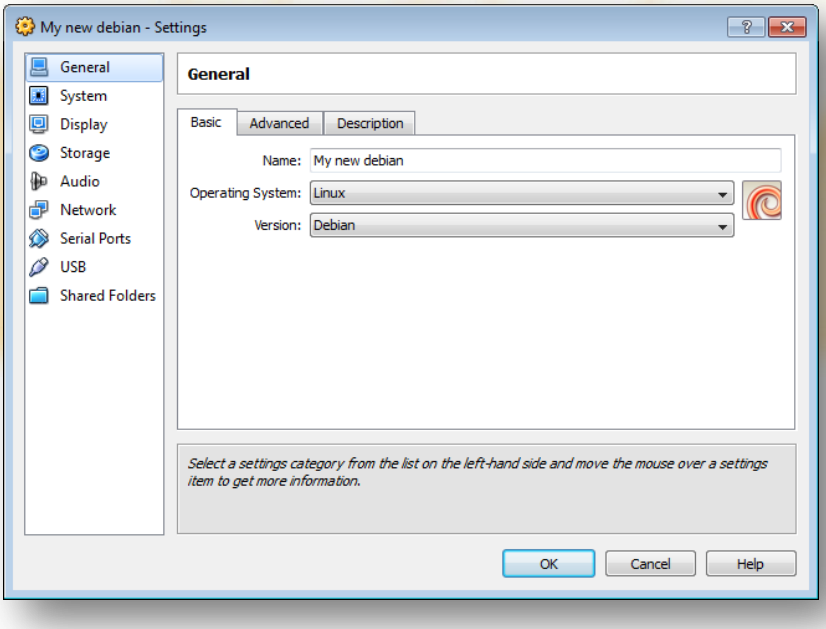

Fig. 14. Settings window.

Settings under System category are very important for our purposes. The Motherboard tab (Fig. 15) determines the size of RAM, boot order and extended features.

The amount of RAM can be changed with no problem in practically any operating system, but you must remember that the size of the virtual memory swap area is defined by most operating system based on the RAM found on installing time. The most operating systems determine the size of the swap area using the double size of your RAM size if you have 1GB or less, and 2GB if you have more than 1 GB. This is important if you install the guest OS using a small amount of memory and after that you notably increase the RAM.

The boot order determines the set of devices and the proper order where the virtual machine will search for bootable media (an installed OS, OS installation disk, live-cd, ...).

Finally we found the extended features of the virtual motherboard. Although we can change this features whenever the virtual machine is powered off, it is not recommendable to change them after the OS installation. Especially change the Enable IO APIC setting will cause most of Windows OS versions crash at boot time. You must enable the Enable IO APIC if you plan to assign more than one CPU.

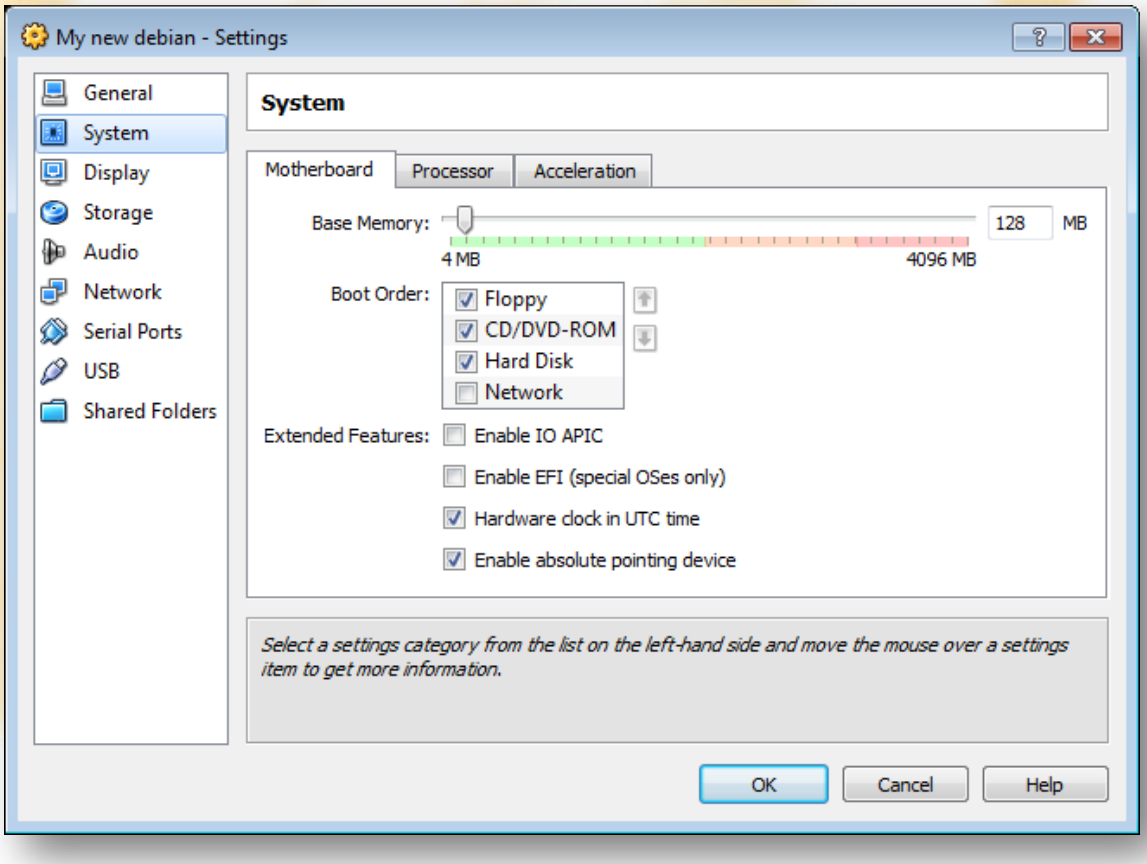

Fig. 15. Motherboard settings.

The Processor tab (Fig. 16) allows us to determine the amount of CPUs we want the virtual machine will have. The Enable PAE/NX allows virtual machine to expose the Physical Address Extensions of the host operating system, if it supports this feature, to the virtual machine.

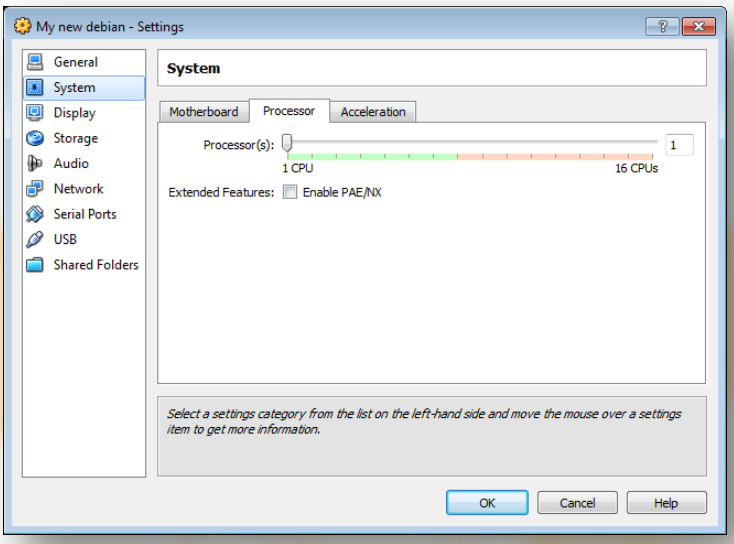

Fig. 16. Processor settings.

The last settings regarding to the System category is the Acceleration settings. Here we can choose to use or not the hardware acceleration features that our host processor and motherboard presents. To use this features our system must support them and the BIOS must be configured to allow the host OS to use them (many BIOS have a setting that enable or disable the use of this features). It is not always recommended the use of the hardware acceleration. It has been reported than VirtualBox and VMWare presents better performance results on many virtual machines when hardware settings are disabled.

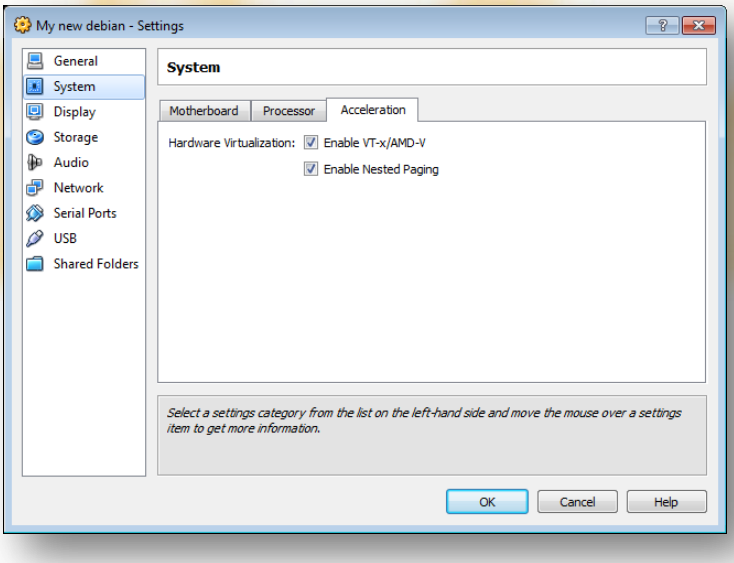

Fig. 17. Acceleration settings.

The Storage settings (Fig. 18) determine the storage devices (hard disks, CD/DVD devices and floppy units), their type and if they are "connected" to one of our physical storage units, to one virtual disk file or to one iso image. We also can select the type of the controller: IDE, SATA and SCSI.

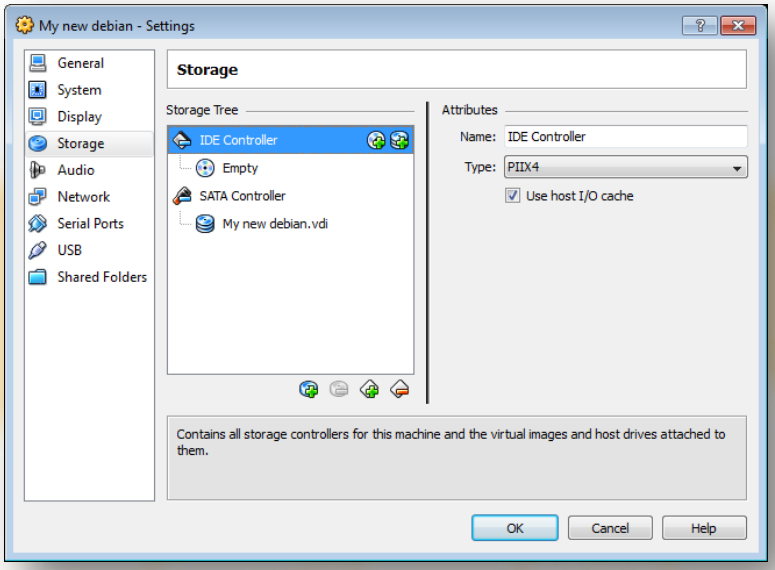

Fig. 18. Storage settings.

Commonly, when creating a new virtual machine, we use one or more CD/DVD disks to install the guest OS. We can specify now the unit that will have the first of these CD/DVDs or connect to the first ISO image. We must select the IDE Controller (included in the default hardware configuration) and select the Empty slot representing our CD/DVD unit (Fig. 19).

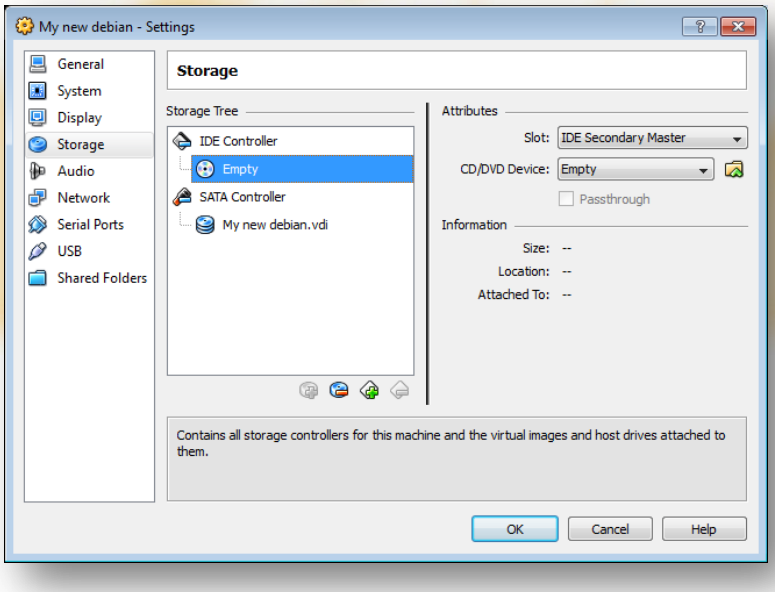

Fig. 19. CD/DVD mounting.

Now we can choose the CD/DVD disk (Fig. 20): connecting to a host drive or to an ISO image. If we choose a host drive the virtual machine will control this physical unit, and it will see the CD/DVD inserted on this unit. If we choose one ISO image the virtual machine will see this CD/DVD image as if it was a real CD/DVD inserted in its CD/DVD unit. We can only choose those ISO images declared on the VirtualBox Virtual Media Manager. We can add new ISO images starting this manager from the VirtualBox main window menu or clicking in the folder icon with a green arrow on the left of the drop-down menu.

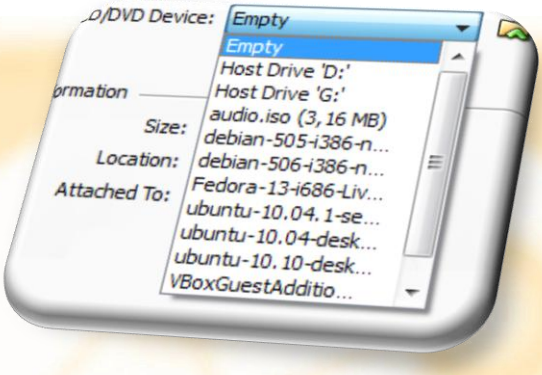

Fig. 20. CD/DVD device.

On the Virtual Media Manager (Fig. 21) we can declare all our available CD/DVD iso images. This manager also manages our virtual hard disks and our floppy images.

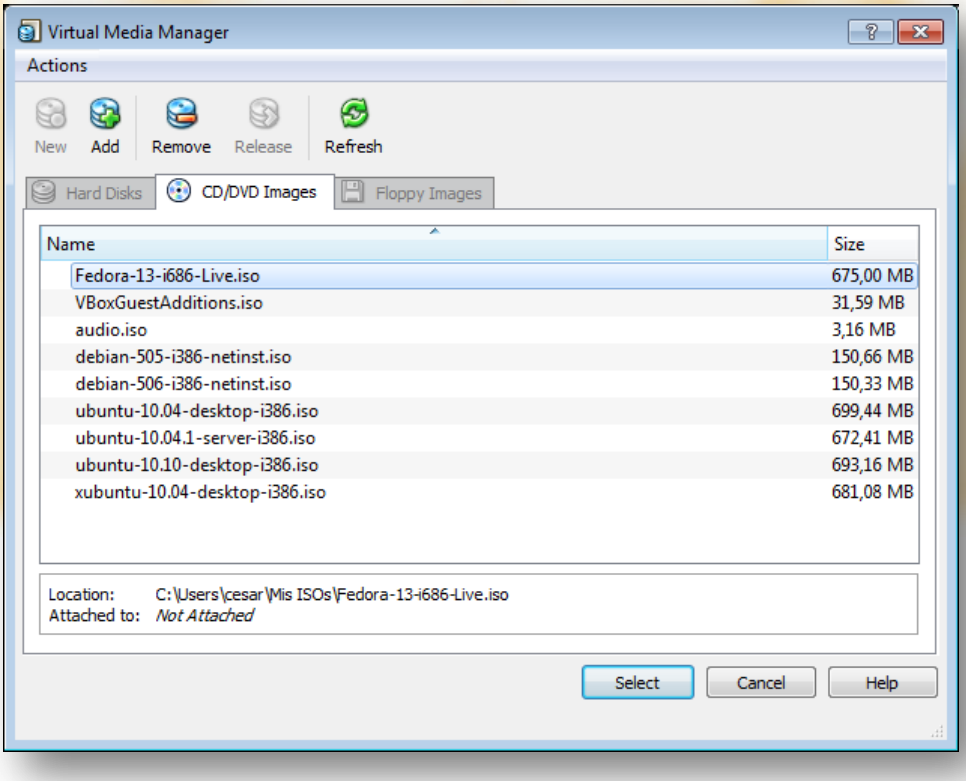

Fig. 21. Virtual media manager.

Another important category when creating a new virtual machine is the Network configuration. Network settings (Fig. 22) bring us the opportunity to declare different network interfaces. VirtualBox can emulate some real adapters; version 3.2.10 of VirtualBox emulates by default a Intel PRO/1000 MT Desktop network device. We must specify the MAC address of this adapter, by default VirtualBox use a random MAC address but we can generate new random MAC address. We also can specify if the virtual device is connected or disconnected to a network cable (simulating the disconnection of the cable).

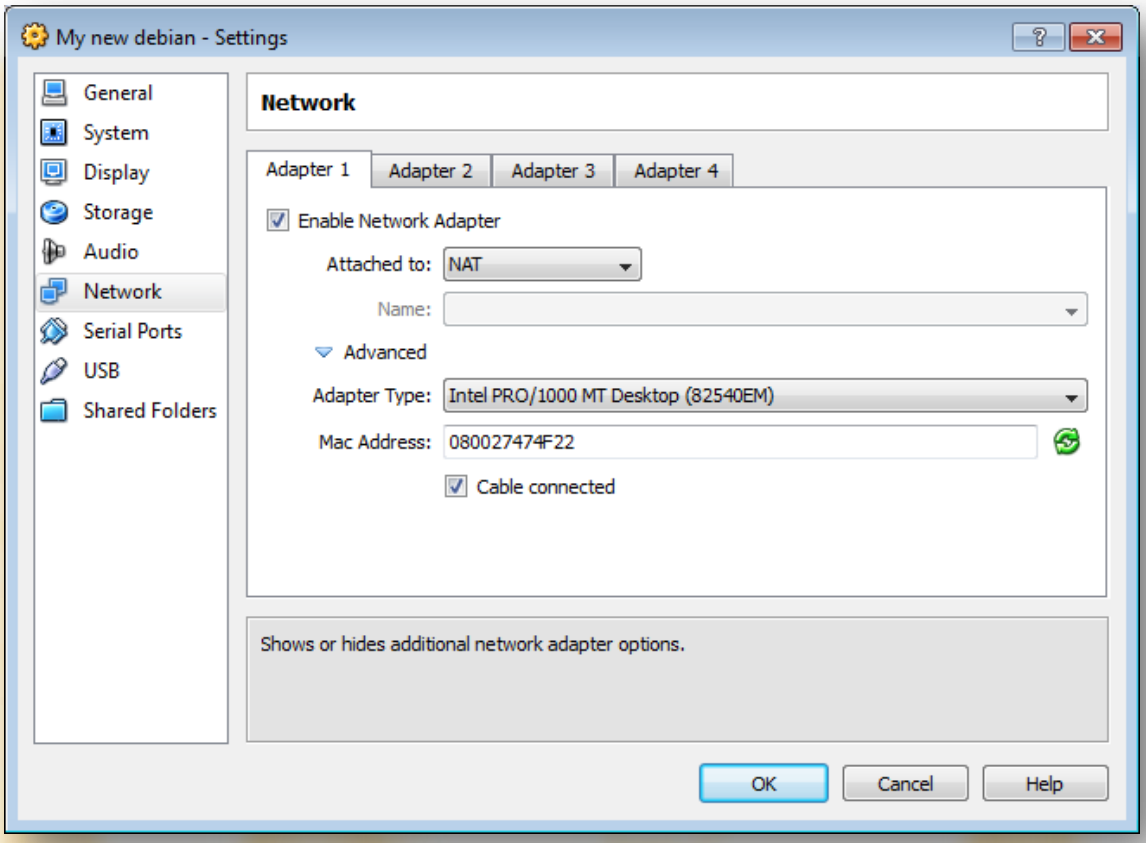

Fig. 22. Network settings.

The most important settings in the Network settings is where the network interface is attached to. We can choose different options (Fig. 23). We have different options:

- Not attached: the network interface is present and active but it is not attached to any network.
- NAT: VirtualBox acts as a NAT server to route all the outgoing traffic and it corresponding incoming related traffic to and from the host networks. This is the default setting, it allows the virtual machine to access the "external" network. Host server is inaccessible to our virtual machine using type of attach.
- Bridged Adapter: VirtualBox acts as a network Bridge between this virtual interface and the specify host interface. This is an easy way to make a virtual machine directly accessible from the external network (for example to make a virtual machine acts as a real server).

Version to be presented on
DBTech's Helsinki Meeting, Nov'10
Please feedback any comment to: cesar@lcc.uma.es

- Internal-only: VirtualBox can create different internal virtual networks. This network only connects virtual machines. By default VirtualBox creates a virtual network called “internal”, but we can create more using the global virtualbox preferences.
- Host-only: VirtualBox creates a virtual interface in our host system and connect this device to the guest virtual interface. This method is used to connect host and guest using a network. If the host has NAT/Route capabilities we can use this type of attach to manually specify routing tables. This option is perfect when we have different virtual servers external accessible and sharing the same physical network interface.

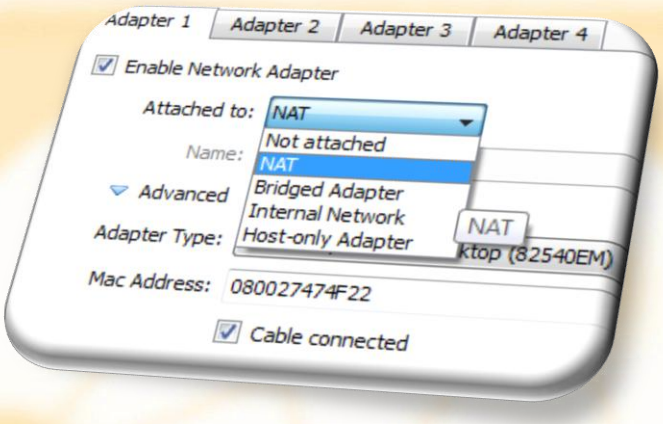

Fig. 23. Attached to.

Now we our virtual machine is ready to start up.

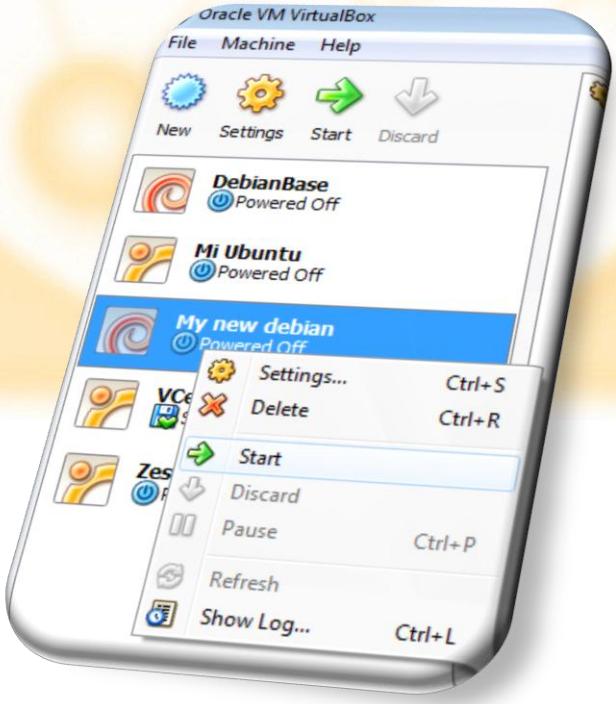

Fig. 24. Start.

If you do not attach a CD/DVD disk (host drive or ISO image) to you virtual machine, the first time it starts a First Run Wizard is launched (Fig. 25).

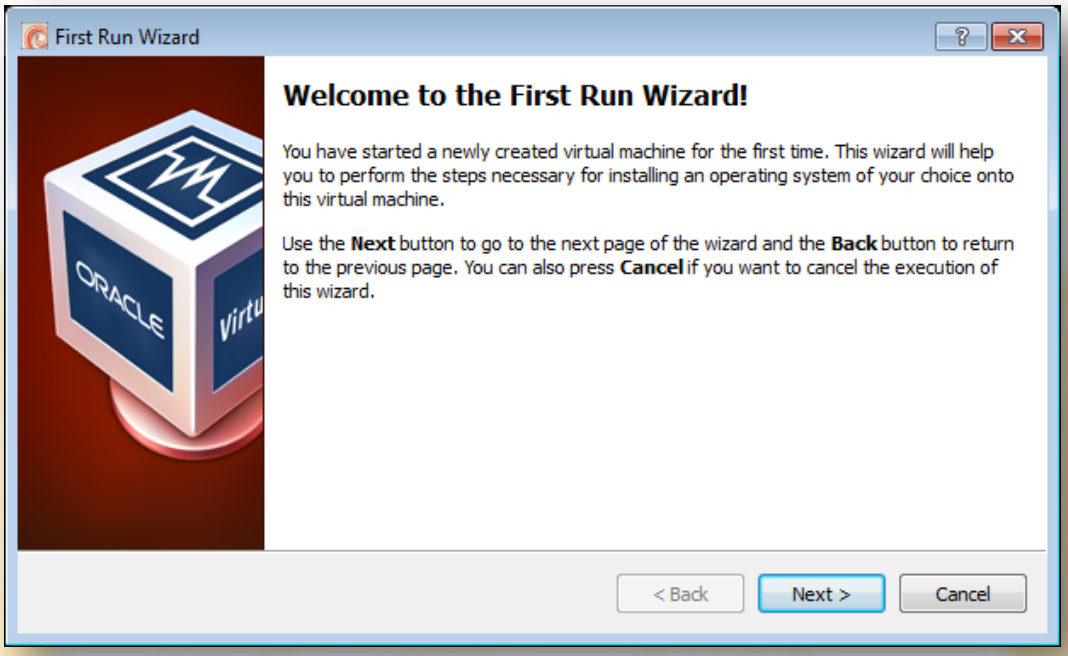

Fig. 25. First run wizard.

The wizard remembers you if you want to mount an installation media (Fig. 26). These are the same settings that we have seen on CD/DVD storage settings (Fig. 20).

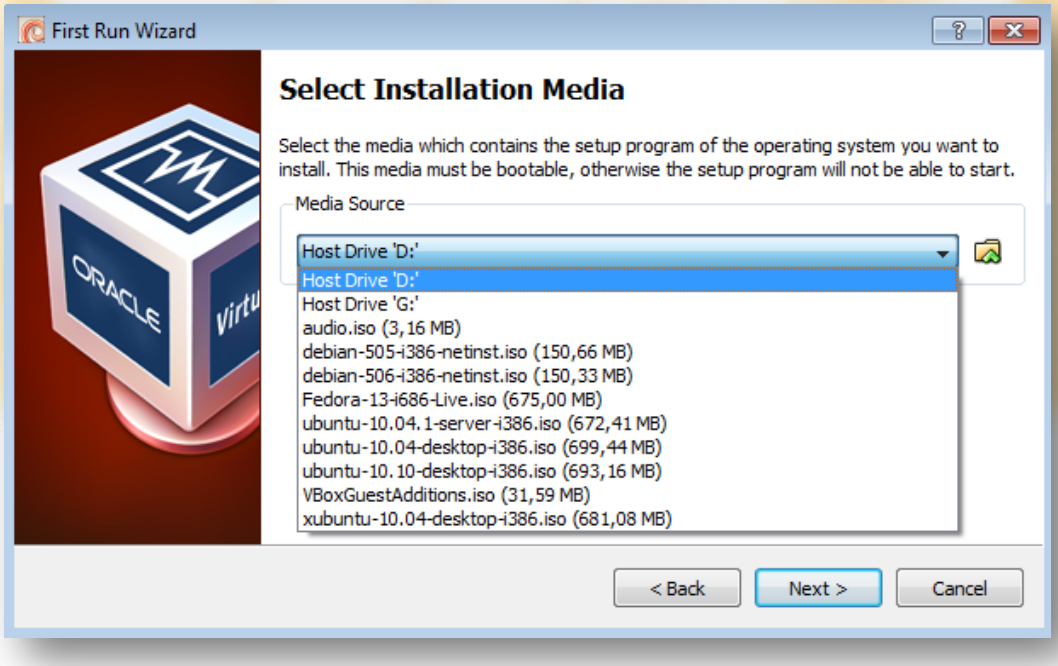

Fig. 26. Installation media.

If you cancel the process and does not specify any installation media the machine will start (Fig. 27)

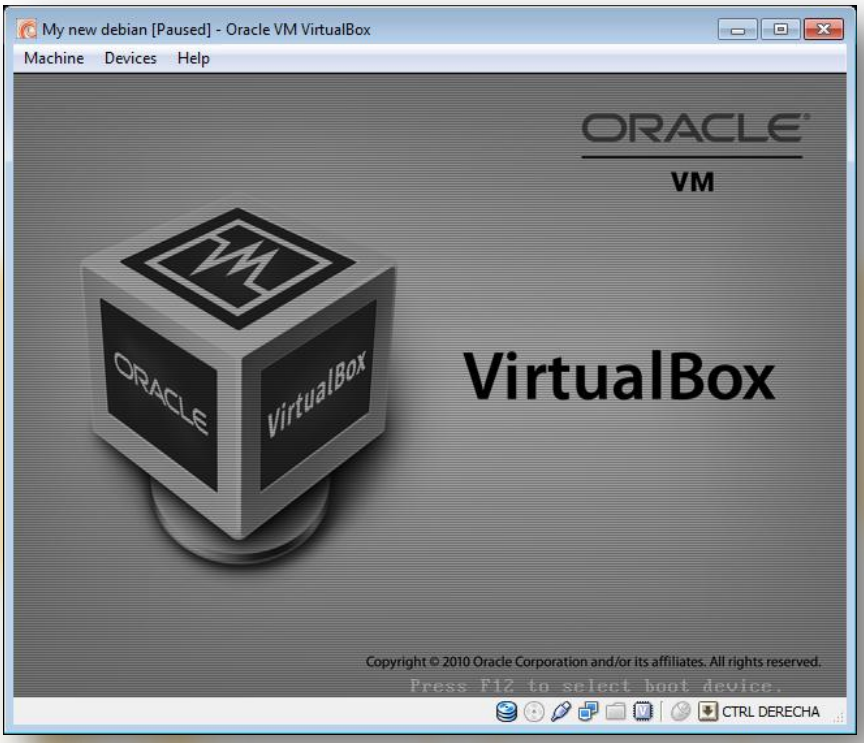

Fig. 27. Starting.

But the BIOS of our virtual machine will show a typical error message when trying to find a bootable media (Fig. 28).

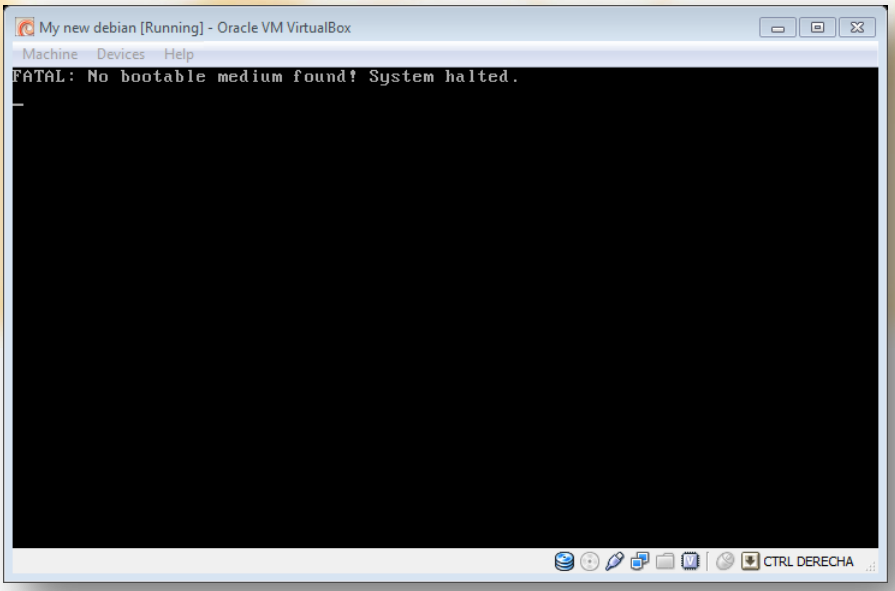

Fig. 28. No bootable media.

The Operative System

We can use many different Operative Systems. For simplicity we have choose a free and distributable version of Linux. So you can copy and distributed the resulting virtual machines without violating any license arrangement. Among the free linux versions, we think that Debian is our best choice. It is stable and server oriented, it does not evolves quickly (produces long time versions), it is widely extended, if we do not use X manager the memory finger print is only 18 MB of RAM, and we can install all our required software using only one command line (because their universal repositories contains all the software we need). Ubuntu, Fedora, OpenSUSE and CentOS are also good choices. You can also use proprietary systems, like Microsoft Windows, if you comply with licenses arrangement.

As we have chosen Debian, we must download the installation media from its webpage www.debian.org. We have chosen the last stable version 5.0.6 for i386 architectures. As we have internet access and the default hardware configuration includes a NAT interface, we choose the net-install CD version. If you plan to install on a computer with no internet access or slow access, perhaps it is recommendable to download the standard CD or DVD images. You only need to download the CD/DVD that will contain the packages you desire; if you don't known what do you need exactly you must download the full version 5 DVD or 31 CD. Why Debian full version is so long?, just because full version not only includes Debian itself but it also includes all the software included in their main repositories, this includes, for example, MySQL, PostgreSQL and PGPool-II.

After attaching the DVD unit of our virtual machine to the Debian net-install 5.0.6 iso image, using storage setting (Fig. 20), we start our virtual machine and bootable CD now is launched by our virtual machine (Fig. 29).

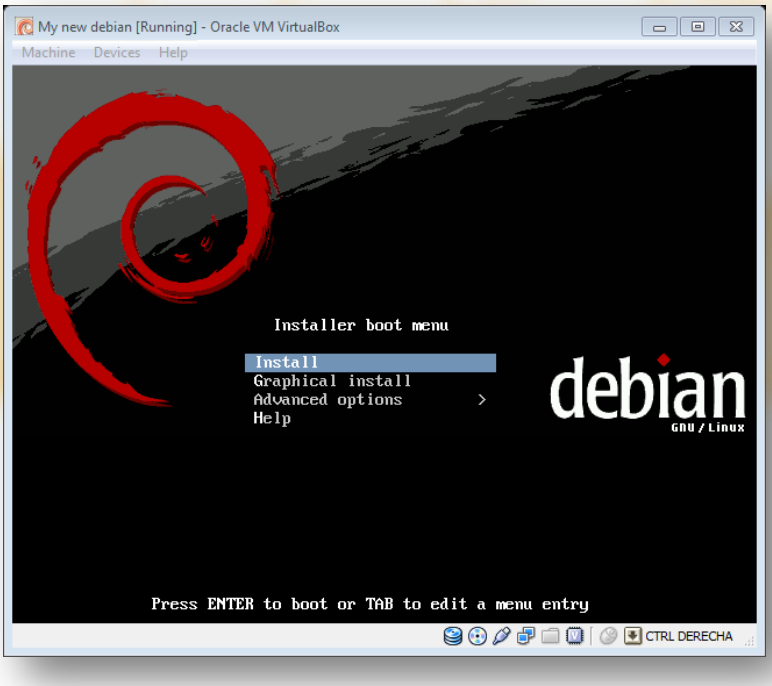

Fig. 29. Debian install menu.

We are going to install a minimal version (this means no X manager, only text mode). We can use the graphical install to install a text based system, but it is faster and consumes less resource to use the default non-graphical installer. So we select Install.

Now the installation process starts. This process needs some settings that the installer will ask us. The first one is the language we want to use (Fig. 30), in this example we use English.

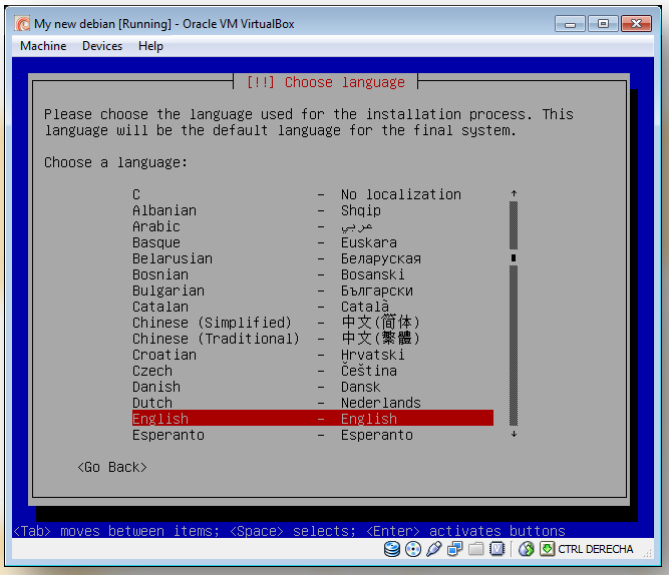

Fig. 30. Choose language.

The second one is our location that will be used to determine regional configurations and the nearest universal repositories. If your location does not appear select “other” (Fig. 31).

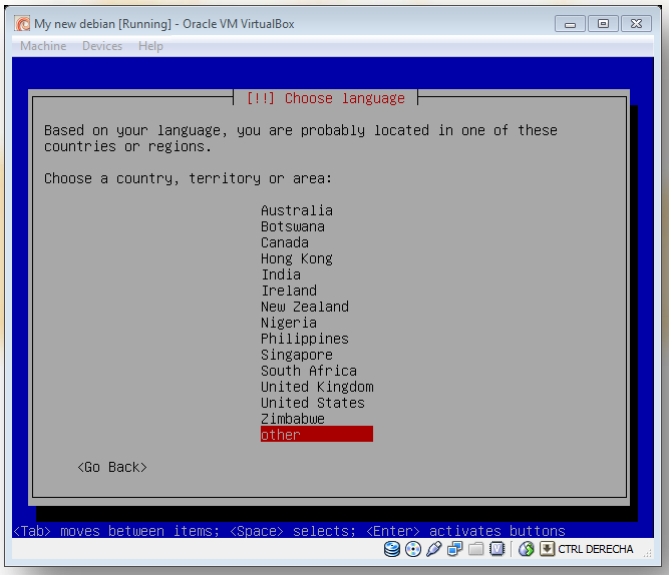

Fig. 31. Region.

These example has been done from Europe (Fig. 32).

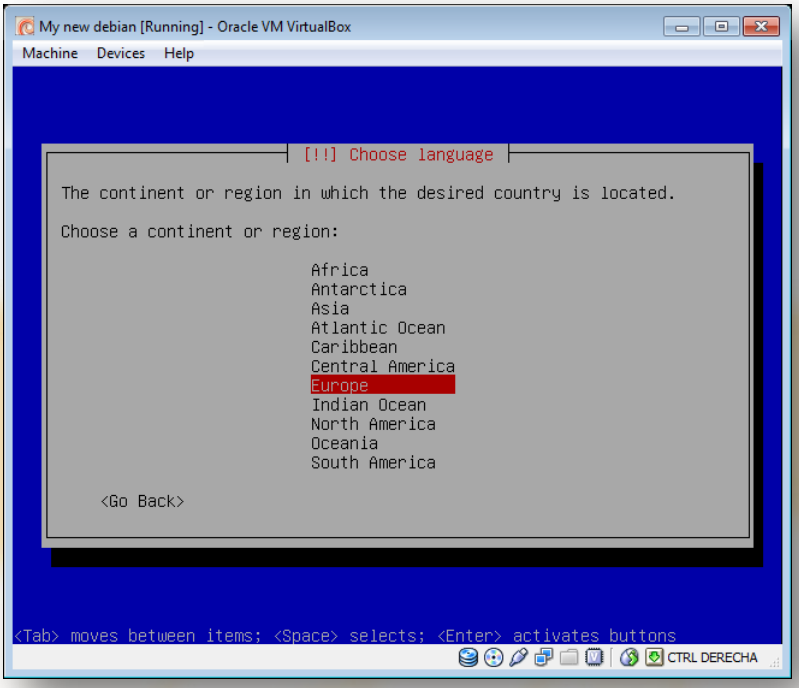

Fig. 32. Region Europe.

More precisely from Spain (Fig. 33).

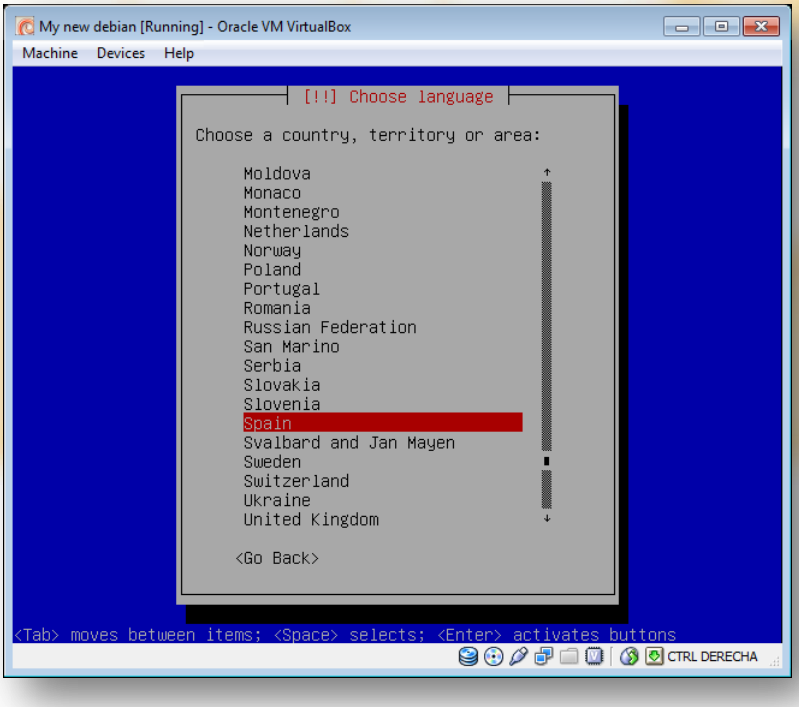

Fig. 33. Spain.

The next question is about our keyboard layout. In the example was used a Spanish keyboard. So we select the Spanish keyboard layout (Fig. 34).

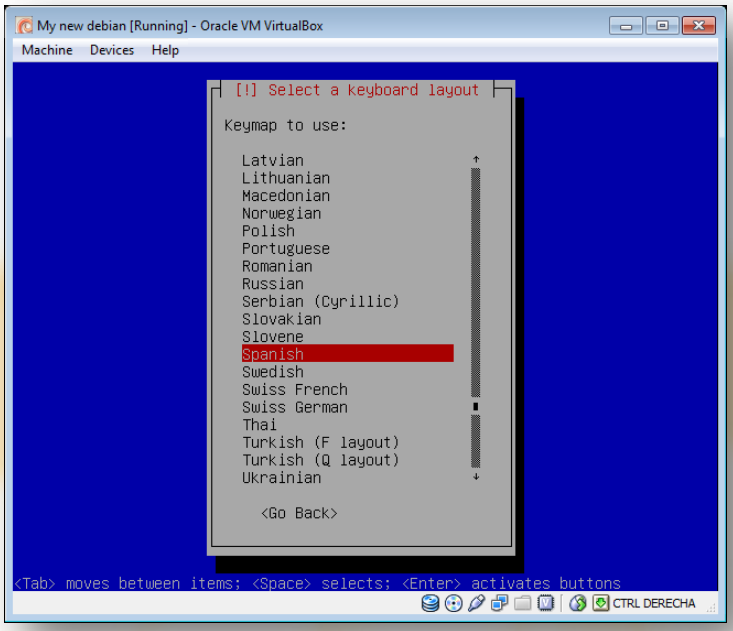

Fig. 34. Keyboard layout.

Now is the turn of the network configuration. The hostname of our machine (), for example debian.

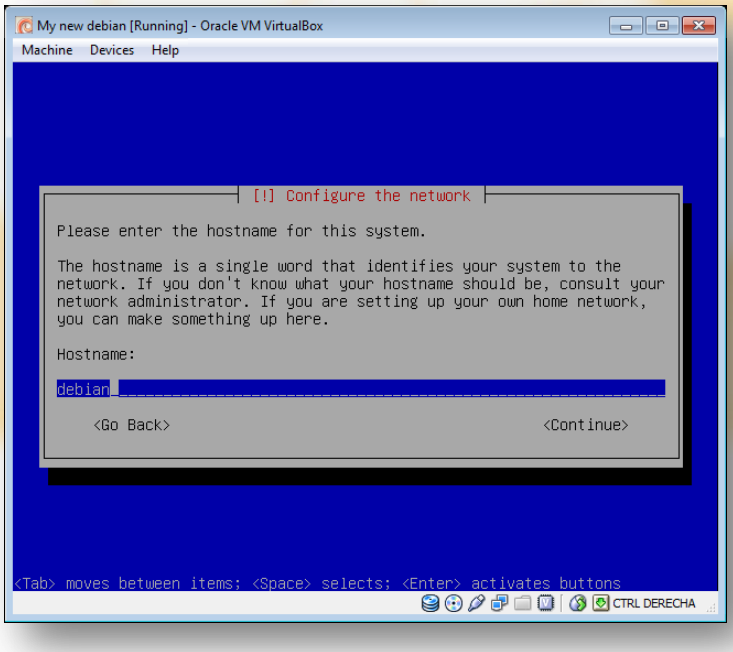

Fig. 35. Hostname.

Second the domain name (Fig. 36). As we do not require complex network configurations, we left this field empty.

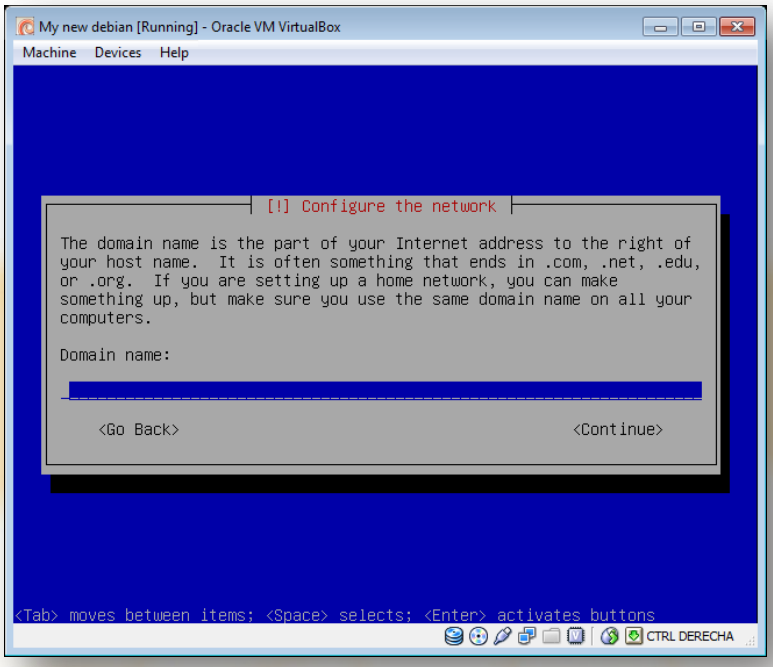

Fig. 36. Domain name.

The clock configuration needs to know a more precise location in Spain (Fig. 37), to determine the time zone.

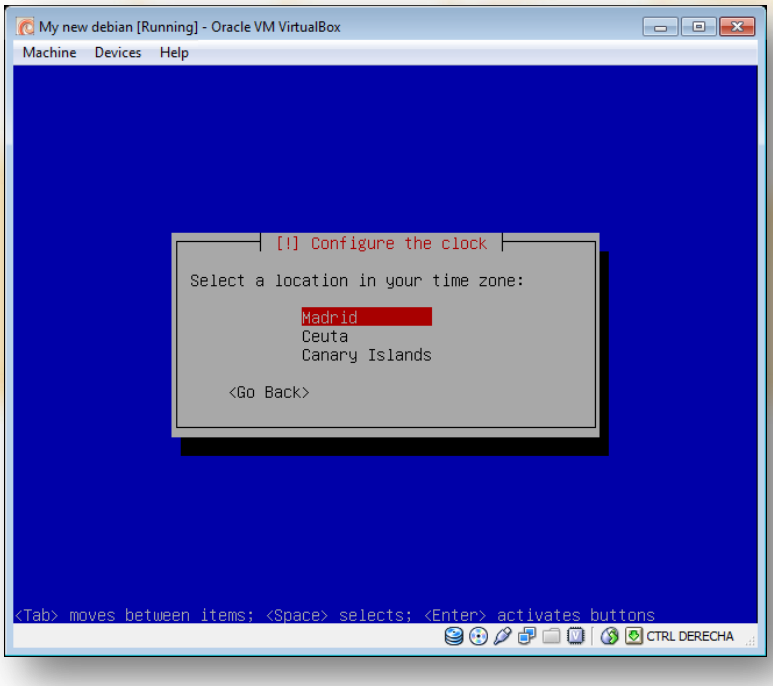

Fig. 37. Time zone.

Now we must select the disk where Debian is going to be installed. The partition wizard starts. This is an easy example, so we select the entire (Fig. 38) virtual disk (Fig. 39).

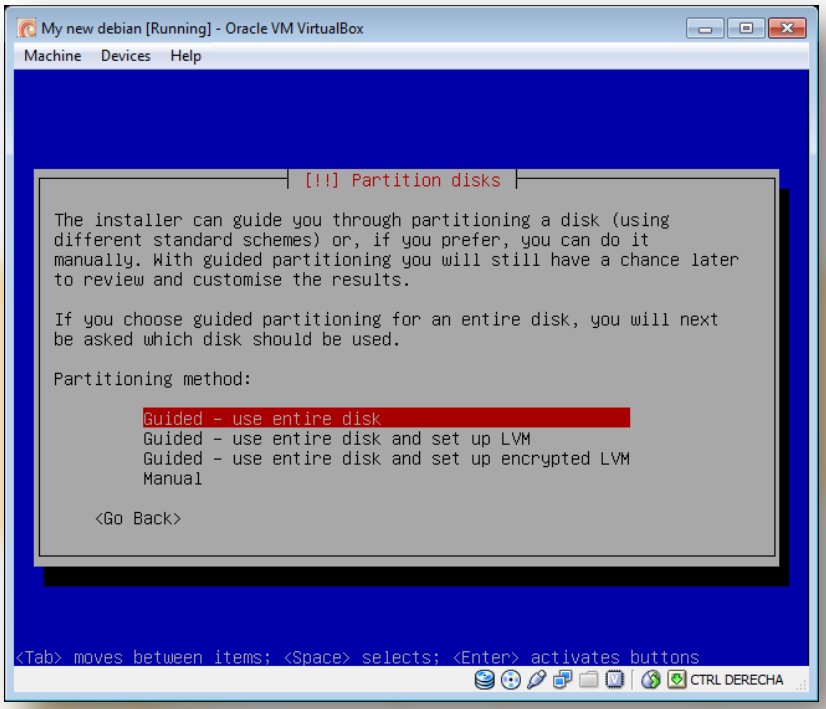

Fig. 38. Partition disks.

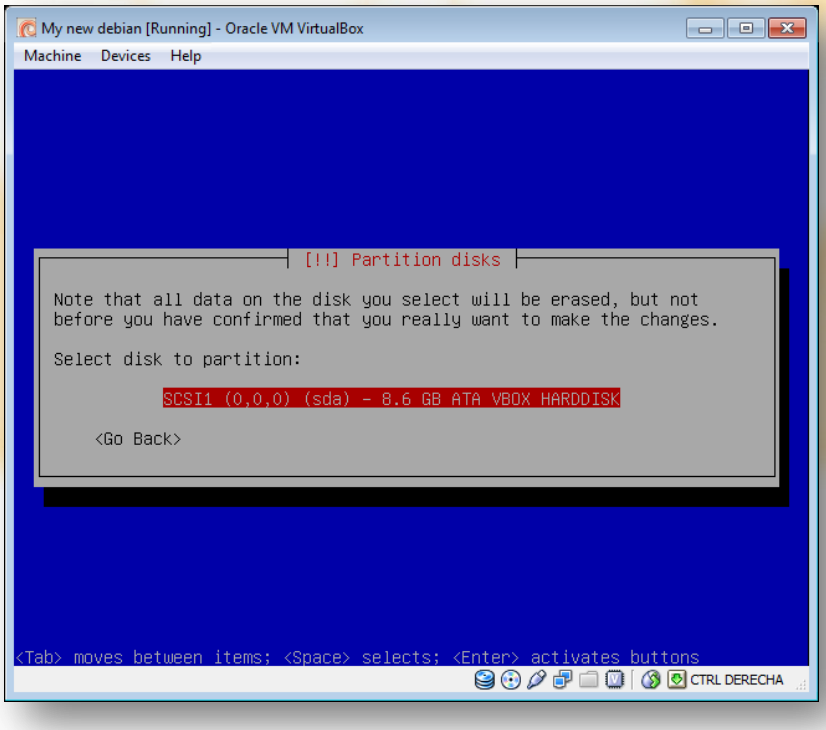

Fig. 39. Disk selection.

And we will put all data together on the same partition (Fig. 40).

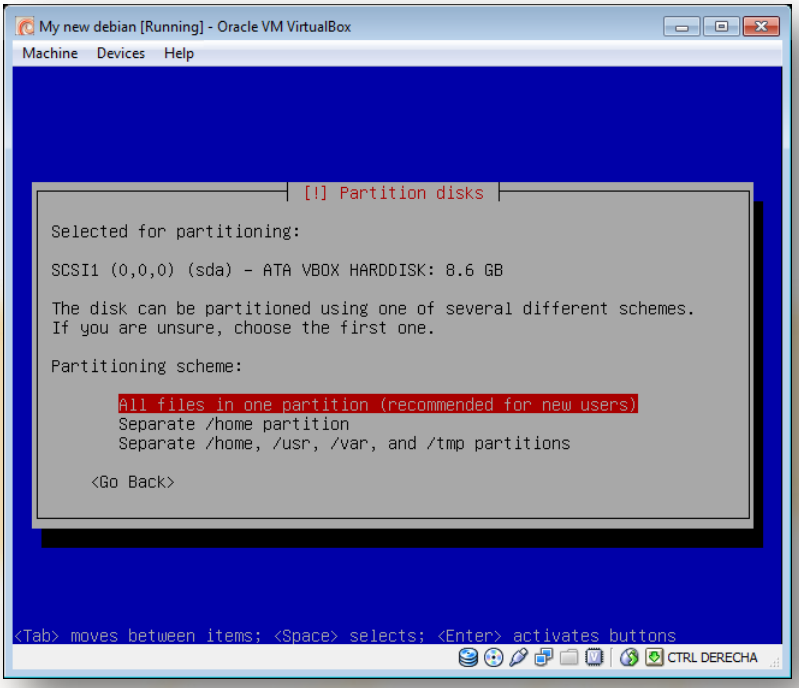

Fig. 40. One partition.

Now we finish the partition process (Fig. 41).

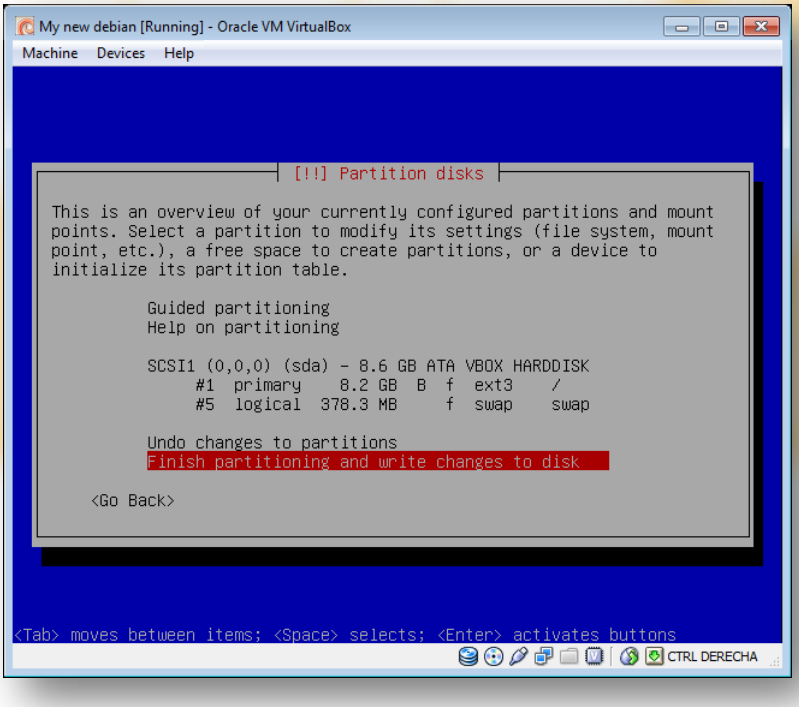

Fig. 41. Finish partitioning.

And of course, we confirm (Fig. 42). Note that the default option on this step is <No>, you cannot damage disk by an accidental multiple ENTER keystroke. Select <YES>.

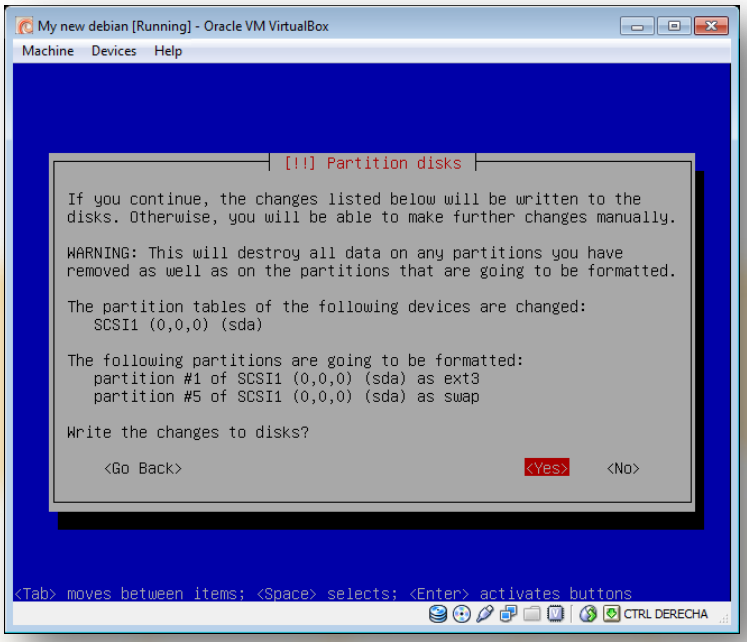

Fig. 42. Confirm changes.

Our root (the administrator user in Linux) will need a password (Fig. 43). Introduce it and verify it again (Fig. 44).

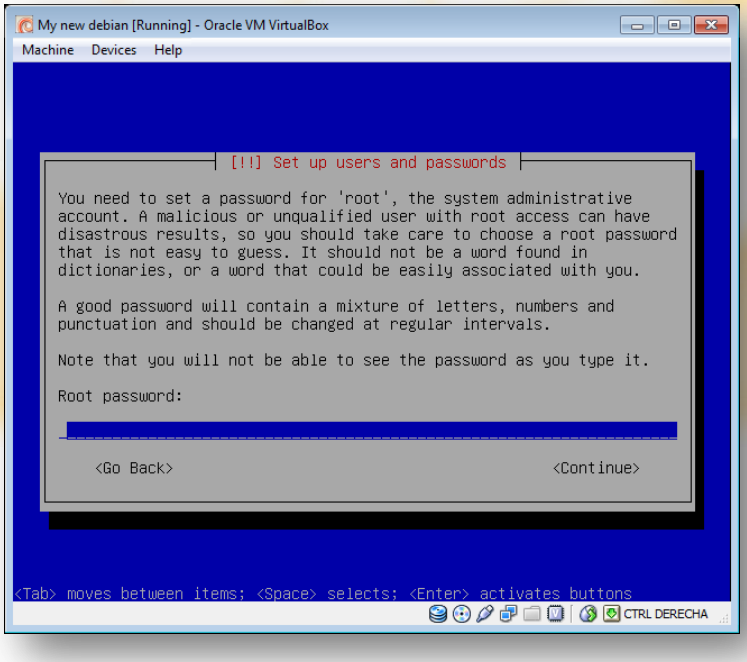

Fig. 43. Root password.

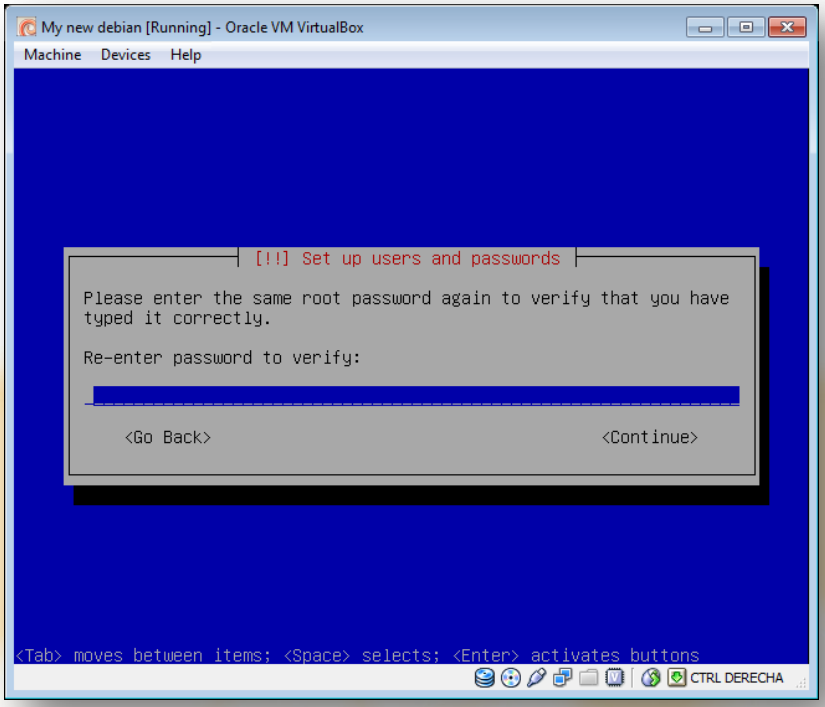

Fig. 44. Verify root password.

Debian prefers that you create a non-root user (your default identity). So it ask for the user full name (Fig. 45), your identity.

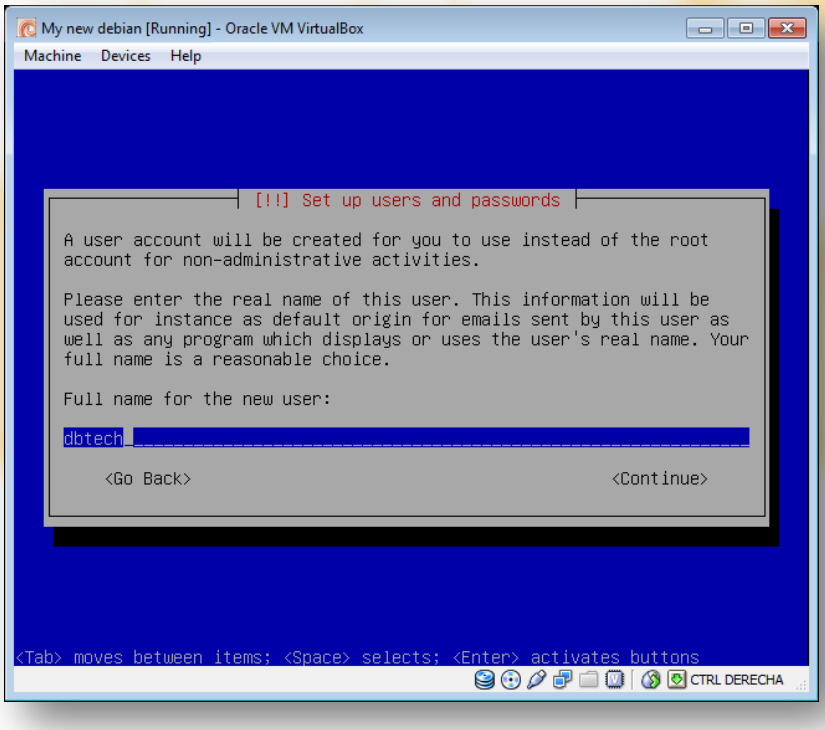

Fig. 45. Full user name.

Now let's introduce the username of the account for this user (Fig. 46).

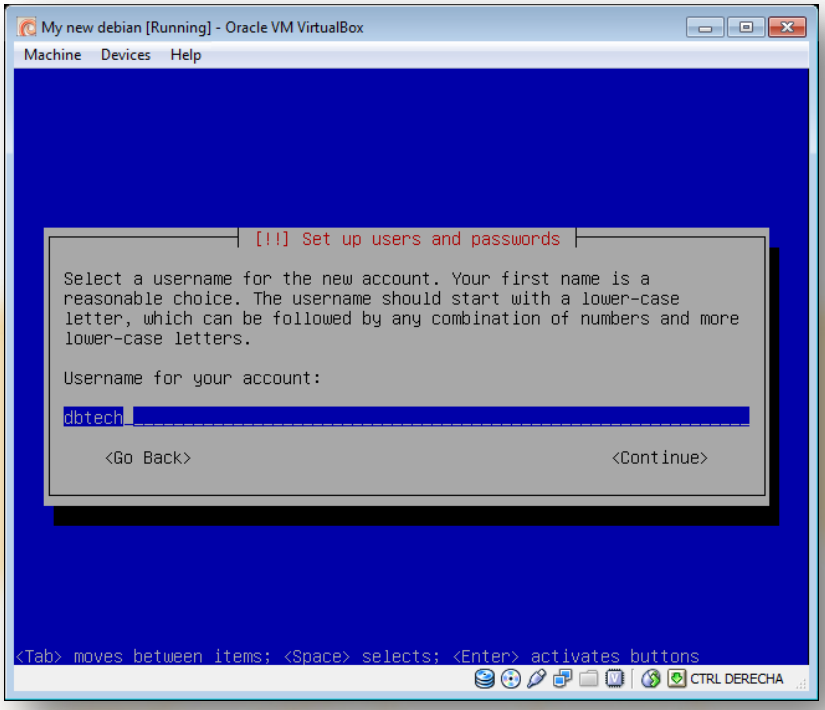

Fig. 46. Username account.

And, again, introduce the password (Fig. 47) and its verification (Fig. 48).

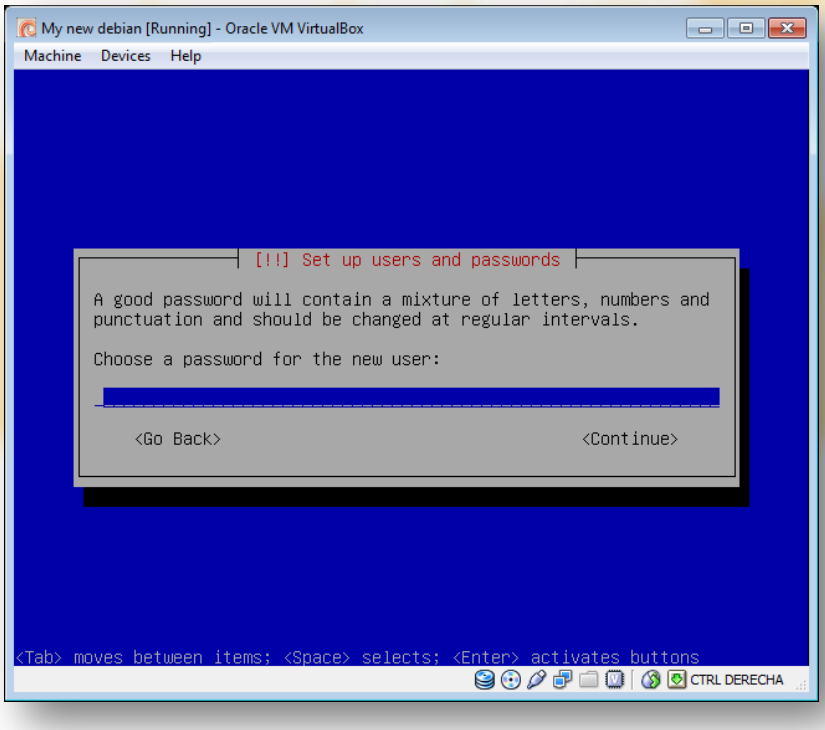

Fig. 47. User password.

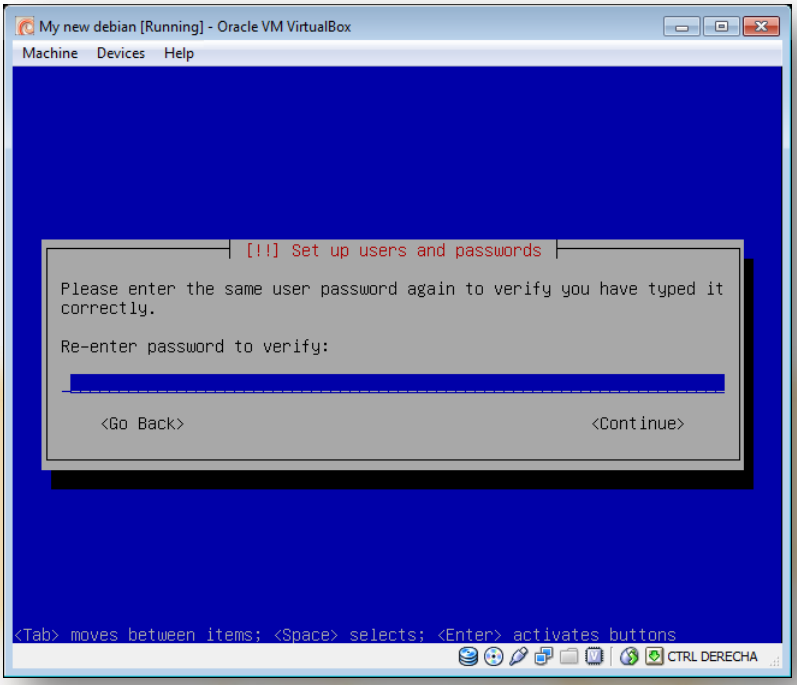

Fig. 48. User password verification.

Now based on our location it suggests a preferred repository, first selecting the country (Fig. 49).

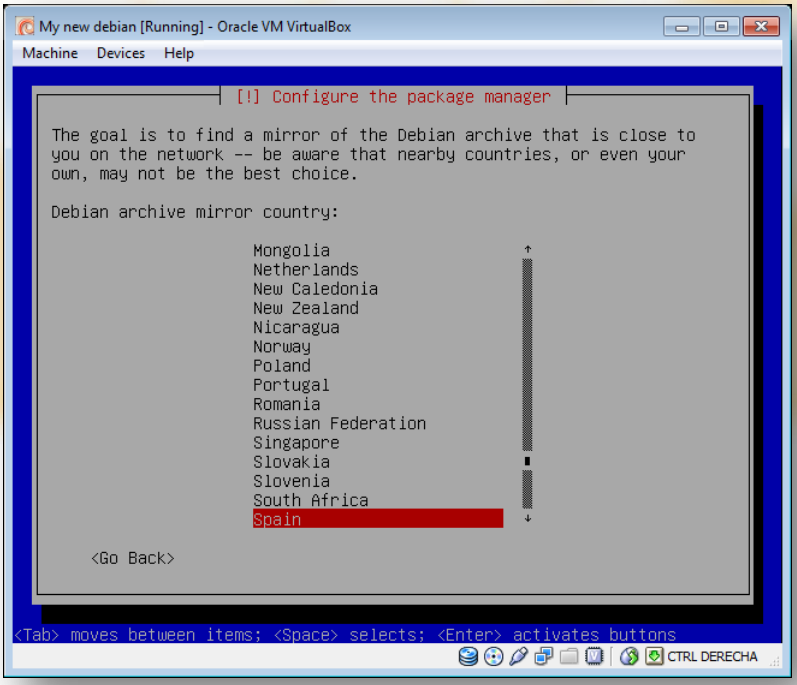

Fig. 49. Repository country.

And second the archive mirror in this country (Fig. 50).

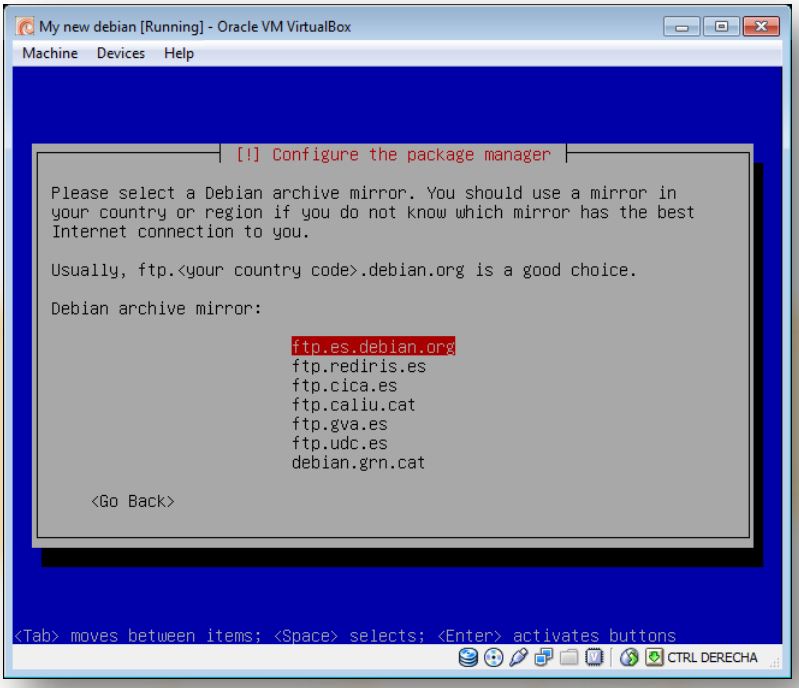

Fig. 50. Archive mirror.

If we use a proxy to access external network we must specify (Fig. 51) now the proxy URL (that also includes username and password, if required). If you have direct access, left it blank.

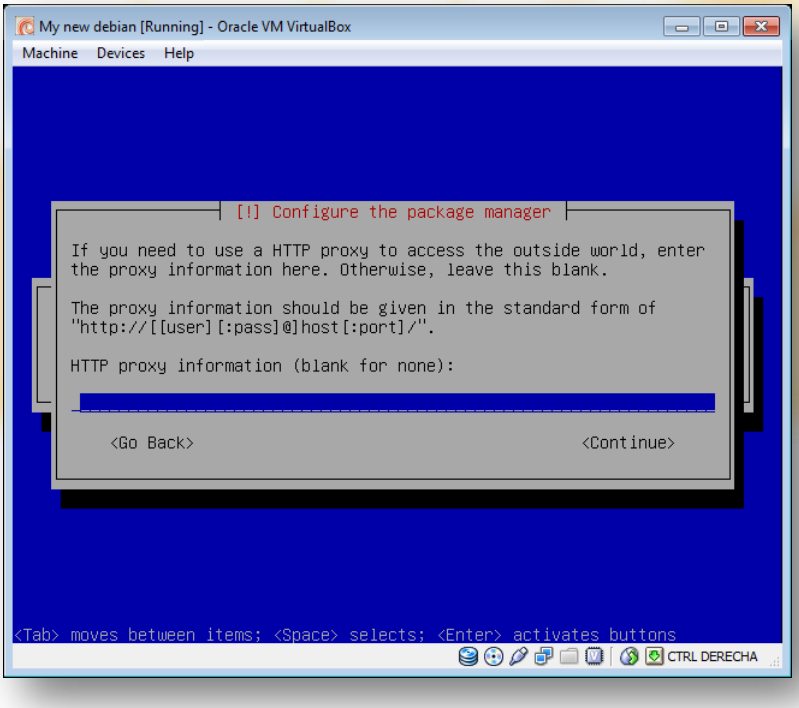

Fig. 51. Proxy.

Dou you want to participate in a package usage survey (Fig. 52)?

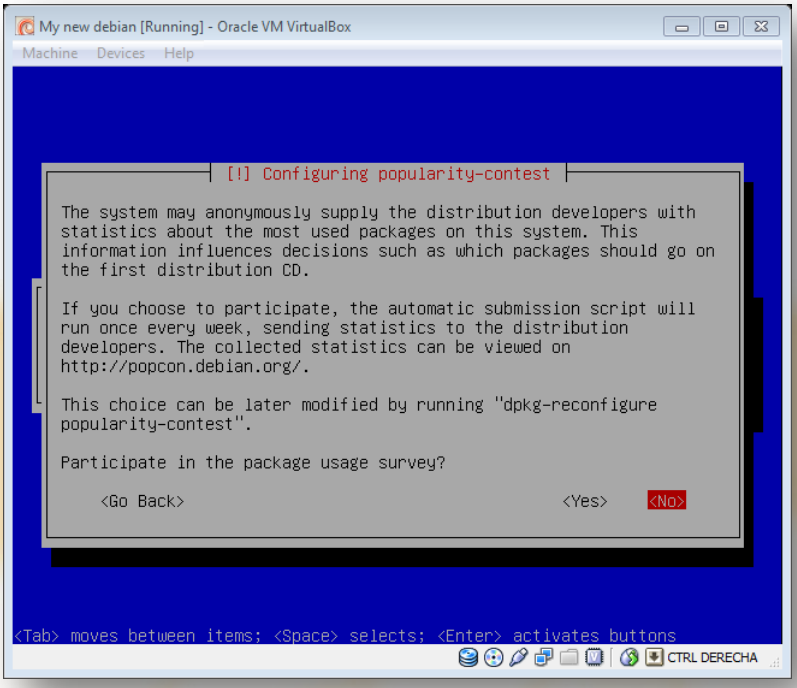

Fig. 52. Popularity-contest.

And now we must select what we want to install (Fig. 53).

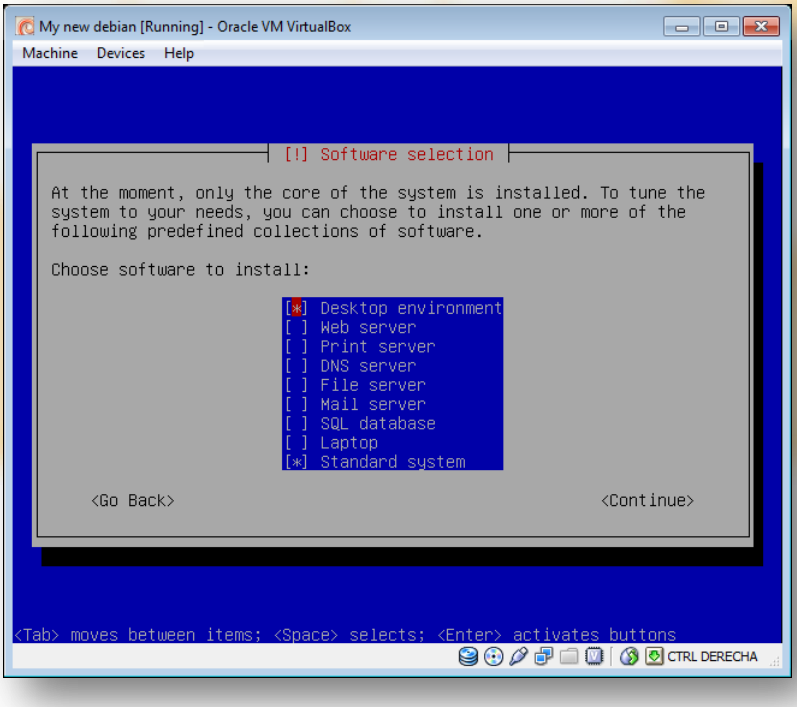

Fig. 53. Software selection.

As we want a minimal installation, we deselect all (Fig. 54).

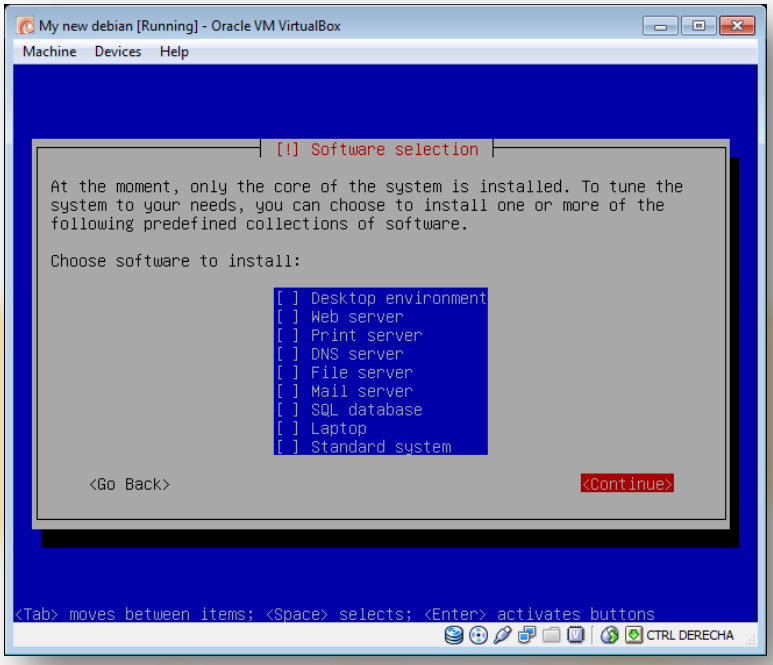

Fig. 54. Minimal installation.

The last thing is the boot loader, by default GRUB (Fig. 55).

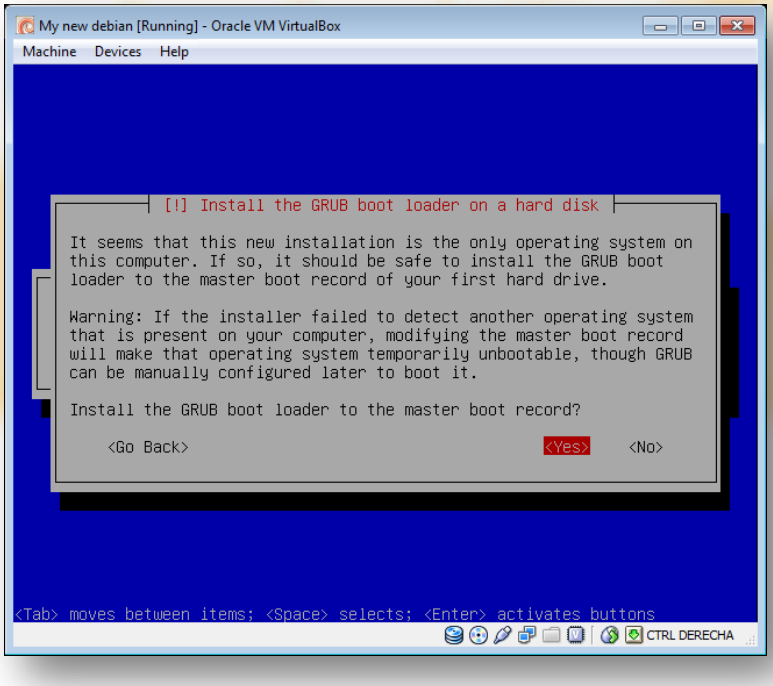

Fig. 55. GRUB.

Now the system is going to restart (Fig. 56).

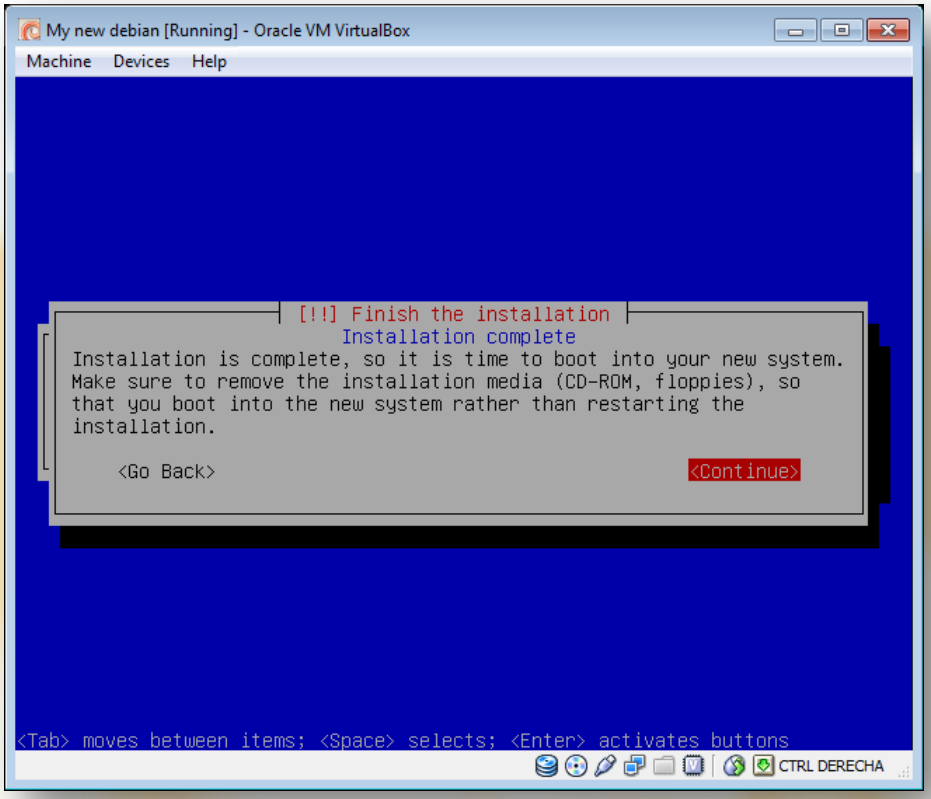

Fig. 56. Installation complete.

Don't forget to remove installation media, you can do this on runtime using the Devices ->CD/DVD Devices menu entry (Fig. 57).

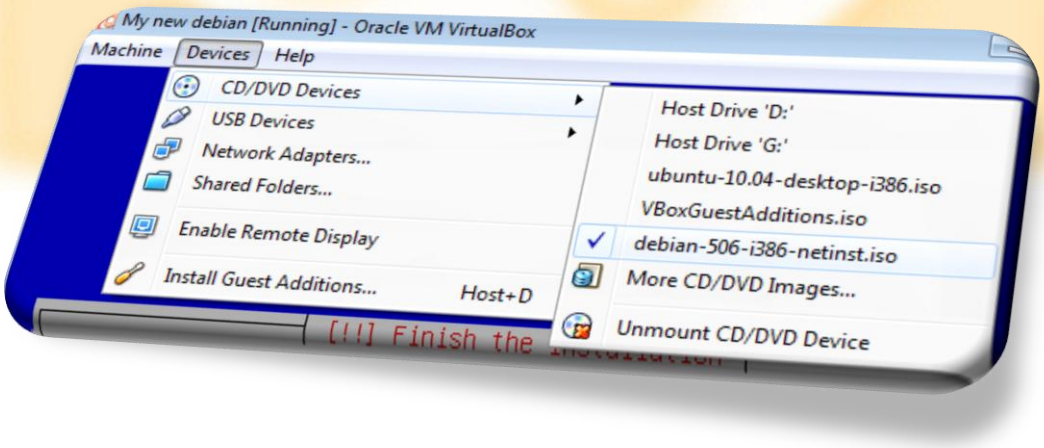

Fig. 57. CD/DVD Devices menu entry.

After that confirm the installation complete windows (Fig. 56) and your virtual machine will restart. If nothing goes wrong you will see the Debian boot process and finally the system will wait for a user login (Fig. 58).

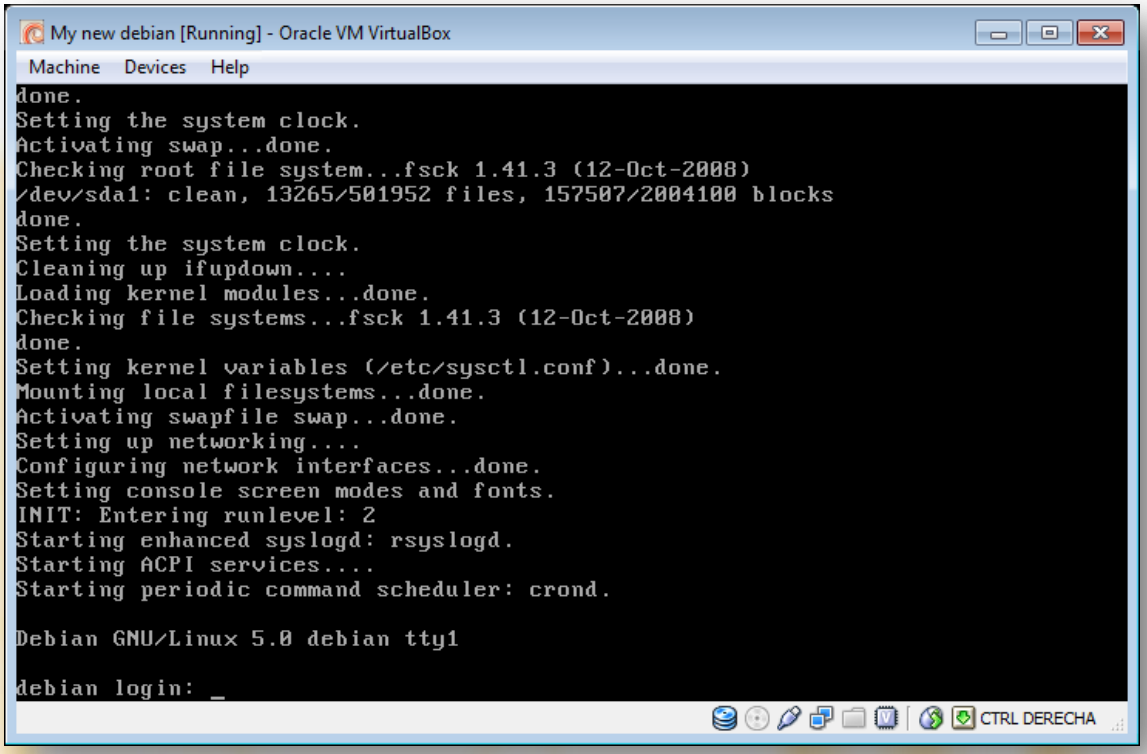

Fig. 58. Login.

And what now?.

If you want to start a session introduce the username and password, or if you want to do administrative tasks login as root.

If you want debian to shutdown your machine, then introduce this command as root:

shutdown –h now

or if you have login as a non-root user introduce:

sudo shutdown –h now

And what about a text editor?:

- If you have used before use “vi”.
- If you are not familiarized with “vi”, the “nano” is a better option.

Version to be presented on
DBTech’s Helsinki Meeting, Nov’10
Please feedback any comment to: cesar@lcc.uma.es

If proxy settings changes after install, how can I change them?. A program can define its own proxy settings, but in debian it is usual that a program try to read two environment variables (http_proxy and ftp_proxy) this variable must have the form:

```
http://username:password@proxyserver.netport/
```

So introduce as root something like this:

```
export http_proxy=http://username:password@proxyserver.net:port/
export ftp_proxy=http://username:password@proxyserver.netport/
```

in your /etc/bash.bashrc or:

```
http_proxy=http://username:password@proxyserver.net:port/
ftp_proxy=http://username:password@proxyserver.netport/
```

in your /etc/environment file.

Database software

Among the different software we can use to test distribution and replication in databases, we have select MySQL and PostgreSQL. Why?. Mainly because they can do all we want to show, they are lightweight databases and they are free. Can I use proprietary database to do test distribution and replication concepts?. Of course, and perhaps they can do it better than the databases we have selected. Perfects candidates, for example, could be Oracle and IBM DB2. But you need a great computer, or you have to be an experienced user in database tuning, to run multiple virtual machines with Oracle or DB2 servers installed and running.

MySQL & PostgreSQL

MySQL and PostgreSQL are both Relational Databases Management Systems with existing versions licensed under GNU GPL public license and they both are present on the universal repositories of the most popular Linux distributions. Although you can download MySQL and PostgreSQL for different platforms (linux, windows, Mac, ...) from their respective web sites, installing into linux from the universal repository is the most easy way to have two "ready to use" database servers.

MySQL was born on the opensource enterprise MySQL AB, initially stabilised on Sweden in 1995. The name has not a clear precedence. Originally it was developed as an alternative API to mSQL by Michael Widenius, one of the MySQL AB cofounder, but his daughter is also called *My*.

MySQL has been evolving and since January 2008 is a subsidiary of Sun Microsystems that is also a subsidiary of Oracle Corporation since April 2009. One particular characteristic of MySQL is the pluggable storage engine architecture. MySQL can use different storage engines like InnoDB, MyISAM, NDB, Memory, Merge, Archive, CSV, Federated and more. We are very interested in NDB and Federated, because they allow us to test replication and distribution concepts.

On the other side we have PostgreSQL. Extrated from its web site, we found that this server born in 1986 by the hand of Michael Stonebraker, and it was Initally called Postgres, as a successor of Ingres. Postgres was commercialized as Illustra that was bought by Informix and integrated into its Universal Server. Finally Informix was purchased by IBM in 2001. But in 1995 two Ph.D. students from Stonebraker's lab, Andrew Yu and Jolly Chen, replaced Postgres' POSTQUEL query language with an extended subset of SQL, renaming the system to Postgres95. In 1996 Postgres started its open source way and PostgreSQL 6.0 appeared.

Now PostgreSQL 9.0 is the latest version, and it still remains as an open source object-relational database system. It is fully extended among organizations, government agencies and companies. At the beginning of this module PostgreSQL's latest version was 8.4, which does not integrate replication support. External tools, as Pgpool-II, are used to provide replication. Although version 9 now introduces built-in replication support, we use Pgpool-II and postgreSQL 8.3 version to shown how replication works using an external replication layer.

MySQL & PostgreSQL are a perfect couple for us to introduce basic replication and distribution concepts. They can be installed, configured and started in Debian with only one line (Fig. 59).

Version to be presented on
DBTech's Helsinki Meeting, Nov'10
Please feedback any comment to: cesar@lcc.uma.es

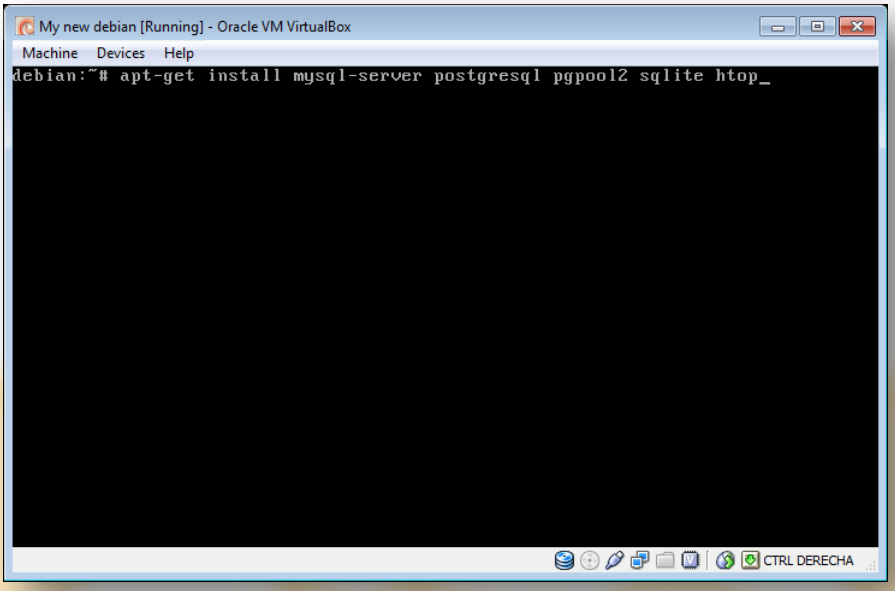

Fig. 59. Installing databases.

The only question you are required to answer is the MySQL root's password (Fig. 60).

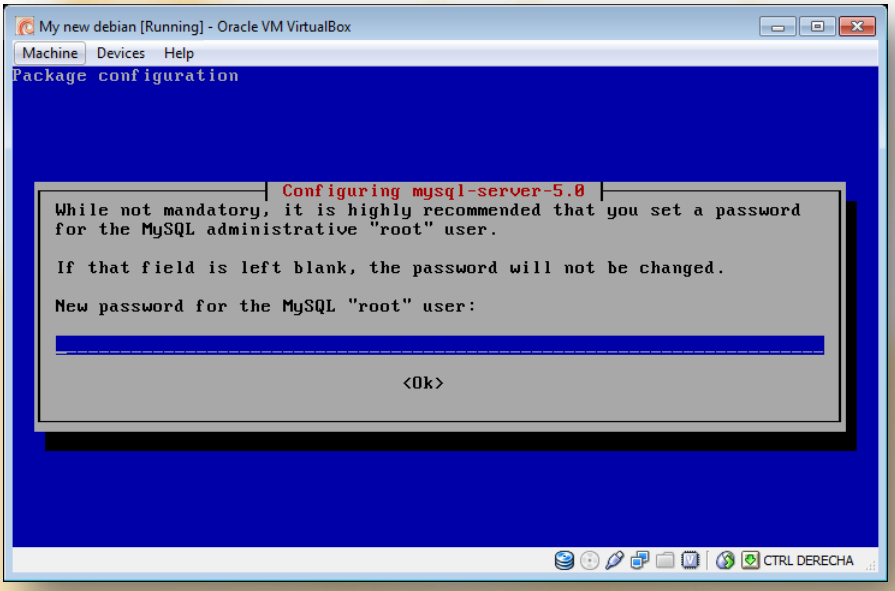

Fig. 60. MySQL root password.

And, of course, you must enter it twice (Fig. 61).

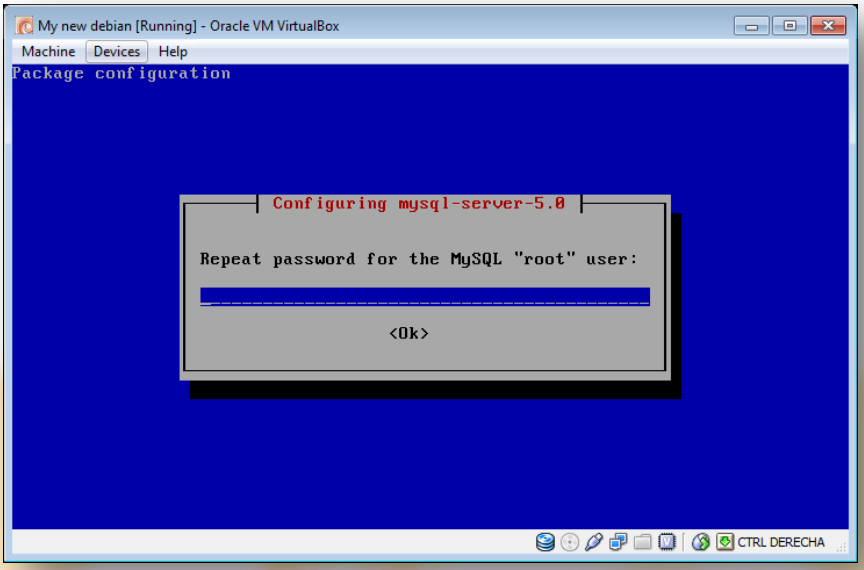

Fig. 61. MySQL root password verification.

Links

Software	URL
Debian	http://www.debian.org
MySQL	http://www.mysql.com
Pgpool-II	http://pgfoundry.org/projects/pgpool/
PostgreSQL	http://www.postgresql.org
VirtualBox	http://www.virtualbox.org

Figures Index

Fig. 1. VirtualBox download page. 4
Fig. 2. VirtualBox GUI. 4
Fig. 3. New machine menu entry. 5
Fig. 4. New Virtual Machine welcome window. 5
Fig. 5. VM Name and OS Type. 6
Fig. 6. Memory. 6
Fig. 7. Virtual Hard Disk. 7
Fig. 8. Create new virtual disk wizard welcome. 7
Fig. 9. Hard disk storage type. 8
Fig. 10. Virtual disk location and size. 8
Fig. 11. Hard disk summary. 9
Fig. 12. Machine created. 9
Fig. 13. Settings meny entry. 10
Fig. 14. Settings window. 10
Fig. 15. Motherboard settings. 11
Fig. 16. Processor settings. 12
Fig. 17. Acceleration settings. 12
Fig. 18. Storage settings. 13
Fig. 19. CD/DVD mounting. 13
Fig. 20. CD/DVD device. 14
Fig. 21. Virtual media manager. 14
Fig. 22. Network settings. 15
Fig. 23. Attached to. 16
Fig. 24. Start. 16
Fig. 25. First run wizard. 17
Fig. 26. Installation media. 17
Fig. 27. Starting. 18
Fig. 28. No bootable media. 18
Fig. 29. Debian install menu. 19
Fig. 30. Choose language. 20
Fig. 31. Region. 20
Fig. 32. Region Europe. 21
Fig. 33. Spain. 21
Fig. 34. Keyboard layout. 22
Fig. 35. Hostname. 22
Fig. 36. Domain name. 23
Fig. 37. Time zone. 23
Fig. 38. Partition disks. 24
Fig. 39. Disk selection. 24
Fig. 40. One partition. 25
Fig. 41. Finish partitioning. 25
Fig. 42. Confirm changes. 26
Fig. 43. Root password. 26

Fig. 44. Verify root password. 27
Fig. 45. Full user name. 27
Fig. 46. Username account. 28
Fig. 47. User password. 28
Fig. 48. User password verification. 29
Fig. 49. Repository country. 29
Fig. 50. Archive mirror. 30
Fig. 51. Proxy. 30
Fig. 52. Popularity-contest. 31
Fig. 53. Software selection. 31
Fig. 54. Minimal installation. 32
Fig. 55. GRUB. 32
Fig. 56. Installation complete. 33
Fig. 57. CD/DVD Devices menu entry. 33
Fig. 58. Login. 34
Fig. 59. Installing databases. 37
Fig. 60. MySQL root password. 37
Fig. 61. MySQL root password verification. 38