**INDIVIDUAL ASSIGNMENT**

**FOR**

**OPERATING SYSTEMS (**CX004-3-3-OPS**)**

**By**

**Adrien Poupa**

**TP040869**

INTAKE: UCFEFREI1603-PDK

**DUE DATE:** 1 June 2016

**NAME OF LECTURER:** MR DHASON PADMAKUMAR

*Debian logo*[1]

# Operating system chosen: Debian 8

[1] Source : Wikimedia https://upload.wikimedia.org/wikipedia/commons/thumb/4/4a/Debian-OpenLogo.svg/775px-Debian-OpenLogo.svg.png

# Table of contents

## 1. Introduction to Debian

The researcher has chosen to talk about Debian. It is not the OS the researches uses as a daily-driver, however it is used on the webserver he uses to host websites. Debian has been created by Ian Murdock in 1993, its name is coming from a contraction of “Ian” and Ian’s girlfriend “Debra”.

Debian is free and open-source, based on a Linux kernel; this distribution is one of the oldest and one of the most popular for personal computers and network servers requiring reliability. Indeed, three development branches are used: unstable for newest programs, testing for programs that have passed the unstable branch, then stable used for production, which is very reliable. It is so stable that other distributions such as Ubuntu are based on both Debian unstable and testing branches.

It has the largest software compilation with over 50,000 software packages. Debian does not require much resources: it is possible to install it with 60 MB of RAM.

To sum up, its advantages are the followings: it is free, open-source and extremely reliable. That is why it is widely used it as a webserver, and why it is so popular among other server distribution (first and 32% of Linux market share for web servers according to W3Techs).

However, it has several drawbacks such as very slow stable release cycle (stable software are frequently deprecated when they hit the stable branch). Plus, it has a very strict policy concerning proprietary software, leading to problems with codecs for example. Moreover, it had issues with the open source community as well, since the Mozilla foundation did not want Debian to alter its software keeping Mozilla’s names.

## 2. Installation and configuration

### a. Installation of the Operating System chosen

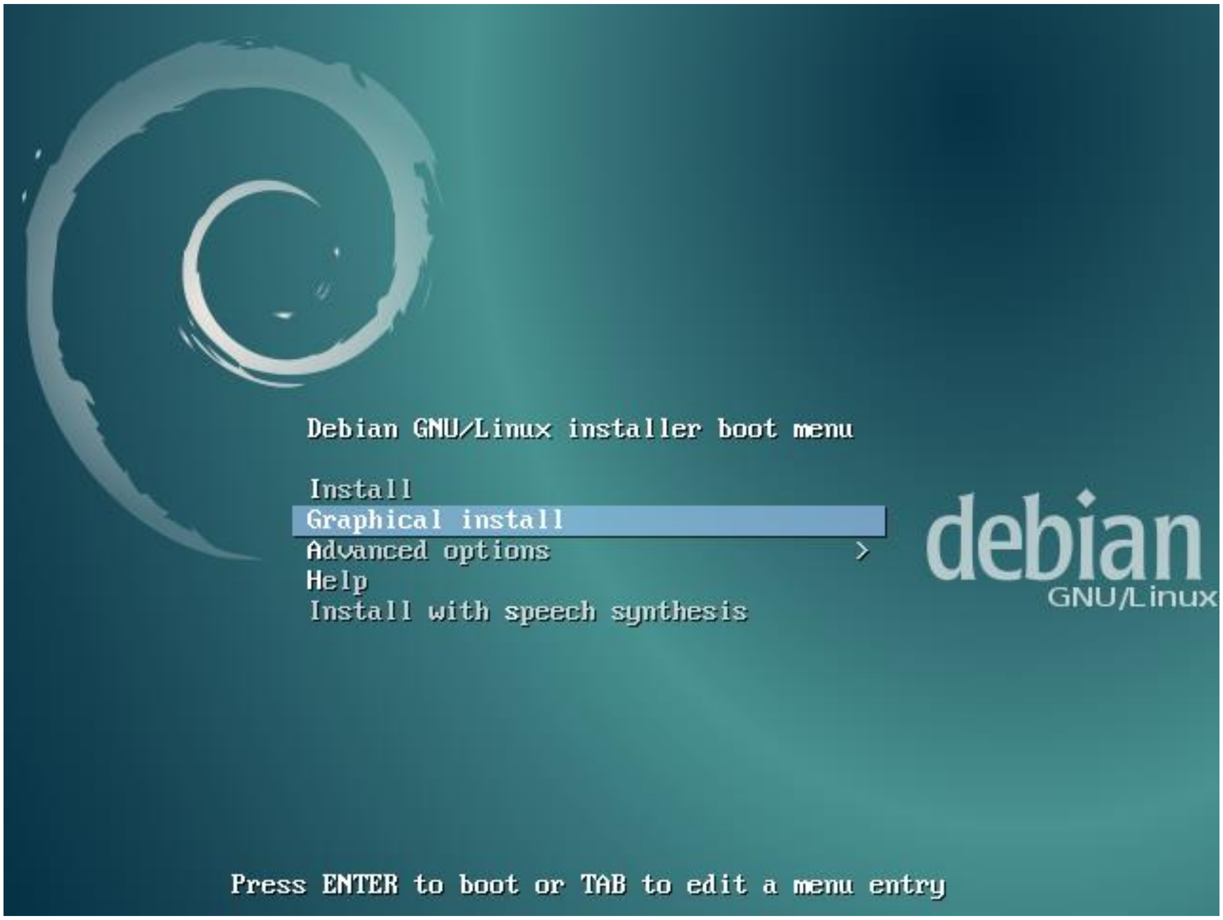
In this tutorial, we will install Debian 8.4.0 using an ISO image. Other means of installation are available, such as live CDs, USB keys or minimal CDs when having an internet connection.

One need to ensure that the CPU architecture on the computer where Debian will be installed matches the architecture of the ISO file. AMD64 should be good enough for modern computers.

First, download the ISO image from the following URL:

http://caesar.acc.umu.se/debian-cd/8.4.0/amd64/iso-cd/debian-8.4.0-amd64-CD-1.iso

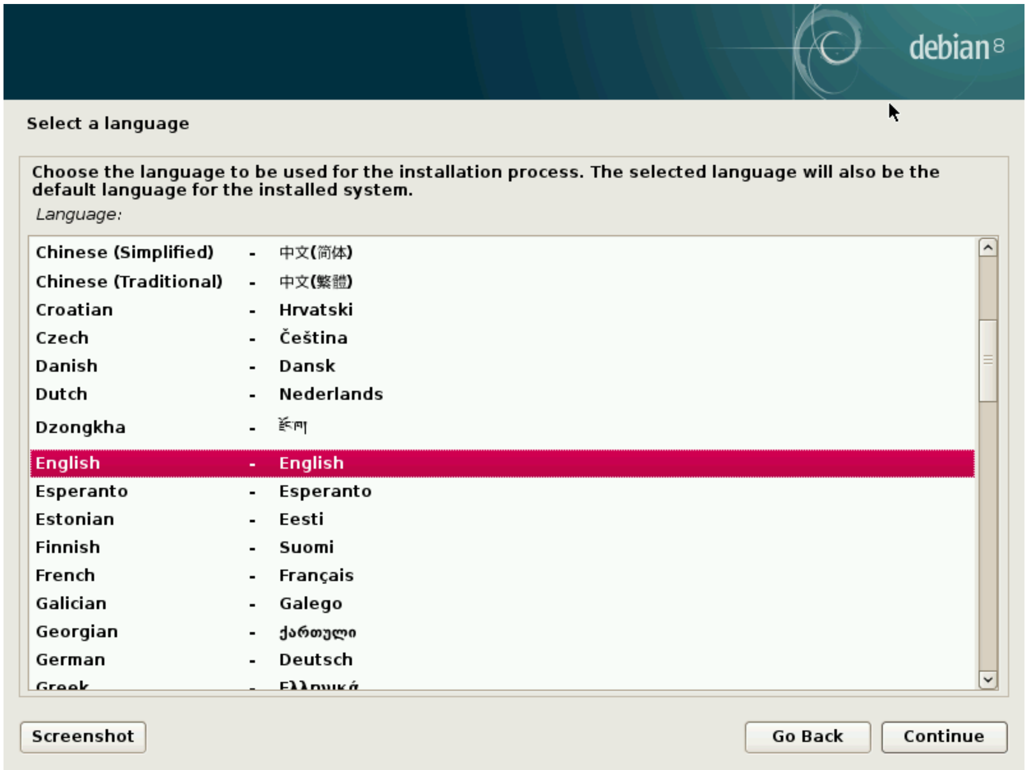
Then we boot into the CD where the ISO has been burnt:


Debian GNU/Linux installer boot menu
Install
Graphical install
Advanced options >
Help
Install with speech synthesis
debian
GNU/Linux

Press ENTER to boot or TAB to edit a menu entry

We select “Graphical install” in order to have a GUI interface during the installation process.

Then, select the language to be installed.



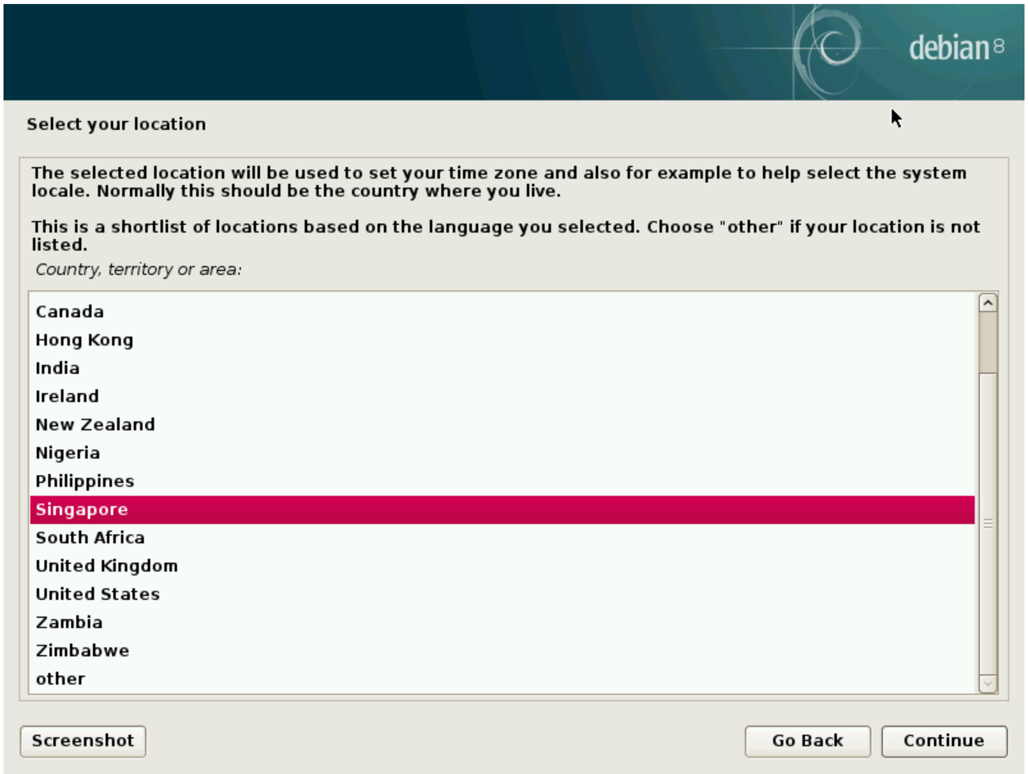
Select a language

Choose the language to be used for the installation process. The selected language will also be the default language for the installed system.

*Language:*

| | | |
|---|---|---|
| Chinese (Simplified) | - | 中文(简体) |
| Chinese (Traditional) | - | 中文(繁體) |
| Croatian | - | Hrvatski |
| Czech | - | Čeština |
| Danish | - | Dansk |
| Dutch | - | Nederlands |
| Dzongkha | - | རྫོང་ཁ |
| English | - | English |
| Esperanto | - | Esperanto |
| Estonian | - | Eesti |
| Finnish | - | Suomi |
| French | - | Français |
| Galician | - | Galego |
| Georgian | - | ქართული |
| German | - | Deutsch |
| Greek | - | Ελληνικά |

Then, select the location of the computer for the clock.



Select your location

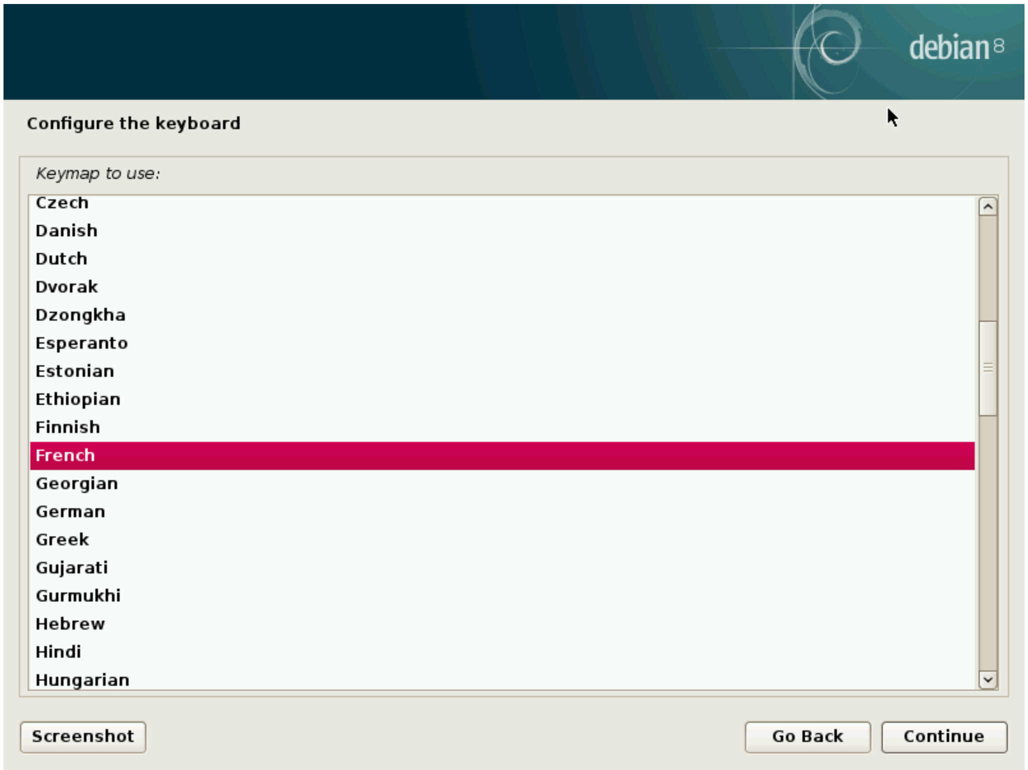
The selected location will be used to set your time zone and also for example to help select the system locale. Normally this should be the country where you live.

This is a shortlist of locations based on the language you selected. Choose "other" if your location is not listed.

*Country, territory or area:*

Canada
Hong Kong
India
Ireland
New Zealand
Nigeria
Philippines
Singapore
South Africa
United Kingdom
United States
Zambia
Zimbabwe
other

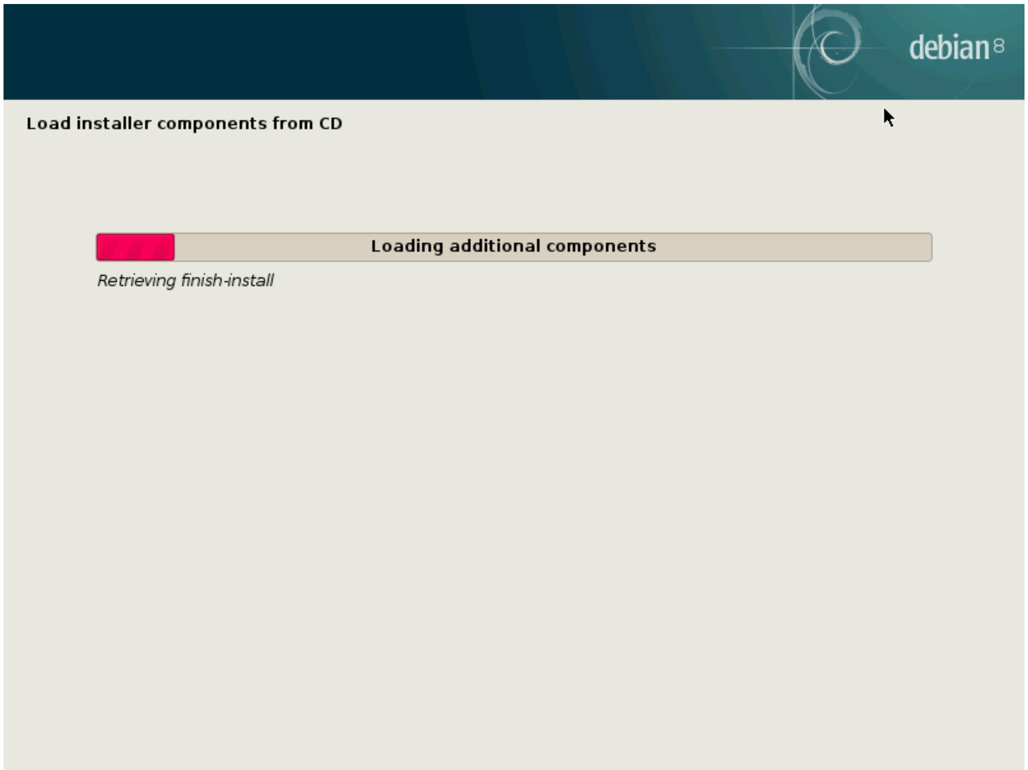
Select the keyboard mapping to use:



Configure the keyboard

*Keymap to use:*

- Czech
- Danish
- Dutch
- Dvorak
- Dzongkha
- Esperanto
- Estonian
- Ethiopian
- Finnish
- French
- Georgian
- German
- Greek
- Gujarati
- Gurmukhi
- Hebrew
- Hindi
- Hungarian

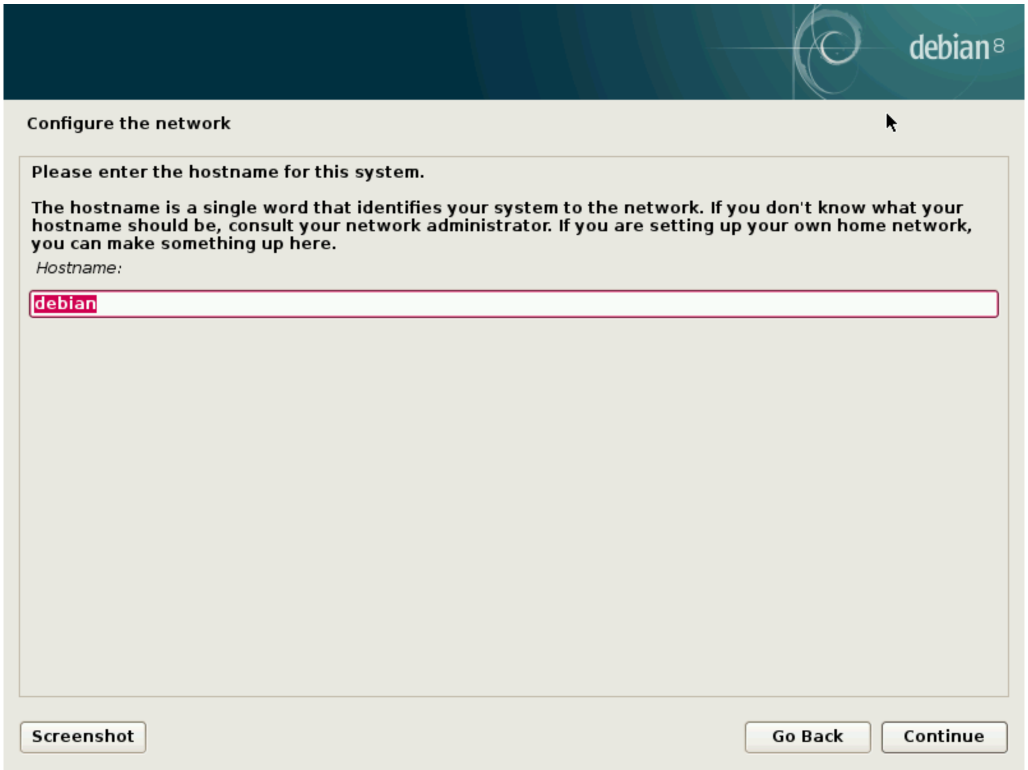


Load installer components from CD

Loading additional components

*Retrieving finish-install*

Enter the hostname. You can leave "debian" as default.



Configure the network

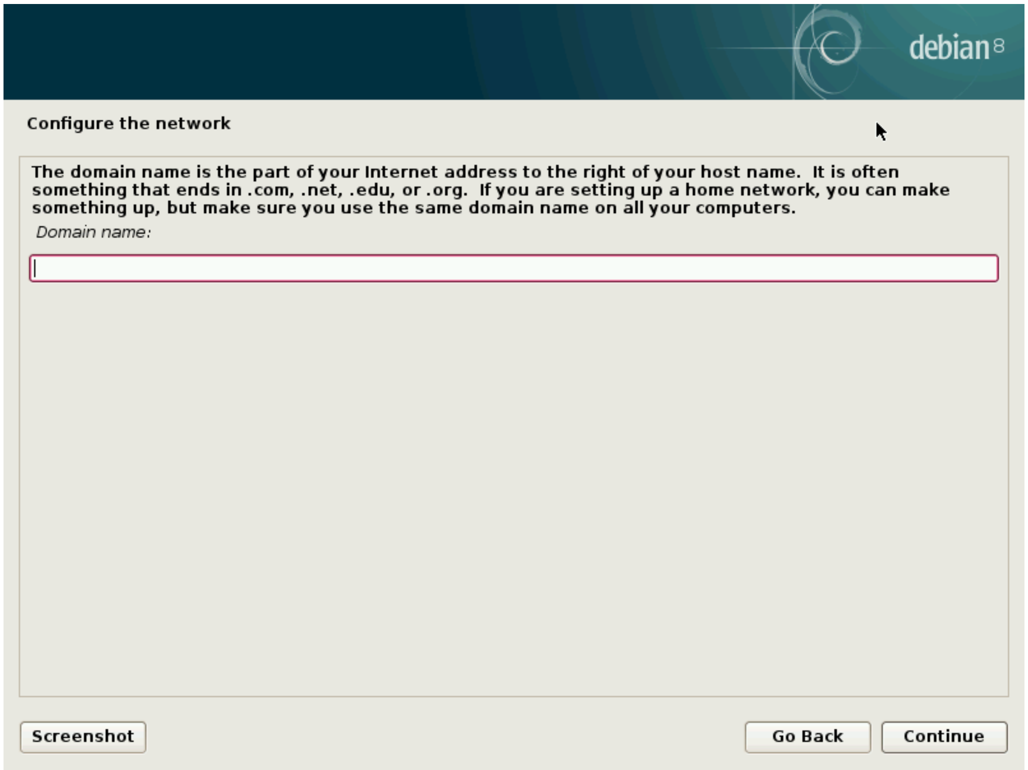
Please enter the hostname for this system.

The hostname is a single word that identifies your system to the network. If you don't know what your hostname should be, consult your network administrator. If you are setting up your own home network, you can make something up here.

Hostname:

debian

Enter your domain name – you can leave empty.

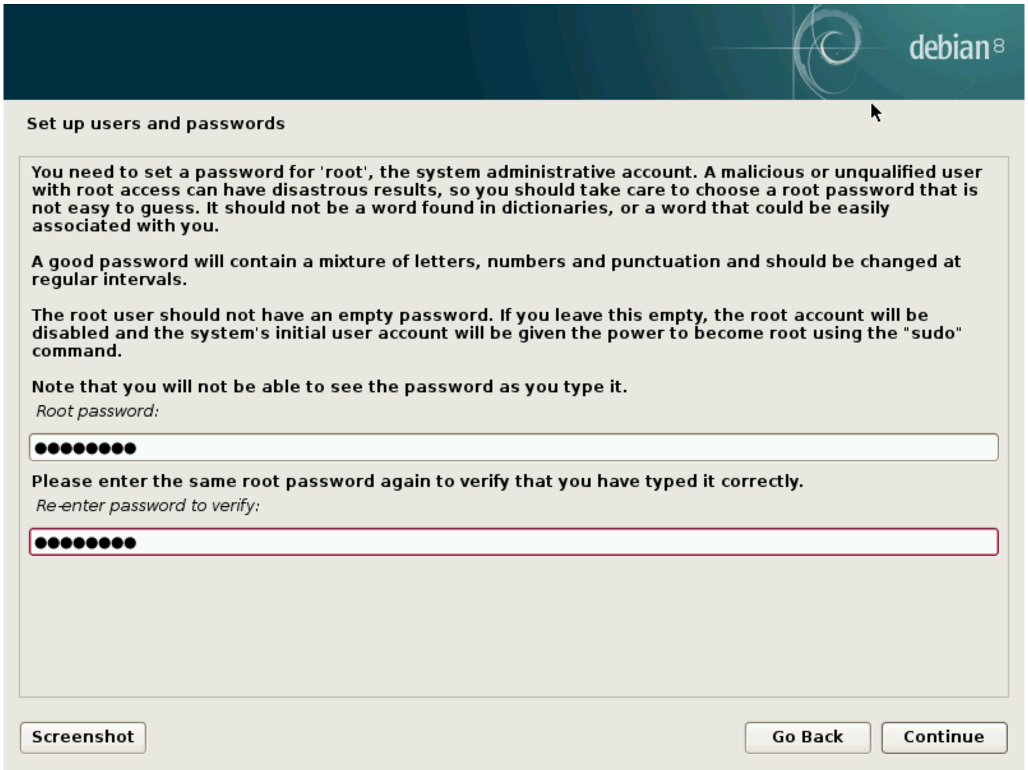


Configure the network

The domain name is the part of your Internet address to the right of your host name. It is often something that ends in .com, .net, .edu, or .org. If you are setting up a home network, you can make something up, but make sure you use the same domain name on all your computers.

Domain name:

Enter the root password twice, very important to administrate the computer.



Set up users and passwords

You need to set a password for 'root', the system administrative account. A malicious or unqualified user with root access can have disastrous results, so you should take care to choose a root password that is not easy to guess. It should not be a word found in dictionaries, or a word that could be easily associated with you.

A good password will contain a mixture of letters, numbers and punctuation and should be changed at regular intervals.

The root user should not have an empty password. If you leave this empty, the root account will be disabled and the system's initial user account will be given the power to become root using the "sudo" command.

Note that you will not be able to see the password as you type it.

*Root password:*

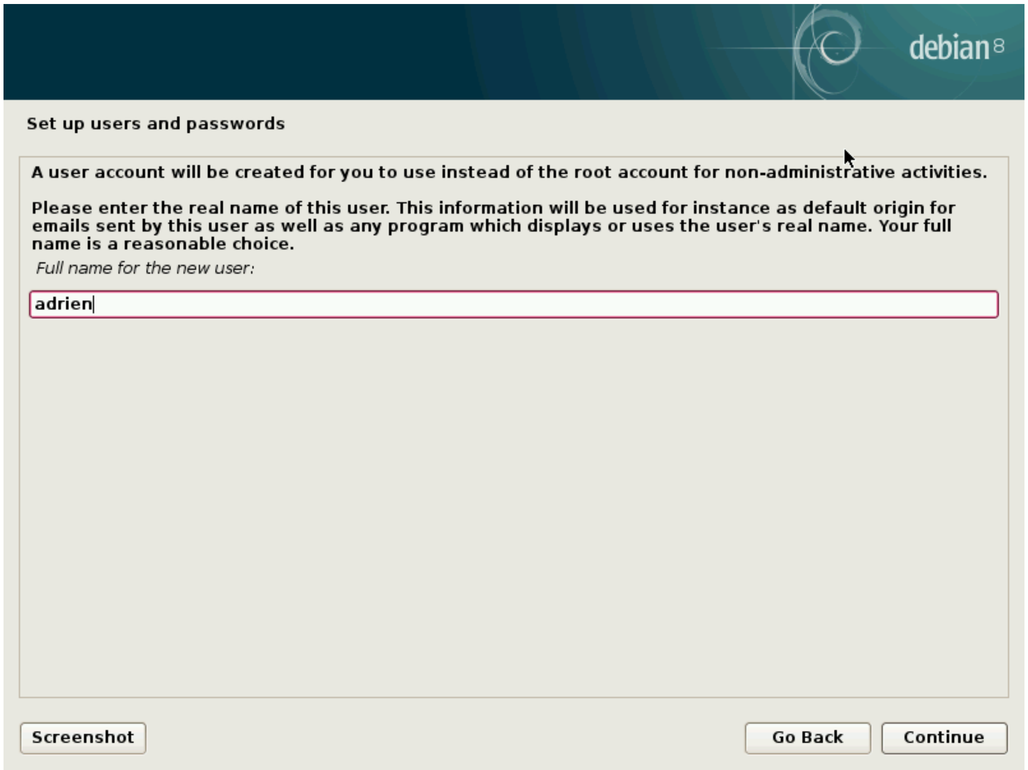
●●●●●●●●

Please enter the same root password again to verify that you have typed it correctly.

*Re-enter password to verify:*

●●●●●●●●

Enter your username. You can create other accounts once the installation is finished:



Set up users and passwords

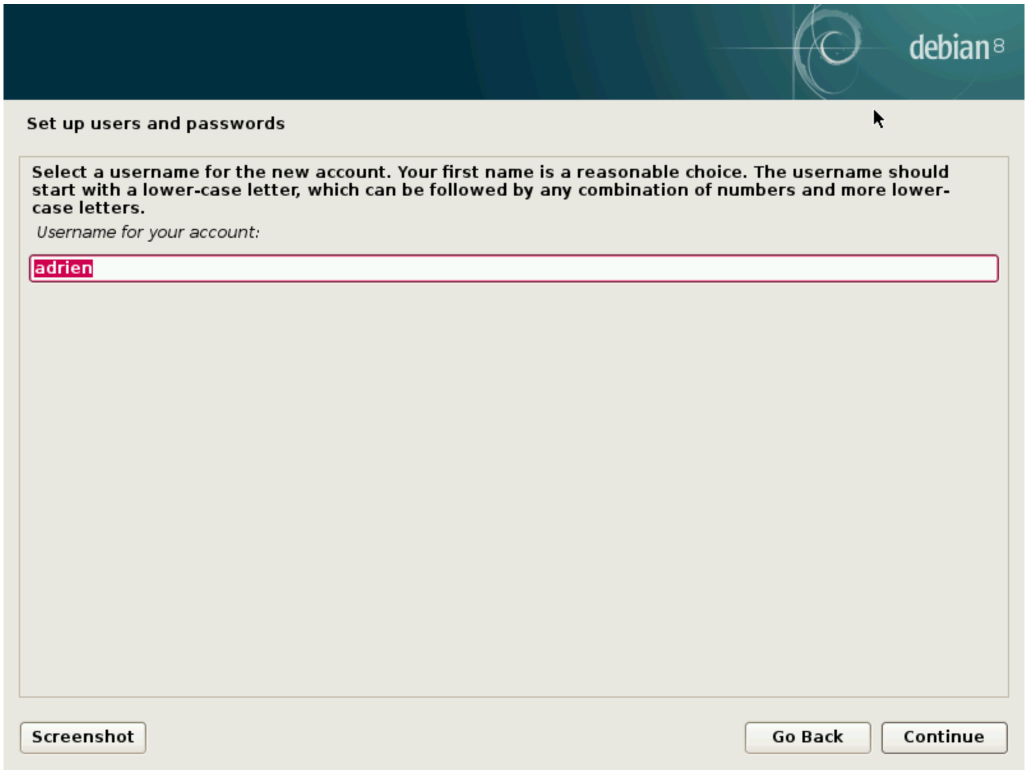
A user account will be created for you to use instead of the root account for non-administrative activities.

Please enter the real name of this user. This information will be used for instance as default origin for emails sent by this user as well as any program which displays or uses the user's real name. Your full name is a reasonable choice.

*Full name for the new user:*

adrien

Enter your username once again.



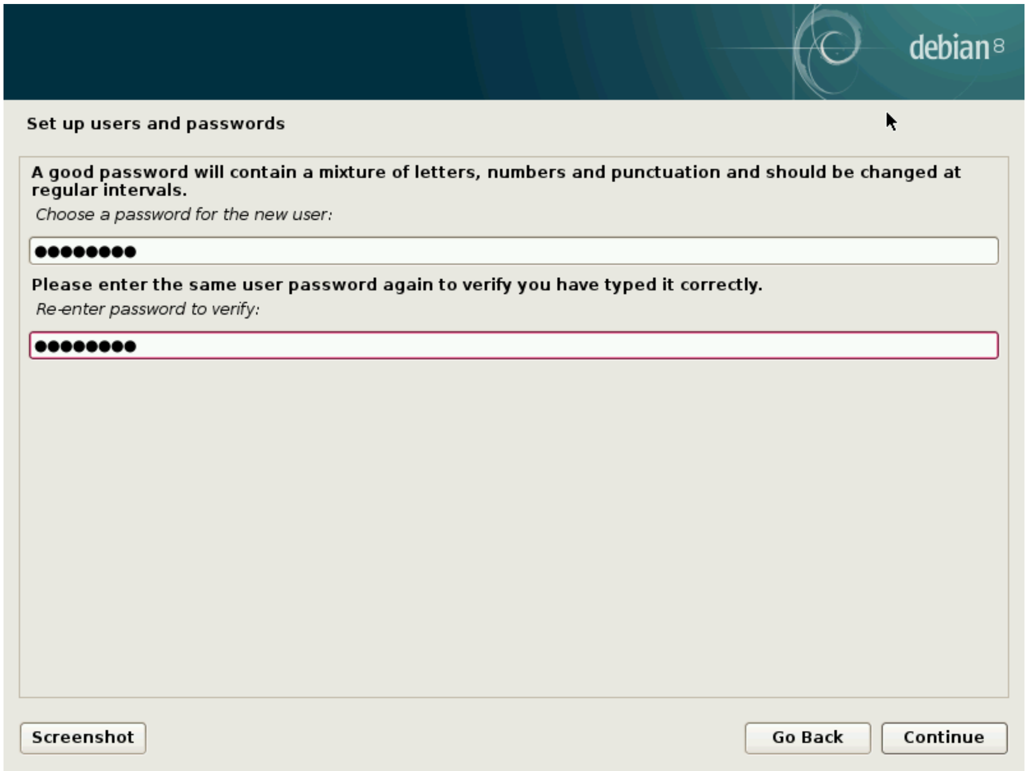
Set up users and passwords

Select a username for the new account. Your first name is a reasonable choice. The username should start with a lower-case letter, which can be followed by any combination of numbers and more lower-case letters.

*Username for your account:*

adrien

Enter the password for the user you just created.



Set up users and passwords

A good password will contain a mixture of letters, numbers and punctuation and should be changed at regular intervals.

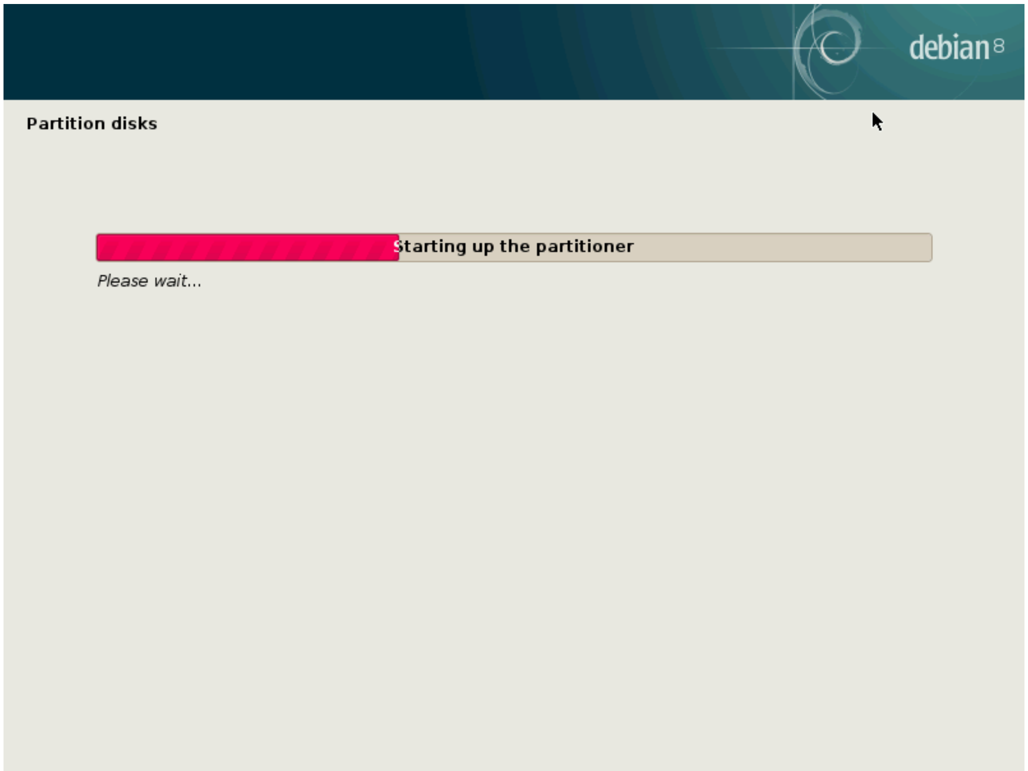
*Choose a password for the new user:*

●●●●●●●●

Please enter the same user password again to verify you have typed it correctly.

*Re-enter password to verify:*

●●●●●●●●

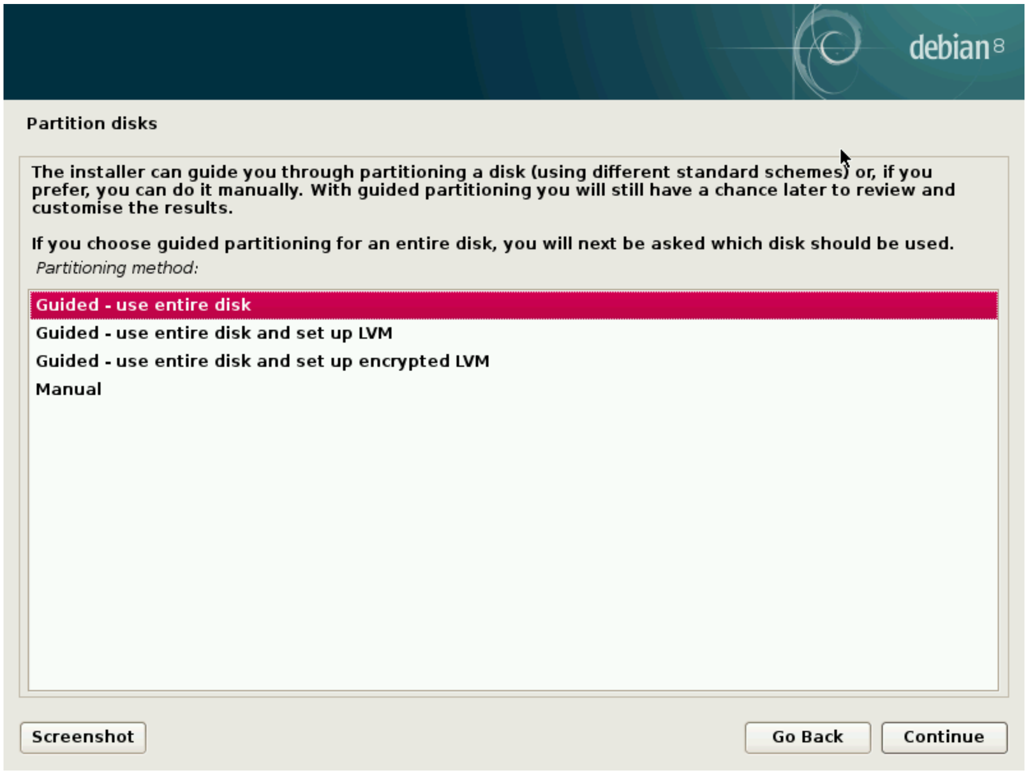


Partition disks

Starting up the partitioner

Please wait...

Now select the partitioning you want. The simplest is to install the system on the entire disk.



Partition disks

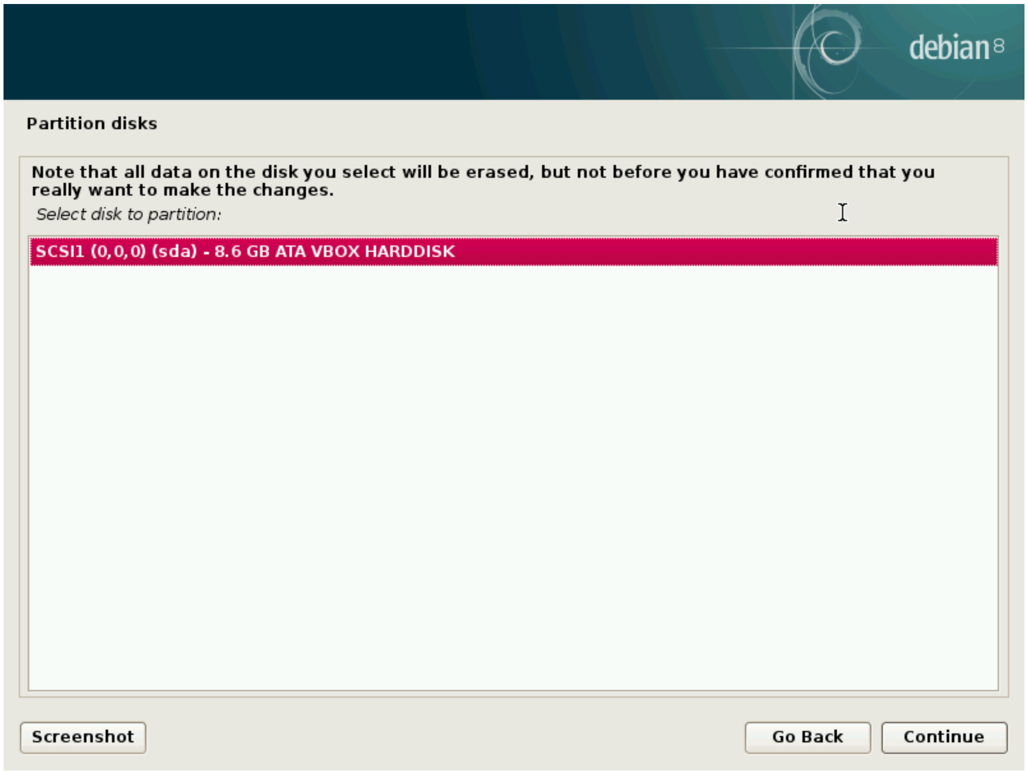
The installer can guide you through partitioning a disk (using different standard schemes) or, if you prefer, you can do it manually. With guided partitioning you will still have a chance later to review and customise the results.

If you choose guided partitioning for an entire disk, you will next be asked which disk should be used.

*Partitioning method:*

- Guided - use entire disk
- Guided - use entire disk and set up LVM
- Guided - use entire disk and set up encrypted LVM
- Manual

Select the disk you want to use.



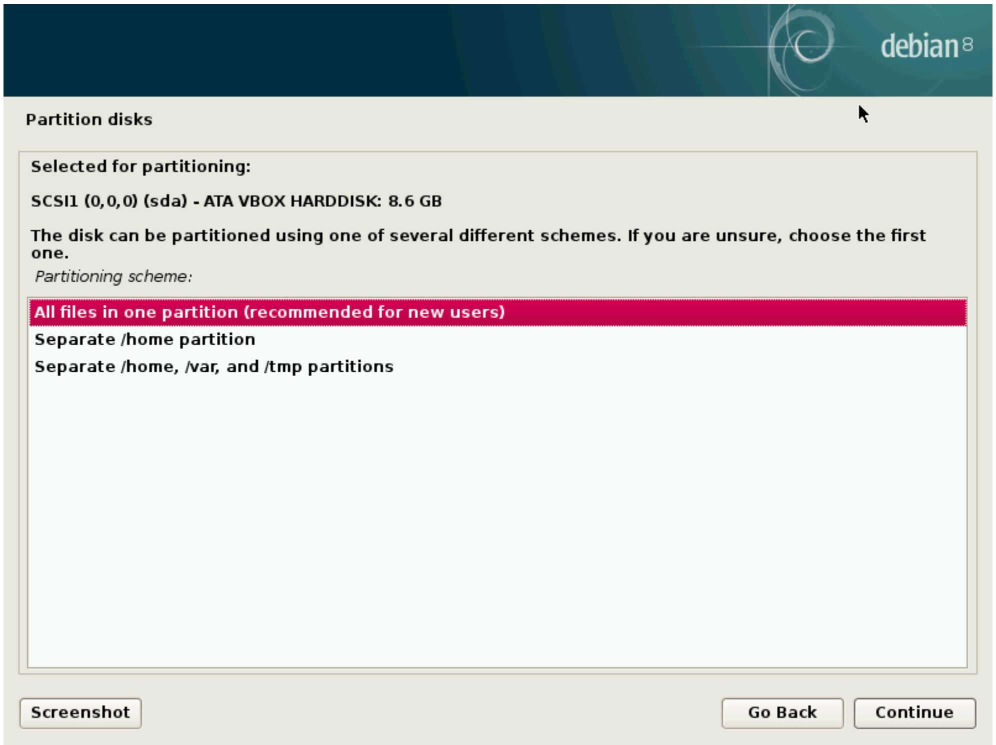
Partition disks

Note that all data on the disk you select will be erased, but not before you have confirmed that you really want to make the changes.

*Select disk to partition:*

SCSI1 (0,0,0) (sda) - 8.6 GB ATA VBOX HARDDISK

Choose if you want to use multiple partitions. The simplest is to put all the files in one single partition.



Partition disks

Selected for partitioning:

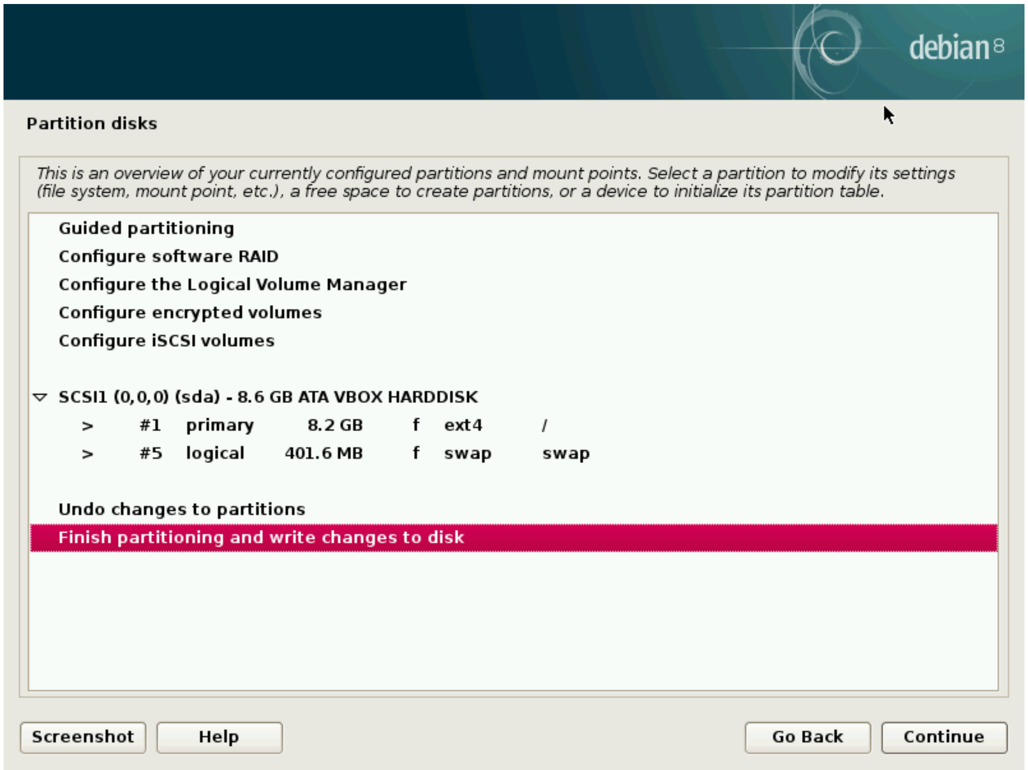
SCSI1 (0,0,0) (sda) - ATA VBOX HARDDISK: 8.6 GB

The disk can be partitioned using one of several different schemes. If you are unsure, choose the first one.

*Partitioning scheme:*

All files in one partition (recommended for new users)
Separate /home partition
Separate /home, /var, and /tmp partitions

Select “Finish portioning…” and hit “Continue”.

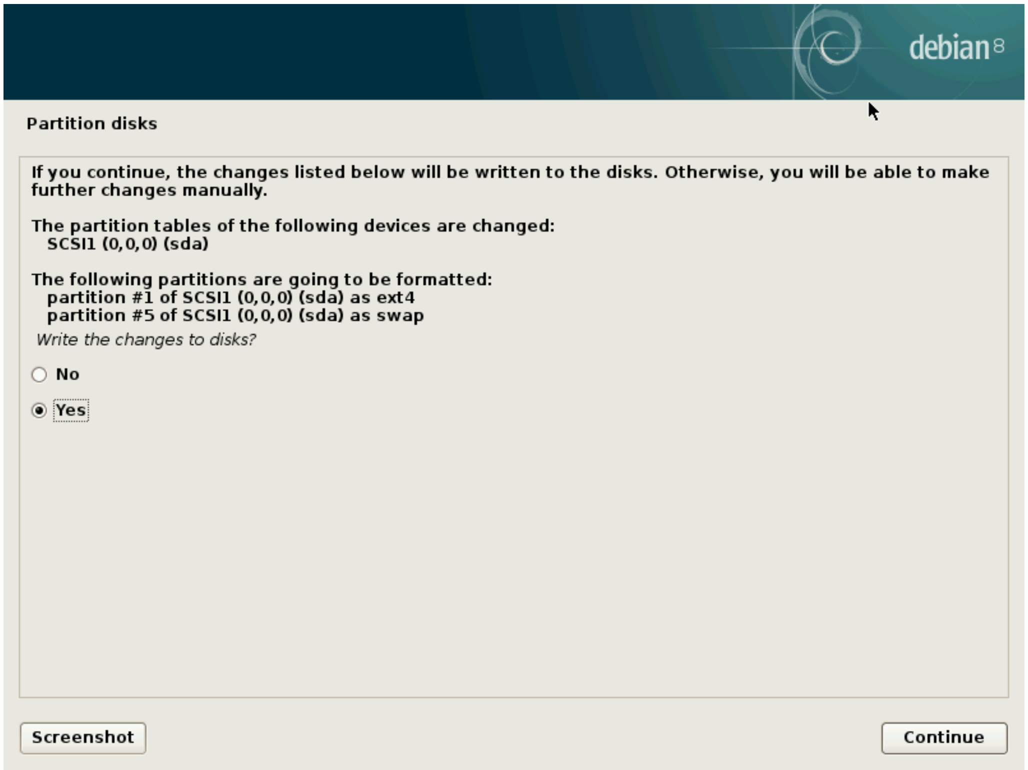


Partition disks

*This is an overview of your currently configured partitions and mount points. Select a partition to modify its settings (file system, mount point, etc.), a free space to create partitions, or a device to initialize its partition table.*

```
Guided partitioning
Configure software RAID
Configure the Logical Volume Manager
Configure encrypted volumes
Configure iSCSI volumes

SCSI1 (0,0,0) (sda) - 8.6 GB ATA VBOX HARDDISK
    >    #1   primary      8.2 GB    f   ext4     /
    >    #5   logical    401.6 MB    f   swap     swap

Undo changes to partitions
Finish partitioning and write changes to disk
```

Select “Yes” to write the changes to disks.



Partition disks

If you continue, the changes listed below will be written to the disks. Otherwise, you will be able to make further changes manually.

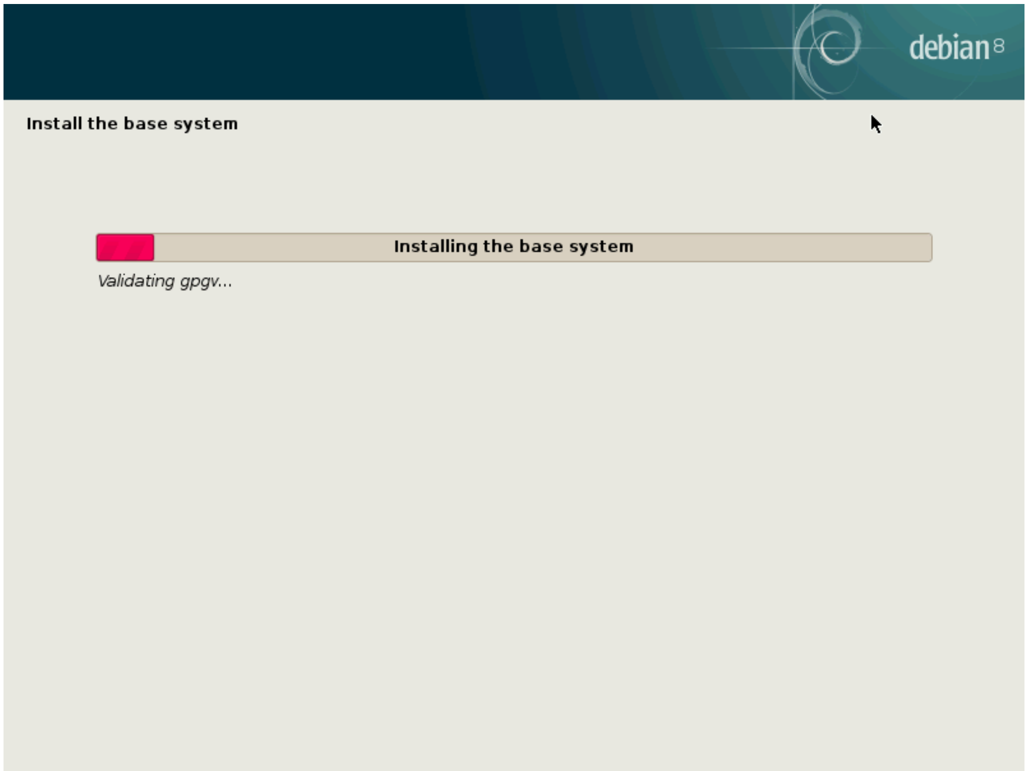
The partition tables of the following devices are changed:
SCSI1 (0,0,0) (sda)

The following partitions are going to be formatted:
partition #1 of SCSI1 (0,0,0) (sda) as ext4
partition #5 of SCSI1 (0,0,0) (sda) as swap

*Write the changes to disks?*

○ No

● Yes

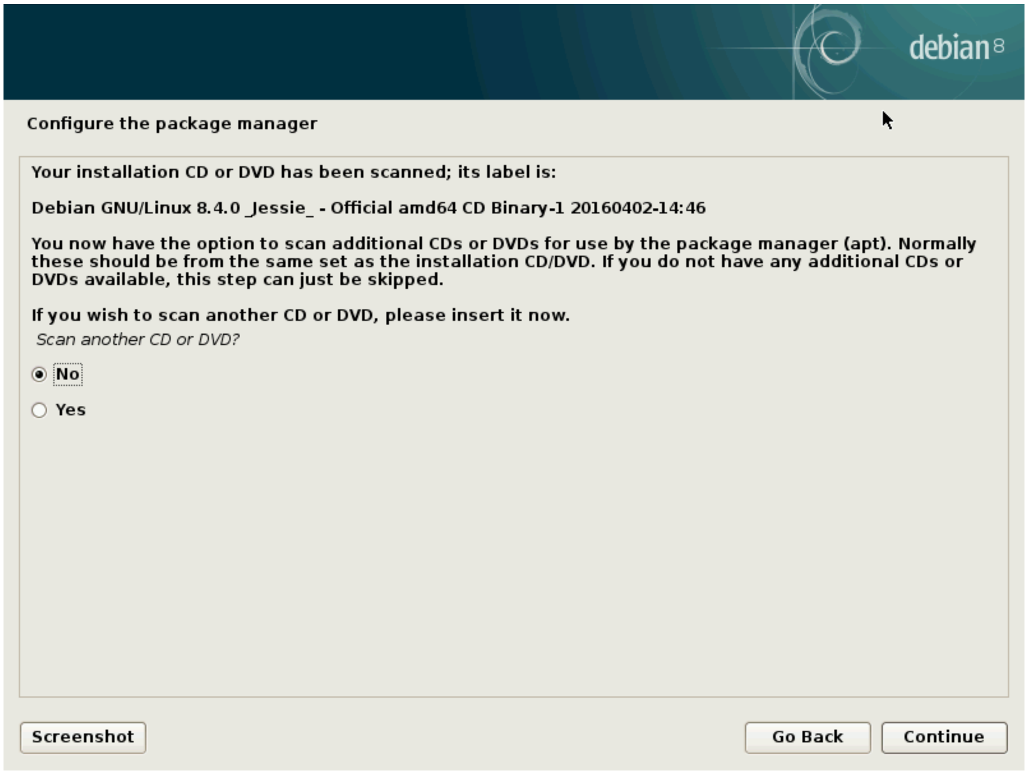


Install the base system

Installing the base system

*Validating gpgv...*

Select “No” as we do not have more CDs to install.



Configure the package manager

Your installation CD or DVD has been scanned; its label is:

Debian GNU/Linux 8.4.0 _Jessie_ - Official amd64 CD Binary-1 20160402-14:46

You now have the option to scan additional CDs or DVDs for use by the package manager (apt). Normally these should be from the same set as the installation CD/DVD. If you do not have any additional CDs or DVDs available, this step can just be skipped.

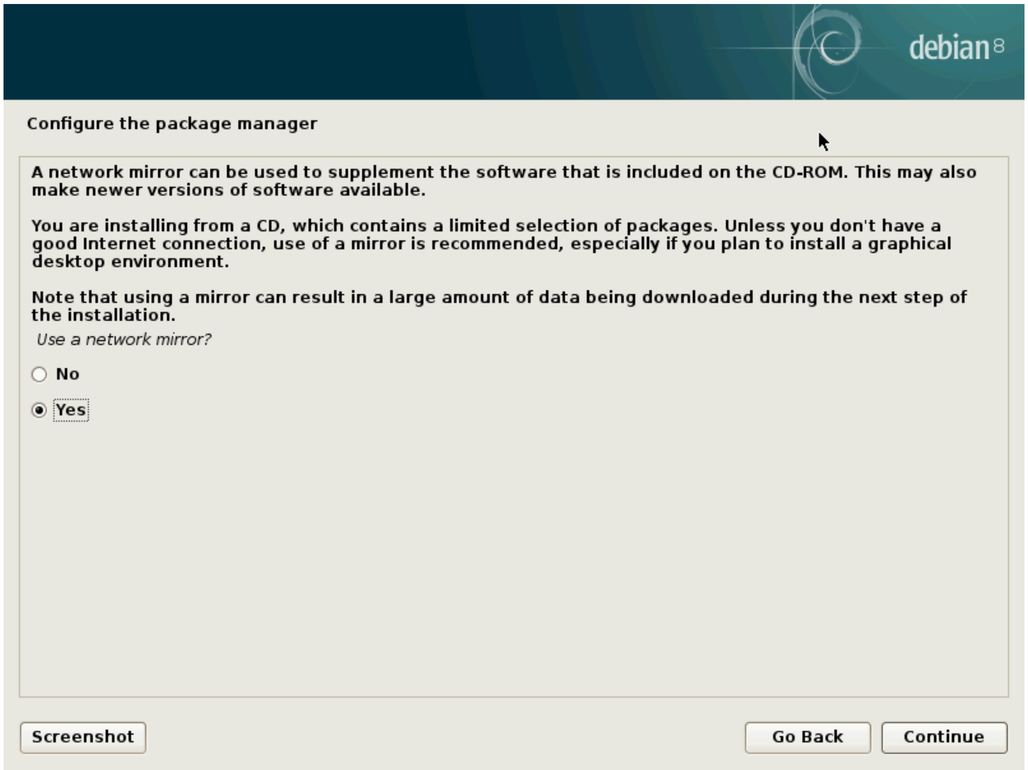
If you wish to scan another CD or DVD, please insert it now.

*Scan another CD or DVD?*

● No

○ Yes

Select “yes” to use a network mirror to install additional packages such as the GUI Gnome.



Configure the package manager

A network mirror can be used to supplement the software that is included on the CD-ROM. This may also make newer versions of software available.

You are installing from a CD, which contains a limited selection of packages. Unless you don't have a good Internet connection, use of a mirror is recommended, especially if you plan to install a graphical desktop environment.

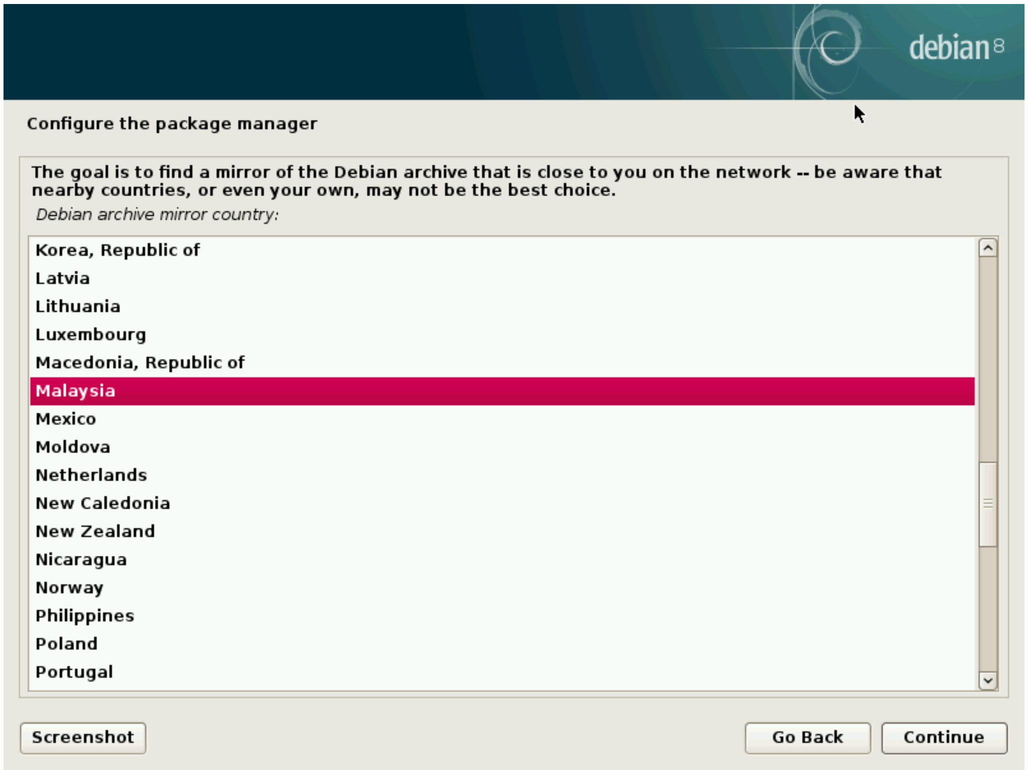
Note that using a mirror can result in a large amount of data being downloaded during the next step of the installation.

*Use a network mirror?*

○ No

● Yes

Select your country in order to have the select the best mirror.



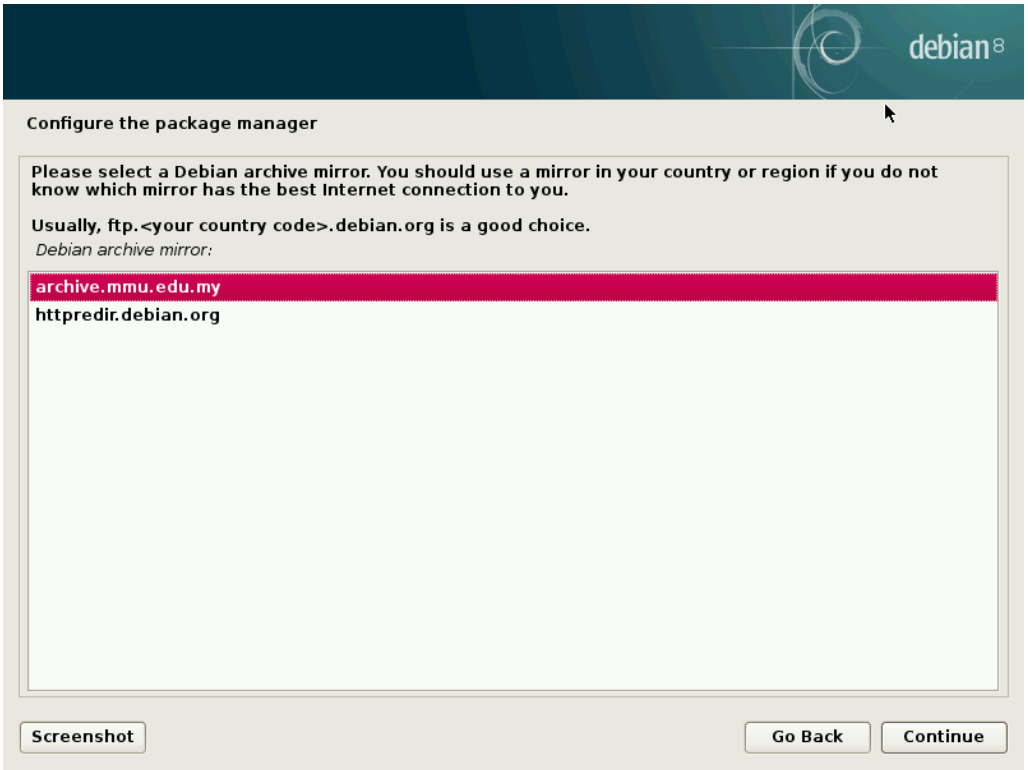
Configure the package manager

The goal is to find a mirror of the Debian archive that is close to you on the network -- be aware that nearby countries, or even your own, may not be the best choice.

*Debian archive mirror country:*

- Korea, Republic of
- Latvia
- Lithuania
- Luxembourg
- Macedonia, Republic of
- **Malaysia**
- Mexico
- Moldova
- Netherlands
- New Caledonia
- New Zealand
- Nicaragua
- Norway
- Philippines
- Poland
- Portugal

Select the appropriate mirror.



Configure the package manager

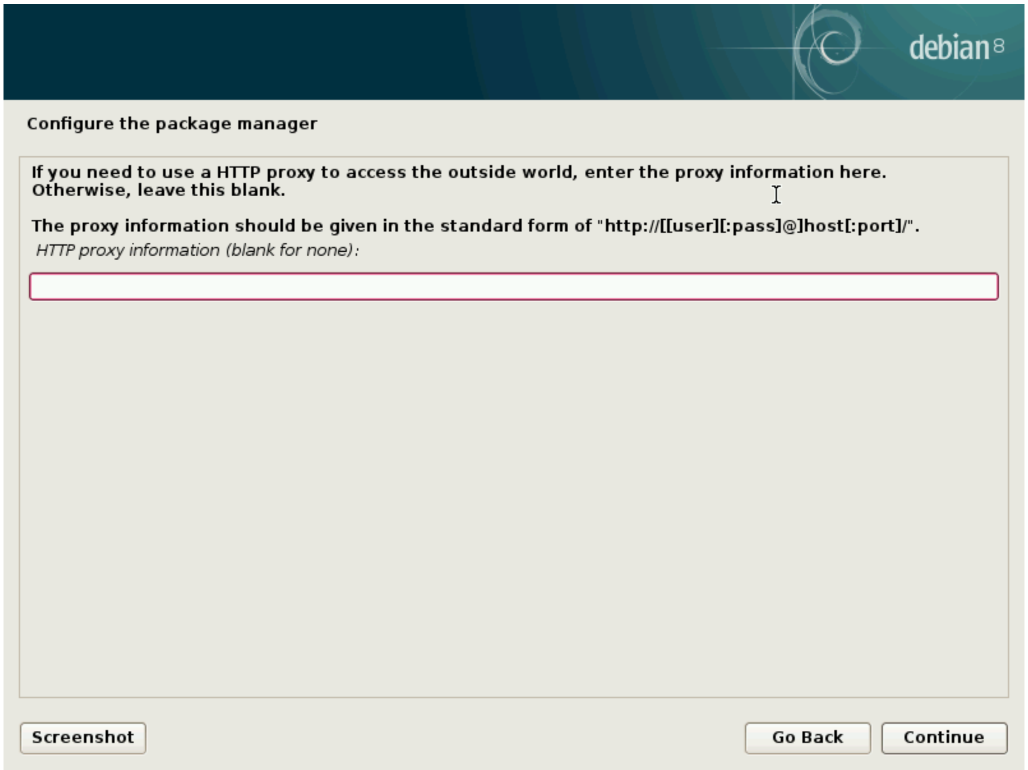
Please select a Debian archive mirror. You should use a mirror in your country or region if you do not know which mirror has the best Internet connection to you.

Usually, ftp.<your country code>.debian.org is a good choice.

*Debian archive mirror:*

- archive.mmu.edu.my
- httpredir.debian.org

Leave blank if you do not use a proxy.

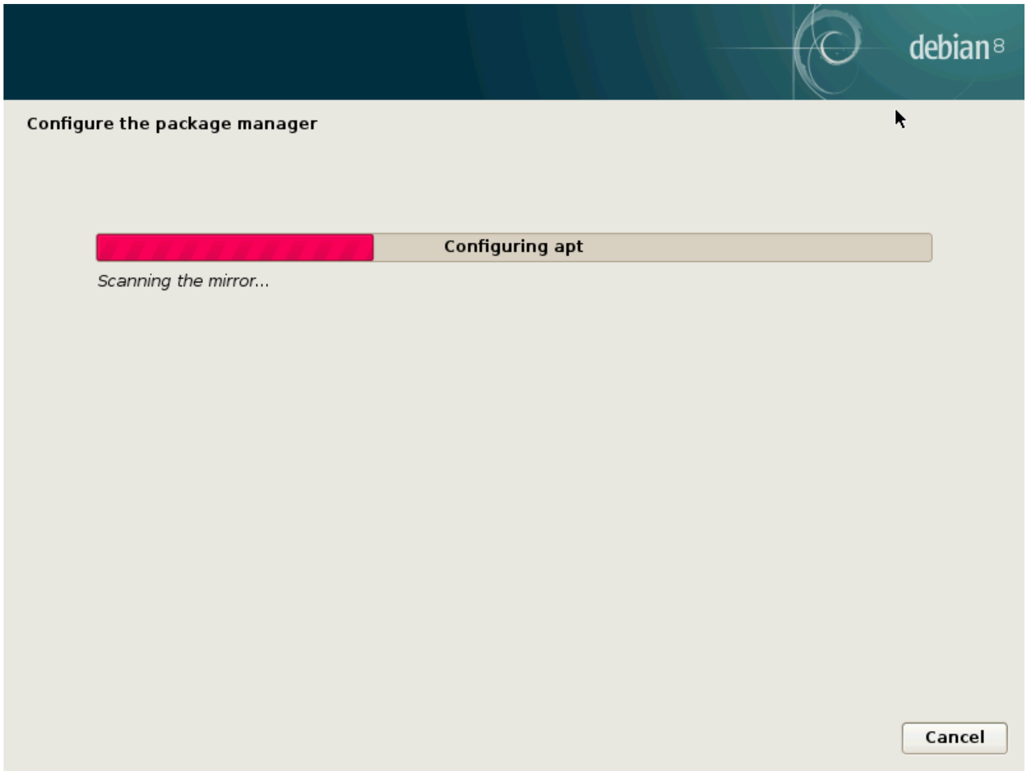


Configure the package manager

If you need to use a HTTP proxy to access the outside world, enter the proxy information here. Otherwise, leave this blank.

The proxy information should be given in the standard form of "http://[[user][:pass]@]host[:port]/".

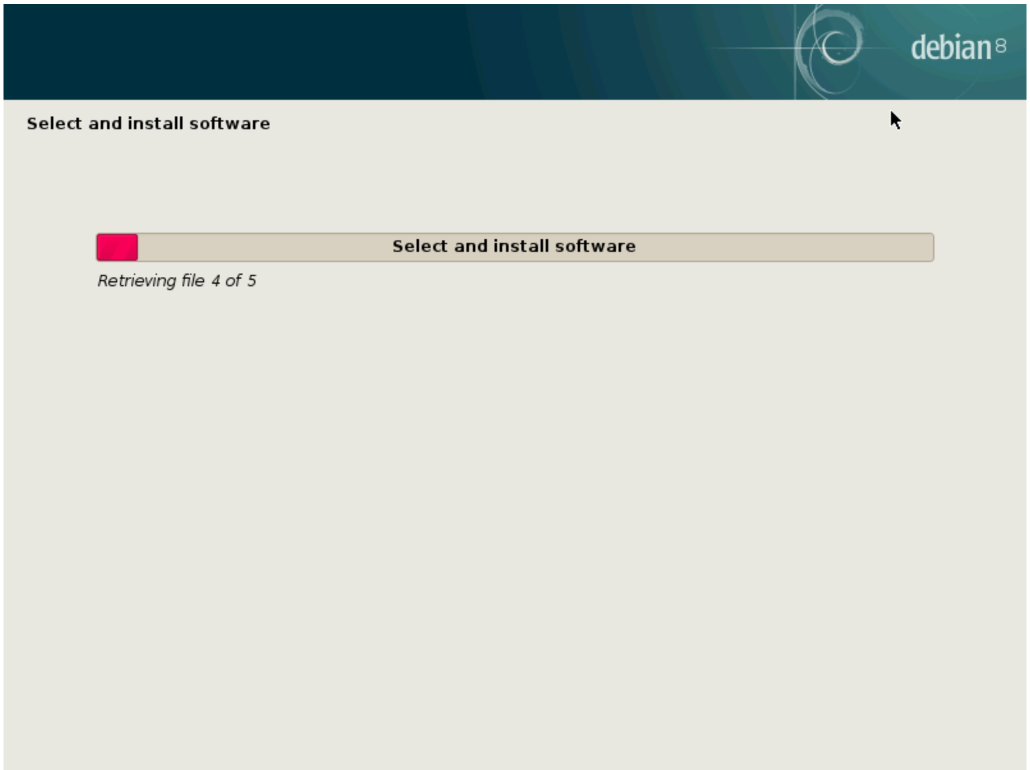
*HTTP proxy information (blank for none):*



Configure the package manager

Configuring apt

Scanning the mirror...

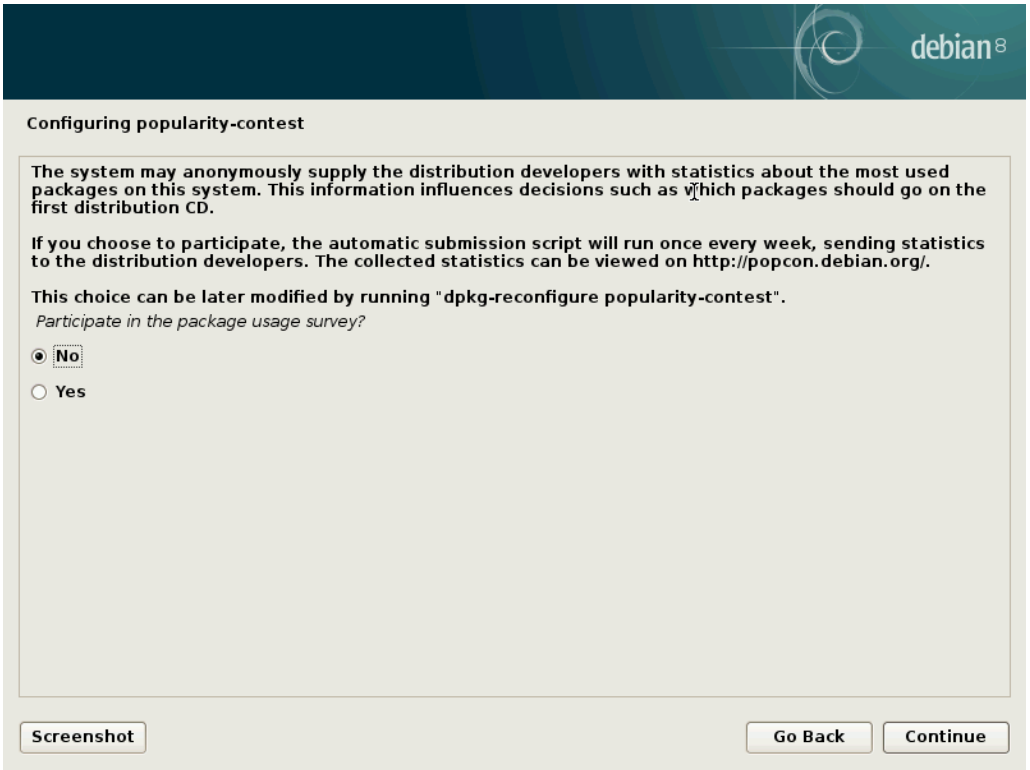


Select and install software

Select and install software

Retrieving file 4 of 5

Select “Yes” or “No” depending if you want to be part of the popularity contest.



Configuring popularity-contest

The system may anonymously supply the distribution developers with statistics about the most used packages on this system. This information influences decisions such as which packages should go on the first distribution CD.

If you choose to participate, the automatic submission script will run once every week, sending statistics to the distribution developers. The collected statistics can be viewed on http://popcon.debian.org/.

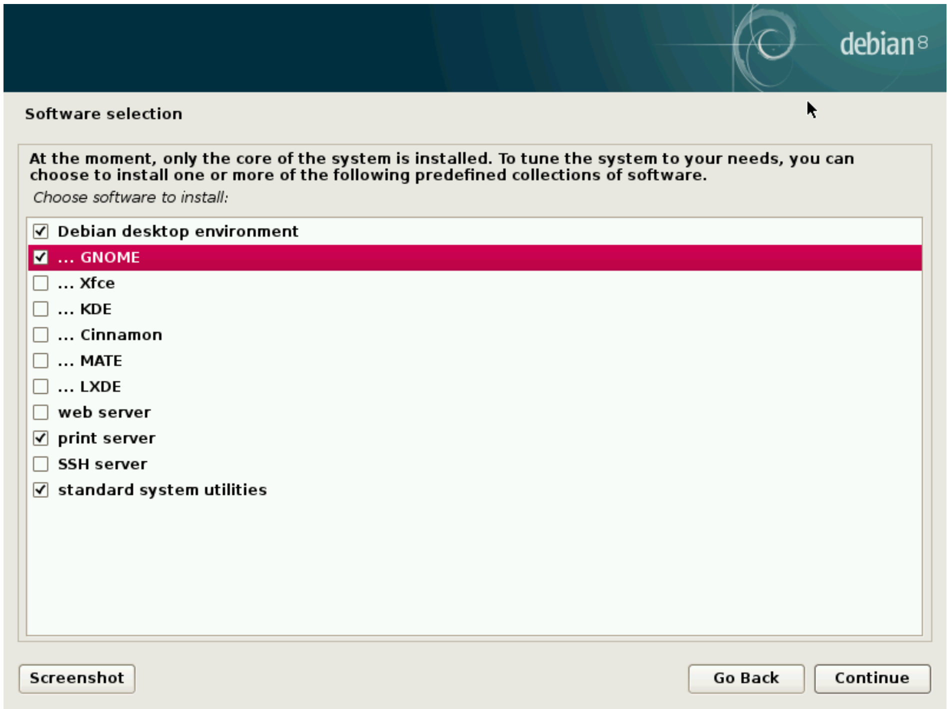
This choice can be later modified by running "dpkg-reconfigure popularity-contest".

*Participate in the package usage survey?*

● No

○ Yes

Check the “GNOME” box to install GNOME GUI. Otherwise, Debian will be command-line based.

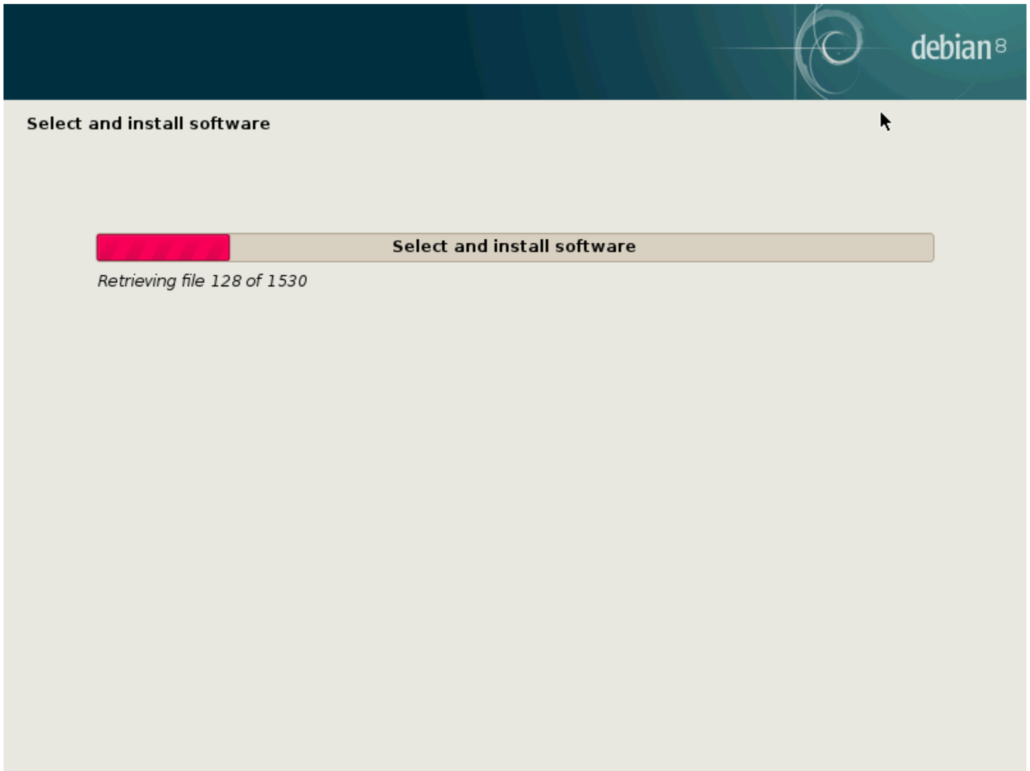


Software selection

At the moment, only the core of the system is installed. To tune the system to your needs, you can choose to install one or more of the following predefined collections of software.

*Choose software to install:*

☑ Debian desktop environment
☑ ... GNOME
□ ... Xfce
□ ... KDE
□ ... Cinnamon
□ ... MATE
□ ... LXDE
□ web server
☑ print server
□ SSH server
☑ standard system utilities

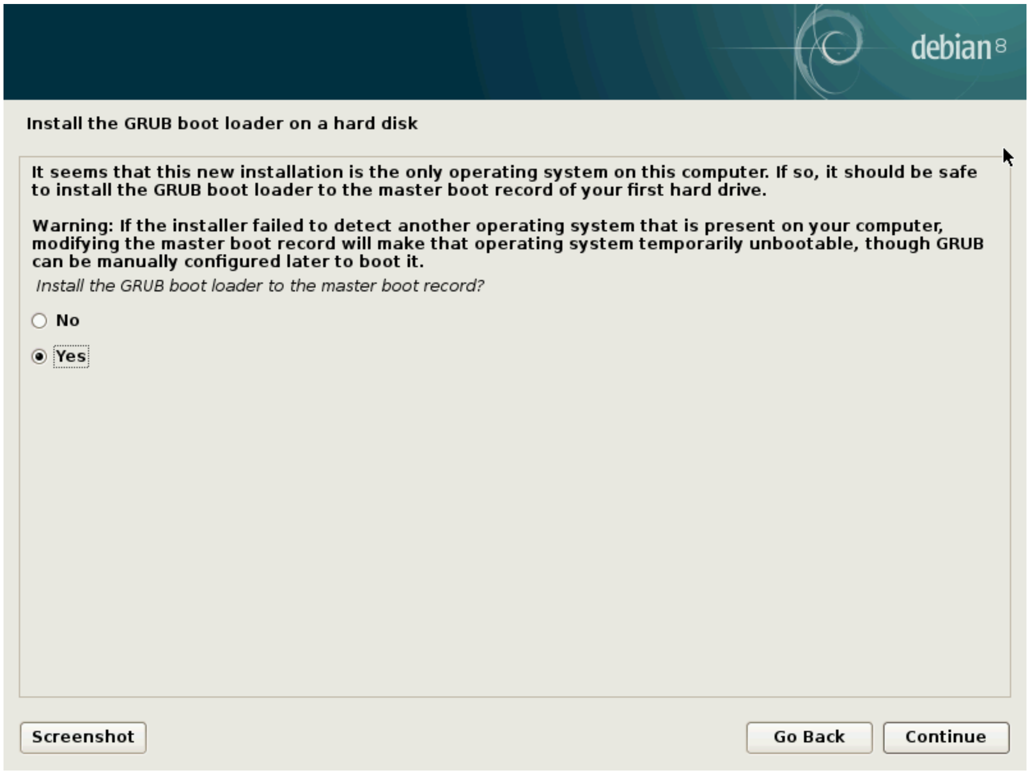


Select and install software

Select and install software

Retrieving file 128 of 1530

Select “Yes” to install the GRUB bootloader.



Install the GRUB boot loader on a hard disk

It seems that this new installation is the only operating system on this computer. If so, it should be safe to install the GRUB boot loader to the master boot record of your first hard drive.

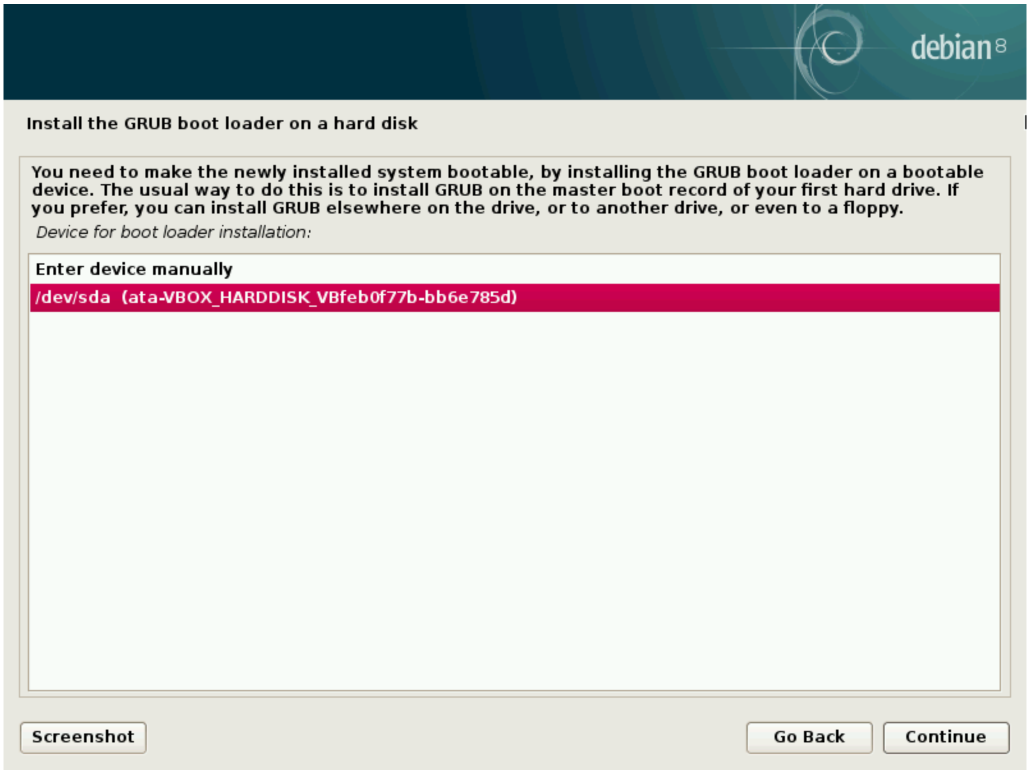
Warning: If the installer failed to detect another operating system that is present on your computer, modifying the master boot record will make that operating system temporarily unbootable, though GRUB can be manually configured later to boot it.

*Install the GRUB boot loader to the master boot record?*

○ No

● Yes

Select the device where the GRUB should be installed.

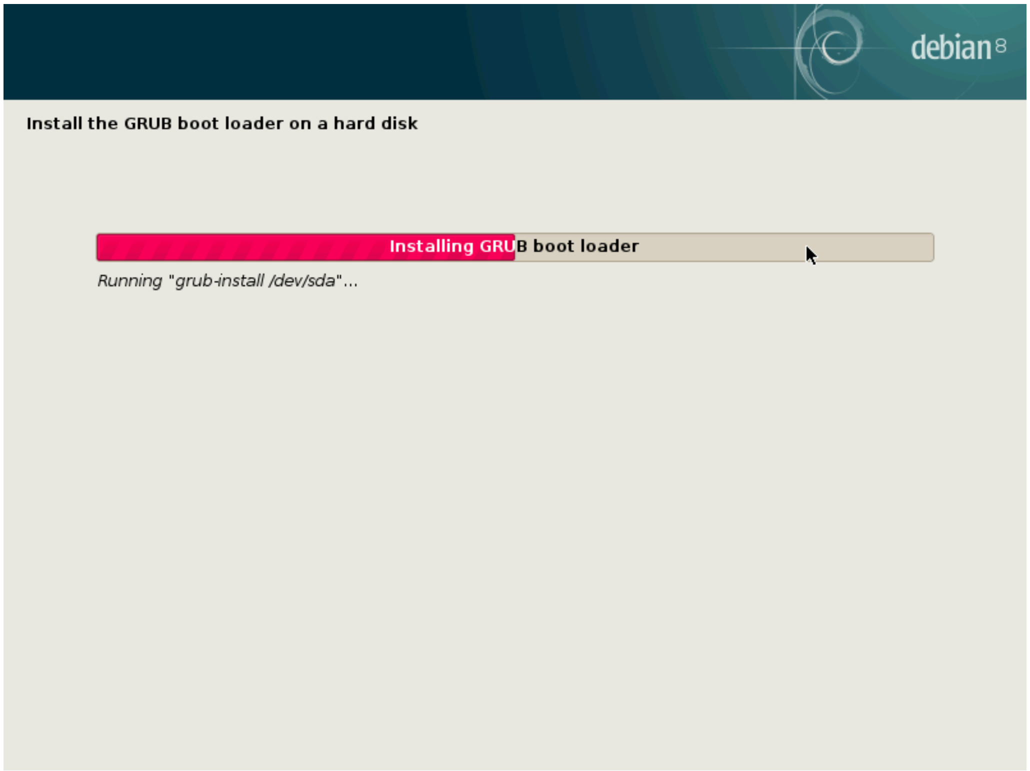


Install the GRUB boot loader on a hard disk

You need to make the newly installed system bootable, by installing the GRUB boot loader on a bootable device. The usual way to do this is to install GRUB on the master boot record of your first hard drive. If you prefer, you can install GRUB elsewhere on the drive, or to another drive, or even to a floppy.

*Device for boot loader installation:*

Enter device manually
/dev/sda (ata-VBOX_HARDDISK_VBfeb0f77b-bb6e785d)

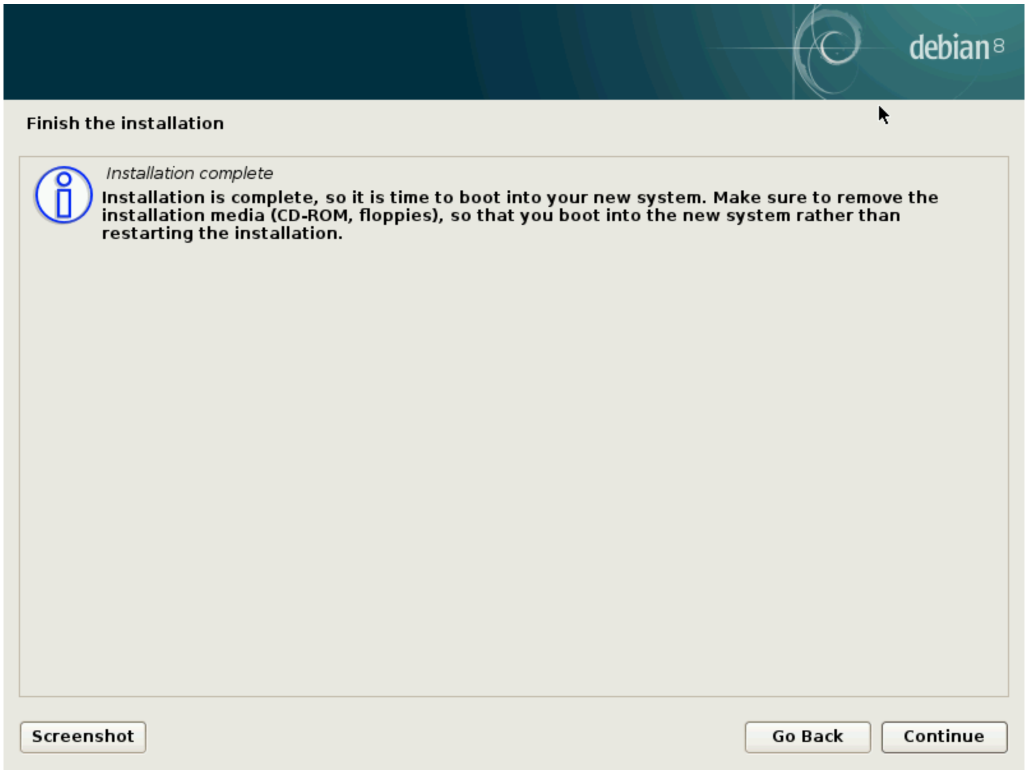


Install the GRUB boot loader on a hard disk

Installing GRUB boot loader

*Running "grub-install /dev/sda"...*

The installation is finished. Remove the CD-Rom and Debian will boot.

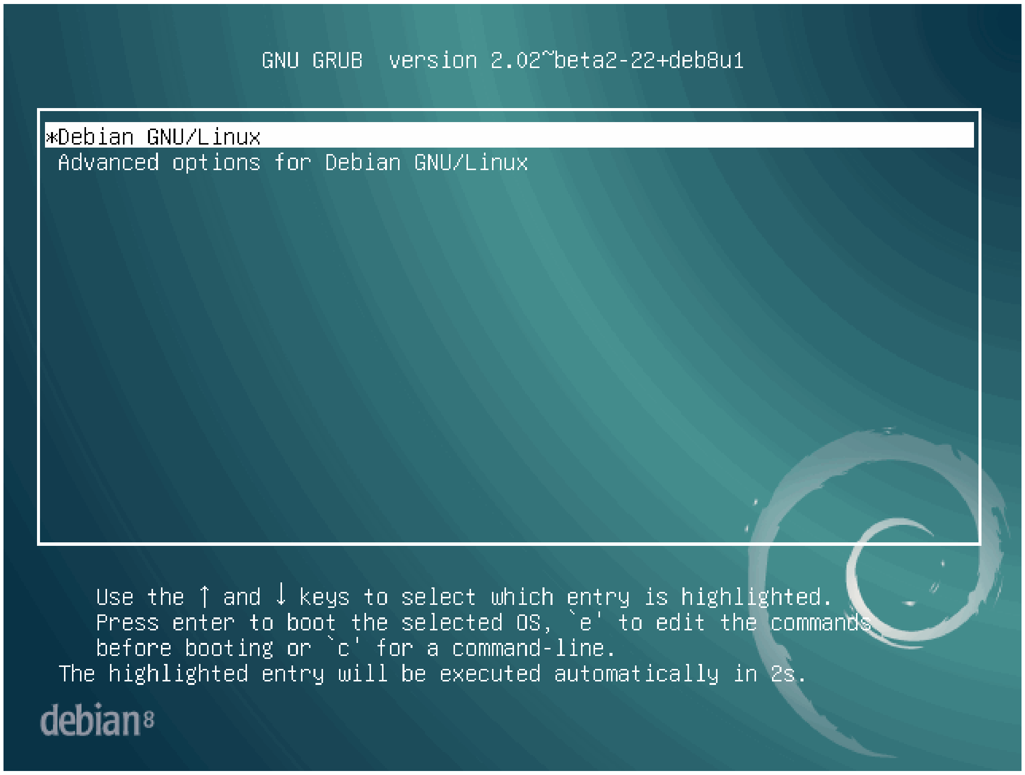


Finish the installation

*Installation complete*

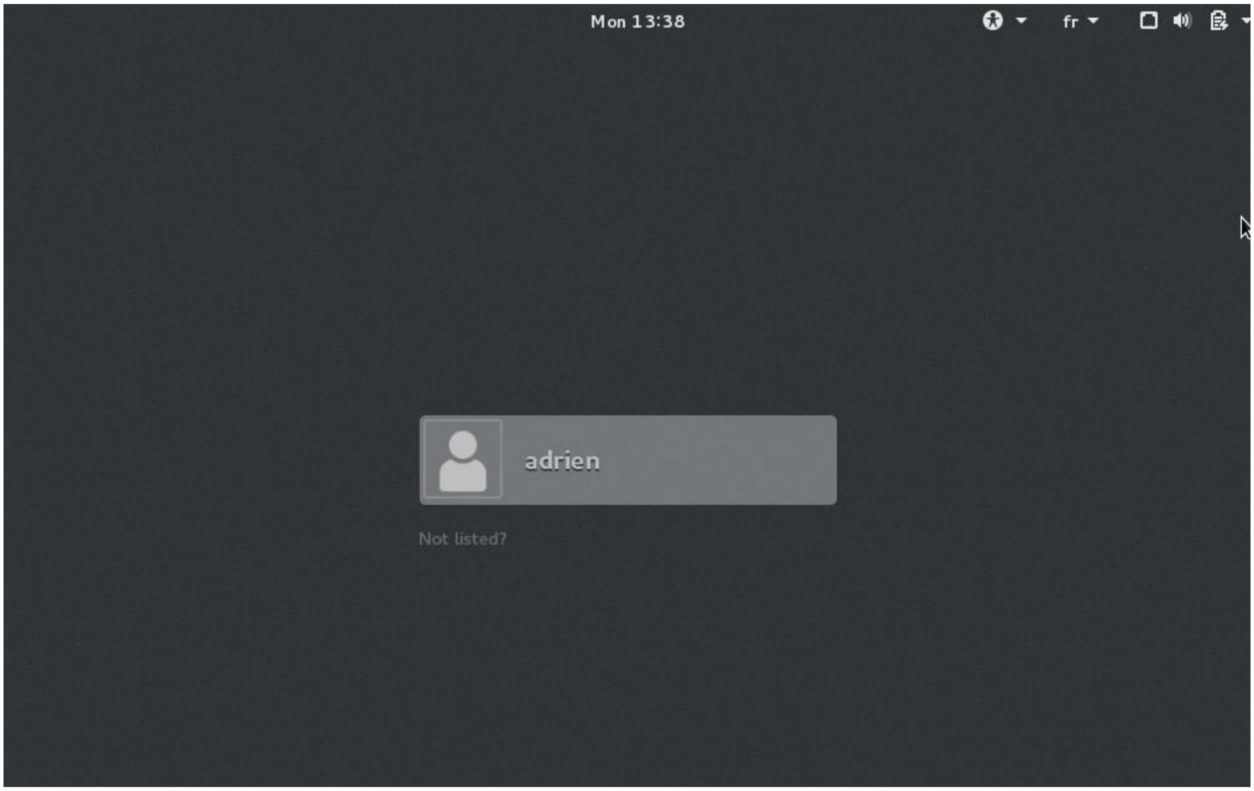
**Installation is complete, so it is time to boot into your new system. Make sure to remove the installation media (CD-ROM, floppies), so that you boot into the new system rather than restarting the installation.**

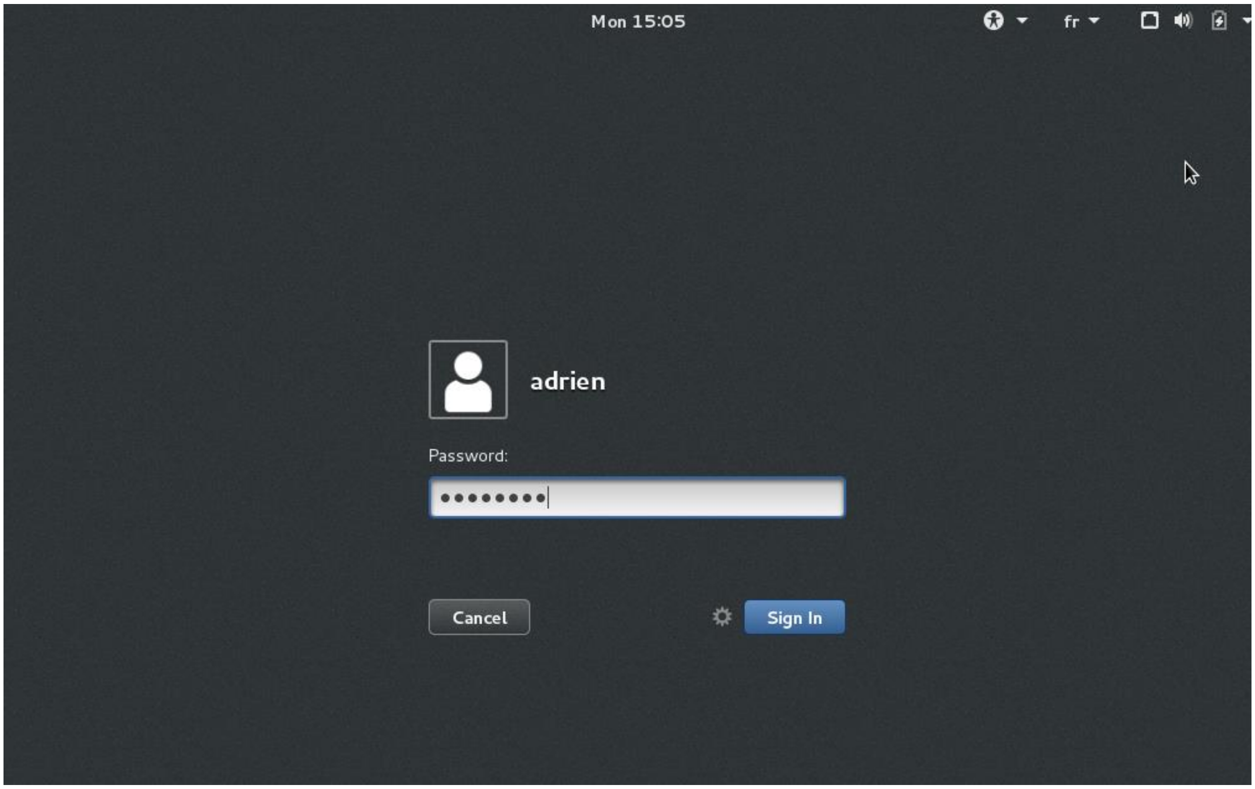
Select Debian from the GRUB menu.



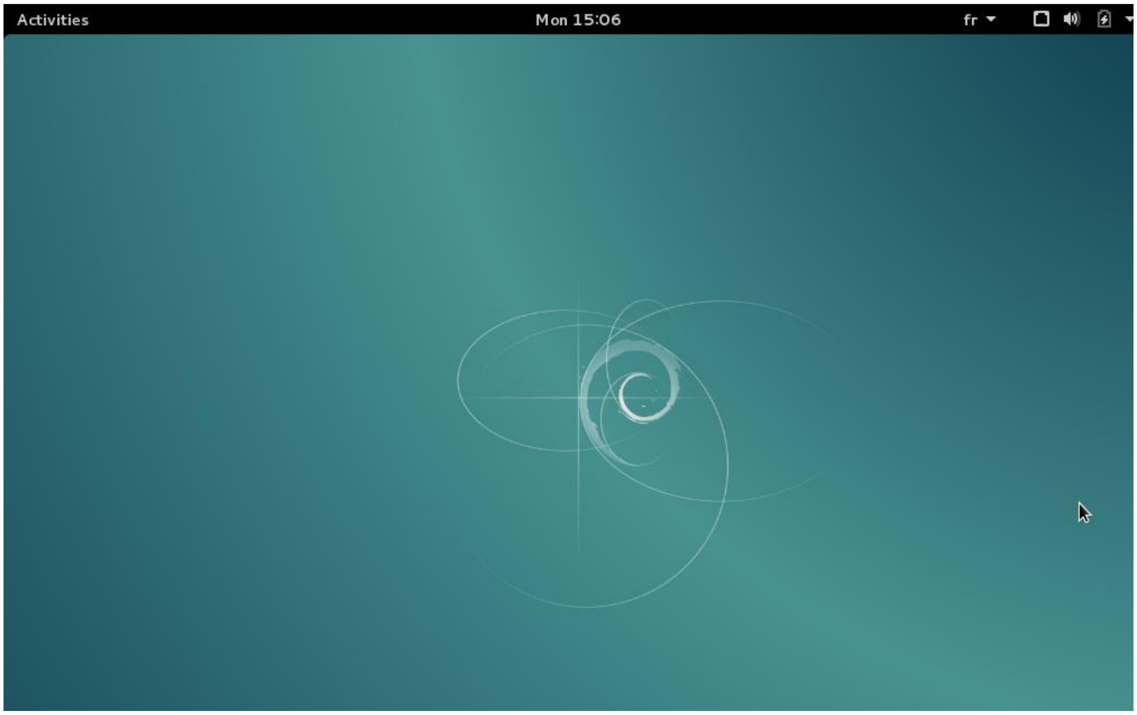
```
GNU GRUB  version 2.02~beta2-22+deb8u1

*Debian GNU/Linux
 Advanced options for Debian GNU/Linux

   Use the ↑ and ↓ keys to select which entry is highlighted.
   Press enter to boot the selected OS, `e' to edit the commands
   before booting or `c' for a command-line.
The highlighted entry will be executed automatically in 2s.
```

Debian is launched.

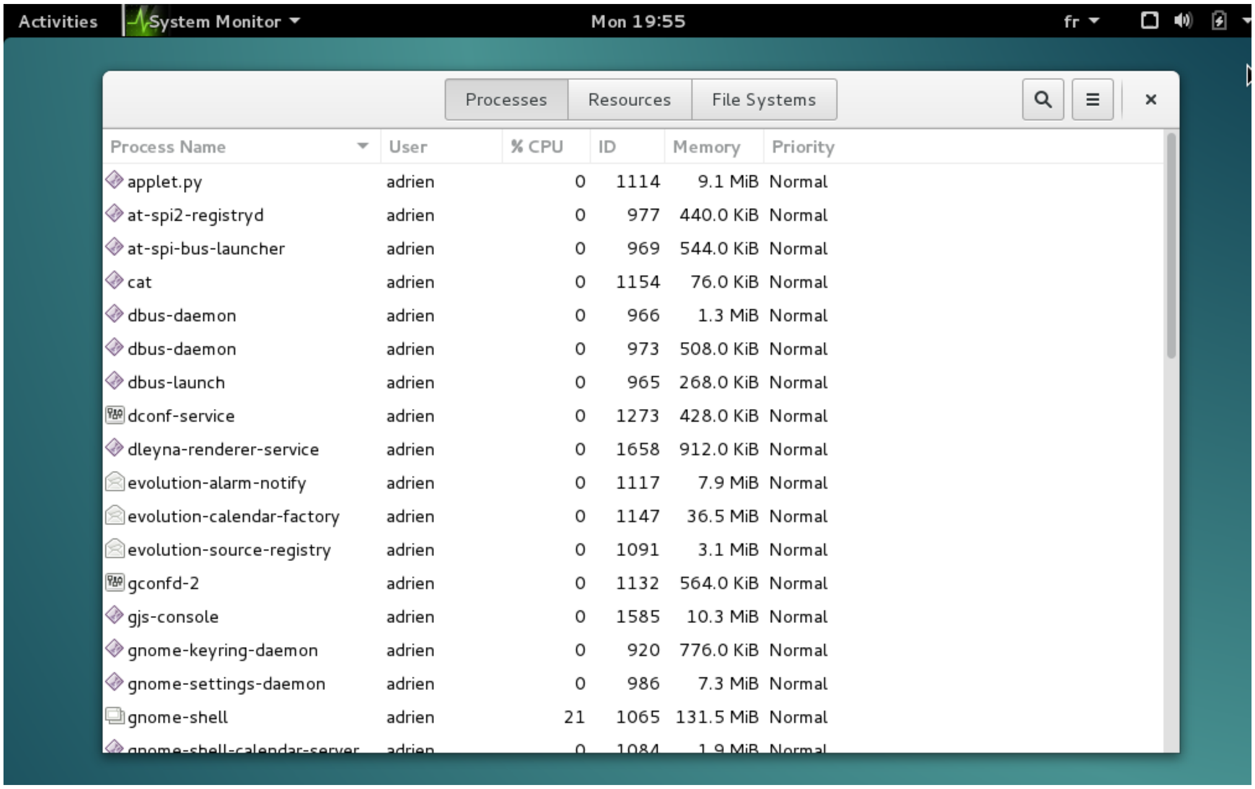




GNOME desktop:



GNOME offers a feature ("System Monitor") to view running processes:



| Process Name | User | % CPU | ID | Memory | Priority |
|---|---|---|---|---|---|
| applet.py | adrien | 0 | 1114 | 9.1 MiB | Normal |
| at-spi2-registryd | adrien | 0 | 977 | 440.0 KiB | Normal |
| at-spi-bus-launcher | adrien | 0 | 969 | 544.0 KiB | Normal |
| cat | adrien | 0 | 1154 | 76.0 KiB | Normal |
| dbus-daemon | adrien | 0 | 966 | 1.3 MiB | Normal |
| dbus-daemon | adrien | 0 | 973 | 508.0 KiB | Normal |
| dbus-launch | adrien | 0 | 965 | 268.0 KiB | Normal |
| dconf-service | adrien | 0 | 1273 | 428.0 KiB | Normal |
| dleyna-renderer-service | adrien | 0 | 1658 | 912.0 KiB | Normal |
| evolution-alarm-notify | adrien | 0 | 1117 | 7.9 MiB | Normal |
| evolution-calendar-factory | adrien | 0 | 1147 | 36.5 MiB | Normal |
| evolution-source-registry | adrien | 0 | 1091 | 3.1 MiB | Normal |
| gconfd-2 | adrien | 0 | 1132 | 564.0 KiB | Normal |
| gjs-console | adrien | 0 | 1585 | 10.3 MiB | Normal |
| gnome-keyring-daemon | adrien | 0 | 920 | 776.0 KiB | Normal |
| gnome-settings-daemon | adrien | 0 | 986 | 7.3 MiB | Normal |
| gnome-shell | adrien | 21 | 1065 | 131.5 MiB | Normal |
| gnome-shell-calendar-server | adrien | 0 | 1084 | 1.9 MiB | Normal |

## b. System Configuration Details

Debian requires very low resources in order to work properly.

If one needs an interface (GUI) such as GNOME or KDE, a minimal amount of 128 megabytes of RAM are required (512 are recommended), as well as 5 gigabytes of hard drive space.

If one does not need an elaborated desktop, the resources needed are less important: only 64 megabytes of RAM are required (256 are recommended) and 1 gigabyte of hard drive space.

However, the actual minimum memory requirements are less than the numbers listed above. On some architectures, it is possible to install Debian with as low as 20 megabytes.

For desktop usage, a Pentium 4 1Ghz is recommended.

## c. Process Control Management – CPU scheduling algorithms

LO – CPU Scheduling algorithms: **program which controls / manages all processes.**

**Each OS uses one CPU scheduling algorithm.**

### i. First-come, first-served

The FCFS is one of the CPU scheduling algorithms. This algorithm executes the processes in sequential orders. This algorithm is not efficient, because it produces high rate of average waiting time. In this algorithm, the process in two front positions in the ready queue is executed first while two process in the last position in the ready queue is invited for execution at the end.

| **Process** | **CPU burst time in ms** | **Waiting time for process** | **Turnaround time** |
|---|---|---|---|
| **P1** | 10 | 0 | 10 |
| **P2** | 1 | 10 | 11 (10 + 1) |
| **P3** | 2 | 11 | 13 (11 + 2) |
| **P4** | 1 | 13 | 14 (13 + 1) |
| **P5** | 5 | 14 | 19 (14 + 5) |

Average waiting time for the processes P1, P2, P3, P4, P5 = 0+10+11+13+14 = 48/5 = 9.6 ms

Turnaround time = waiting time + CPU burst time

Average turnaround time for the processes P1, P2, P3, P4, P5 = 10+11+13+14+19 = 67/5 = 13.4 ms

| | | | | |
|---|---|---|---|---|
| P1 | P2 | P3 | P4 | P5 |

0 10 11 13 14 19

Front Last, ready queue

## ii. Priority

| Process | CPU Burst time in ms | Priority | Waiting time | Turnaround time |
|---|---|---|---|---|
| **P1** | 10 | 0 | 0 | 10 |
| **P2** | 4 | 4095 | 33 | 37 |
| **P3** | 9 | 95 | 24 | 33 |
| **P4** | 1 | 40 | 23 | 24 |
| **P5** | 13 | 1 | 10 | 23 |

Average waiting time for the processes P1, P2, P3, P4, P5 = 0+33+24+23+10 = 90/5 = 18 ms

Average turnaround time for the processes P1, P2, P3, P4, P5 = 10+37+33+24+23 = 127/5 = 25.4 ms

Note: low number represent high-priority

| P1 | P5 | P4 | P3 | P2 |
|---|---|---|---|---|

0 10 23 24 33 37

High priority Low priority

Priority CPU scheduling algorithm: one of the CPU scheduling algorithm. In this algorithm, each process enters into the ready queue an integer number called “priority number”. In some situations, low priority number (0) will be given “high respect” while in some situations the high priority number will be given “high respect”.

The high priority process will be executed first while the low priority process will be executed at the end.

Weakness: this algorithm caver a problem called “starvation”. Low priority process may not be getting a chance to enter into the CPU. The low priority process may be ignored by the CPU. This problem is called “saturation”.

Solution to solve the starvation: aging technique – it is a simple technique, which either increments or decrements the priority number of the low priority process by 1.

For example:

| ↓ | | ↑ |
|---|---|---|
| 97 | 98 | |
| 96 | 97 | |
| 95 | 95 | |

It is a preemptive scheduling algorithm.

FCFS is not a preemptive algorithm.

### iii. Round robin

Time slice = 2ms.

| Process | CPU Burst time in ms | Waiting time (ms) | Turnaround time (ms) |
|---|---|---|---|
| **P1** | 10 = 8 = 6 = 4 = 2 = 0 | 21 (0+7+6+4+4) | 31 |
| **P2** | 4 = 2 = 0 | 9 (2+7) | 13 |
| **P3** | 9 = 7 = 5 = 3 = 1 = 0 | 23 (4+7+4+4+4) | 32 |
| **P4** | 1 = 0 | 6 | 7 |
| **P5** | 13 = 11 = 9 = 7 = 5 = 3 = 1 = 0 | 24 (7+6+4+4+3) | 37 |

Average waiting time: 24ms

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 8 | 2 | 7 | 0 | 11 | 6 | 0 | 5 | 9 | 4 | 3 | 7 | 2 | 1 | 5 | 0 | 0 | 3 | 1 | 0 |
| P1 | P2 | P3 | P4 | P5 | P1 | P2 | P3 | P5 | P1 | P3 | P5 | P1 | P3 | P5 | P1 | P3 | P5 | P5 | P5 |

0 2 4 6 7 9 11 13 15 17 19 21 23 25 27 29 31 32 34 36 37

The round robin is an efficient CPU algorithm. This algorithm does not cause the "starvation" problem. It is a preemptive CPU scheduling algorithm. This algorithm gives equal respect to each process. This algorithm handles all processes equally. It produces a small amount of average waiting time. The time-slice or time quantum concept is used in this algorithm.

### iv. Shortest job first

| Process | CPU Burst time in ms | Waiting time in ms |
|---|---|---|
| P1 | 10 | 14 |
| P2 | 4 | 1 |
| P3 | 9 | 5 |
| P4 | 1 | 0 |
| P5 | 13 | 24 |

| P4 | P2 | P3 | P1 | P5 |
|---|---|---|---|---|

0 1 5 14 24 37

Low CPU burst time High

Average waiting time: 44/5 = 8.8ms

Average turnaround time: 81/5 = 16.2ms

Every CPU scheduling algorithm has its own weakness and strengths. The shortest job first is one of the simplest CPU scheduling algorithms. In this algorithm, the CPU first executes the process which requires a least CPU's valuable (burst) time. The CPU executes the process which requires a huge amount of time at the end. This algorithm also causes "starvation" among the processes which require a huge amount of CPU time (CPU burst time).

The process, which require a huge CPU's time, may be ignored or may not be getting a change to enter into the CPU.

Solution to resolve the starvation problem: aging technique.

### v. Completely Fair Scheduler

Debian does not use any of the algorithms explained above. Instead, it uses the Completely Fair Scheduler (CFS). It aims to maximize overall CPU utilization while also maximizing interactive performance. It is an improvement of Fair-share scheduling which strategy is to strategy is to recursively apply the round-robin scheduling algorithm at each level of abstraction (processes, users, groups, etc.).

To sum up, it is similar to the round-robin algorithm as it tries to give each processes equal chances to complete.

## 3. Memory Management – Memory allocation algorithms

### a. Introduction to the Memory management

Having the following processes in the waiting list, let’s see how they fill RAM.

| Process ID | Size of process in kb |
|---|---|
| P1 | 1500 |
| P2 | 100 |
| P3 | 100 |
| P4 | 23 |
| P5 | 1400 |
| P6 | 99 |

| | |
|---|---|
| OS area | 0kb<br>640kb |
| Free hole area 1760kb | 640kb<br>2400kb |

| | |
|---|---|
| OS area | 0kb<br>640kb |
| P1 1500kb | 640kb<br>2140kb |
| Free hole area 260kb | 2140kb<br>2400kb |

| | |
|---|---|
| OS area | 0kb<br>640kb |
| P1 1500kb | 640kb<br>2140kb |
| P2 100kb | 2140kb<br>2240kb |
| Free hole area 160kb | 2240kb<br>2400kb |

| | |
|---|---|
| OS area | 0kb<br>640kb |
| P1 1500kb | 640kb<br>2140kb |
| P2 100kb | 2140kb<br>2240kb |
| P3 100kb | 2240kb<br>2340kb |
| Free hole area 60kb | 2340kb<br>2400kb |

| | |
|---|---|
| OS area | 0kb<br>640kb |
| P1 1500kb | 640kb<br>2140kb |
| P2 100kb | 2140kb<br>2240kb |
| P3 100kb | 2240kb<br>2340kb |
| P4 23kb | 2340kb<br>2363kb |
| Free hole area 37kb | 2363kb<br>2400kb |

| | |
|---|---|
| OS area | 0kb<br>640kb |
| Free hole area 1: 1500kb | 640kb<br>2140kb |
| P2 100kb | 2140kb<br>2240kb |
| P3 100kb | 2240kb<br>2340kb |
| P4 23kb | 2340kb<br>2363kb |
| Free hole area 2: 37kb | 2363kb<br>2400kb |

Swap out P1 to accomodate P5

| | |
|---|---|
| OS area | 0kb<br>640kb |
| P5 1400kb | 640kb<br>2040kb |
| Free hole area 1: 100kb | 2040kb<br>2140kb |
| P2 100kb | 2140kb<br>2240kb |
| P3 100kb | 2240kb<br>2340kb |
| P4 23kb | 2340kb<br>2363kb |
| Free hole area 2: 37kb | 2363kb<br>2400kb |

Swap-in P5 to the first hole area

Swap in: moving process from backing store (BS = HD) to RAM

Swap out: transferring unwanted process from the RAM to BS

Swapper: it is the process created at system startup time, which is also the first process created by the system. It is referred as the ‘idle task’ and ensures that at least one process is in the process scheduling queue.

Frame: the user area of the RAM, subdivided into logical equal sized partitions, called “frames”. Each frame can hold only one age at a time.

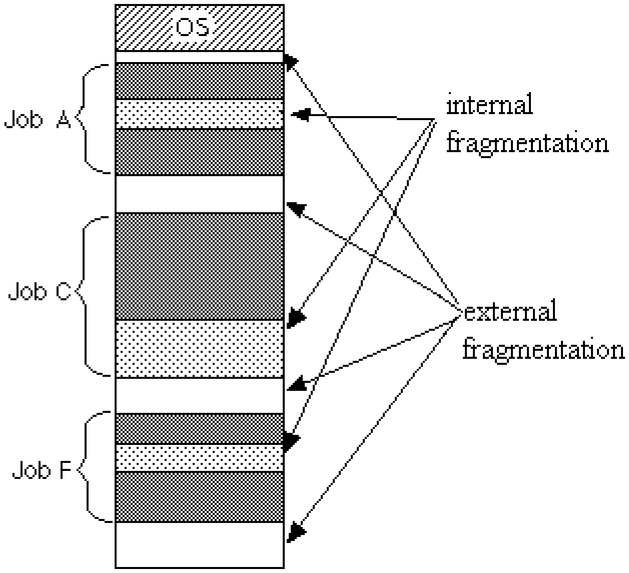
Page: each bug process is subdivided into smaller processes called “pages”

Pager: it is a software component. It is a part of an operating system. It handles pages while the swapper handles processes.

## b. Internal fragmentation, External Fragmentation and Compaction techniques

Fixed memory partition/allocation causes a special problem called “internal fragmentation”. It means that the free hole areas found here and there inside the RAM cannot be merged or cannot be reused because it has already been inefficiently assigned to a process. In other terms, the free hole areas are not contiguous.

The dynamic allocation/partition causes a special problem called “external fragmentation”. It happens in some situations, where the free hole areas found inside RAM cannot be reused even though the sum of the size of all free hole areas is superior or equal to the size of a needy process.



*Source: http://stackoverflow.com/questions/1200694/internal-and-external-fragmentation*

This diagram shows the difference between internal fragmentation that happens inside a process and external fragmentation that happens between processes.

This implies that the free hole areas not found at one location inside the RAM. They are found at various locations inside the RAM.

For example, the sum of the size of the free hole areas 1 and 2 is 38kb. The size of the needy process P7 is 38kb. But the RAM is unable to accommodate the P7 unless the free hole areas are merged together, which is explained in the following diagrams.

| | |
|---|---|
| OS area | 0kb<br>640kb |
| P5 1400kb | 640kb<br>2040kb |
| Free hole area 1: 1kb | 2040kb<br>2041kb |
| P2 100kb | 2041kb<br>2141kb |
| P3 100kb | 2141kb<br>2241kb |
| P4 23kb | 2241kb<br>2264kb |
| Free hole area 2: 37kb | 2264kb<br>2301kb |

| | |
|---|---|
| OS area | 0kb<br>640kb |
| P5 1400kb | 640kb<br>2040kb |
| P2 100kb | 2040kb<br>2140kb |
| P3 100kb | 2140kb<br>2240kb |
| P4 23kb | 2240kb<br>2263kb |
| P7 38kb | 2263kb<br>2301kb |

37+1 = 38kb free compacted

Compaction technique: it merges all the free hole areas together in order to make a big free hole area so that the big merged area free hole can be reused to accommodate any needy process.

To sum up, what are the differences between external and internal fragmentation?

| External fragmentation | Internal fragmentation |
|---|---|
| The dynamic memory allocation partition causes the external fragmentation | The fixed memory partition allocation causes the internal fragmentation problem |
| The fixed hole areas found inside the RAM can be merged and reused | The free hole areas found inside the RAM cannot be merged or reused |
| The compaction technique is applied in order to merge all free hole areas | The compactor technique cannot be applied here |

## c. Page replacement algorithms

Page replacement algorithms are responsible for the decision to swap out memory pages when a new page of memory has to be allocated. They decide which memory page should be swapped out.

Each operating systems requires it in order to maintain its stability and avoid having a large number of page faults, which happen when there is no free page available to satisfy the allocation. It is crucial because it minimizes total time waiting for memory.

For example, we will compare the two algorithms FIFO and LRU with the same sequence of processes: 0,4,1,4,2,4,3,4,2,4,0,4,1,4,2,4,3,4.

The FIFO (First In, First Out) is one of the simplest replacement algorithms. In this algorithm, the page who first entered into the RAM, is chosen for page-out. This algorithm causes a high rate of page faults. The entire frames are searched for in order to find a page, to be page-out.

| 0 | 4 | 1 | 4 | 2 | 4 | 3 | 4 | 2 | 4 | 0 | 4 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | **0** | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 0 |
|  | 4 | 4 | 4 | 4 | 4 | 3 | 3 | 3 | 3 | 3 | 3 |
|  |  | 1 | 1 | 1 | 1 | 1 | 4 | 4 | 4 | 4 | 4 |
| PF=1 | PF=2 | PF=3 | NoPF | PF=4 | No PF | PF=5 | PF=6 | No PF | No PF | PF=7 | No PF |

| 1 | 4 | 2 | 4 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 4 | 4 | 4 |
| 1 | 1 | 1 | 1 | 3 | 3 |
| 4 | 4 | 2 | 2 | 2 | 2 |
| PF=8 | No PF | PF=9 | PF=10 | PF=11 | No PF |

Number of page fault that would occur/happen = 11

LRU (Least Recently Used) is one of the efficient page replacement algorithms. This provides better performance. For this algorithm causes a minimum number of page fault events. This algorithm always looks for the page which has not been referred (or used) for a longest period of time, in order to page-out.

| 0 | 4 | 1 | 4 | 2 | 4 | 3 | 4 | 2 | 4 | 0 | 4 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | **0** | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
|  | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
|  |  | 1 | 1 | 1 | 1 | 3 | 3 | 3 | 3 | 0 | 0 |
| PF=1 | PF=2 | PF=3 | NoPF | PF=4 | No PF | PF=5 | NoPF | No PF | No PF | PF=6 | No PF |

| 1 | 4 | 2 | 4 | 3 | 4 |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 3 | 3 |
| 4 | 4 | 4 | 4 | 4 | 4 |
| 0 | 0 | 2 | 2 | 1 | 1 |
| PF=7 | No PF | PF=8 | No PF | PF=9 | No PF |

Number of page fault that would occur/happen = 9

Debian has a Linux kernel which claims to use a “Page Frame Reclaiming Algorithm”, which is basically a Least Recently Used algorithm.

## d. Memory allocation strategies

Below are the 3 memory allocation strategies:

- First-fit: it starts searching operation from the first free hole area in RAM. The searching operation stops at the moment it finds the first free hole which can accommodate a needy process

- Best-fit: it starts its searching operation from the first free hole area. The entire free holes are searched in order to find the smallest free hole area which can accommodate a needy process. It is efficient because it causes less memory wastage.

- Worst-fit: it starts its searching operation from the first free hole area in RAM. The entire free hole areas are searched in order to find the biggest free hole area which can accommodate a needy process. It stops its searching operation at the moment it finds the biggest free hole area to accommodate the needy process. No OS uses this strategy for it is a greedy strategy and it causes a huge memory wastage each time.

For example, if we have the following situation and we want to allocate 12kb, let's see how the algorithms would behave:

| |
|---|
| OS area |
| Free hole area 1: 6kb |
| Allocated memory |
| Free hole area 2: 14kb |
| Allocated memory |
| Allocated memory |
| Free hole area 3: 19kb |
| Allocated memory |
| Free hole area 4: 11kb |
| Allocated memory |
| Free hole area 5: 13kb |

| |
|---|
| OS area |
| Free hole area 1: 6kb |
| Allocated memory |
| Free hole area 2: 14kb |
| Allocated memory |
| Allocated memory |
| Free hole area 3: 19kb |
| Allocated memory |
| Free hole area 4: 11kb |
| Allocated memory |
| **New allocated memory: 12kb** |
| **Free hole area 5: 1kb** |

Best fit

| |
|---|
| OS area |
| Free hole area 1: 6kb |
| Allocated memory |
| Free hole area 2: 14kb |
| Allocated memory |
| Allocated memory |
| **New allocated memory: 12kb** |
| **Free hole area 3: 7kb** |
| Allocated memory |
| Free hole area 4: 11kb |
| Allocated memory |
| Free hole area 5: 13kb |

Worst fit

| |
|---|
| OS area |
| Free hole area 1: 6kb |
| Allocated memory |
| **New allocated memory: 12kb** |
| Free hole area 2: 2kb |
| Allocated memory |
| Allocated memory |
| Free hole area 3: 19kb |
| Allocated memory |
| Free hole area 4: 11kb |
| Allocated memory |
| Free hole area 5: 13kb |

First fit

Belady, an OS expert, found that the number of page-fault events reduced when the number of frames increased. But the OS expert failed to demo that this happens in all page-replacement algorithms.

Debian uses an optimistic memory allocation strategy, meaning that even though the system returns a non-null object when using the instruction malloc, there is no guarantee that the memory is actually available. It means that the memory allocation procedure will always succeed.

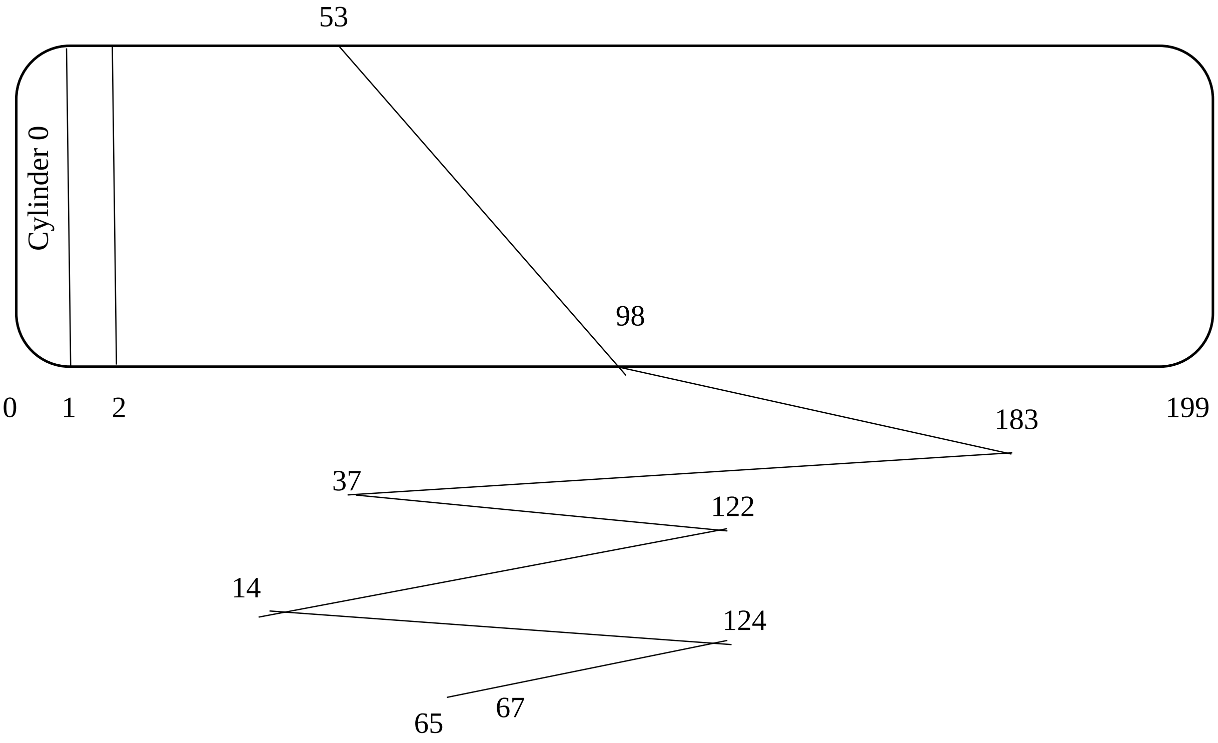
However, memory is not actually committed to the requesting process until it is really used by the process. If the memory is already full, one or more processes will be killed.

## 4. Secondary-Storage Management – Disk Scheduling algorithms

The queue is: 53, 98, 183, 37, 122, 14, 124, 65, 67.

The head is initially at 53, the head is at 0 and the tail at 199.

### a. First-come, first-served



Disk movements:

53~98=45

98~183=85

183~37=146

37~122=85

122~14=108

14~124=110

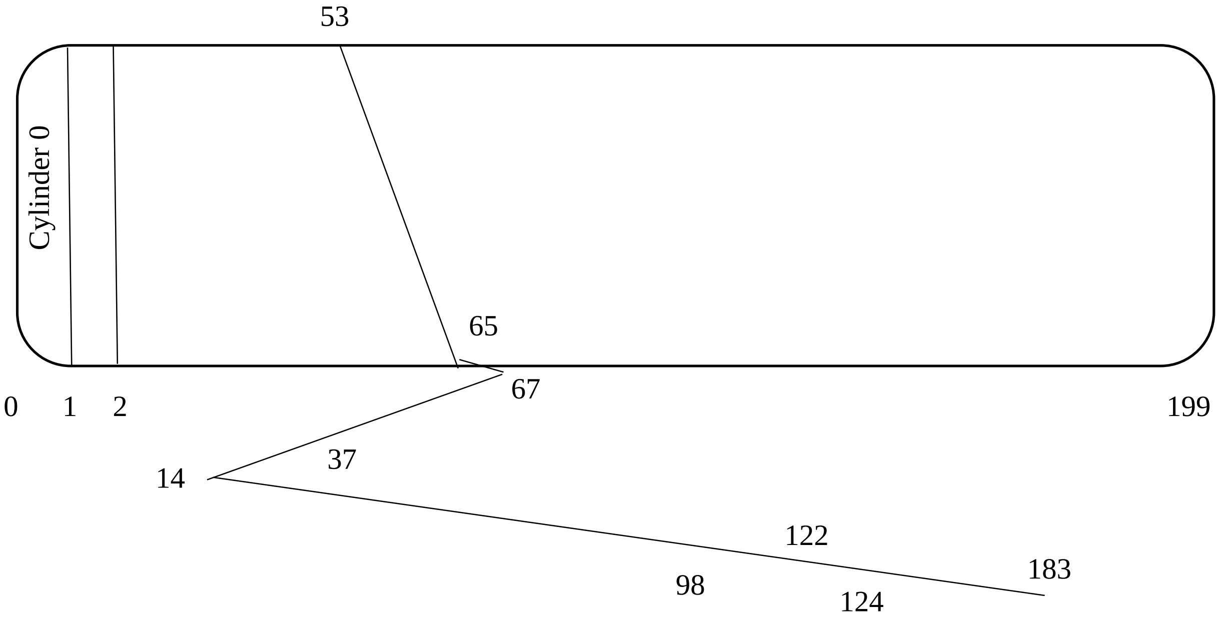
124~65=59

65~67=2

- ⇨ 640 cylinders (total number of disk movements occurred)
- ⇨ 7 swings

First-come, first-served is one of the simplest disk scheduling algorithms. All read/write requests are served by the read/write head. No request is suffering from the “starvation” problem. The read/write head gives service to each request in sequential order. This algorithm causes a high rate of swings.

## b. Shortest Seek Time First



Disk movements:

53~65=12

65~67=2

67~37=30

37~14=23

14~98~84

98~122=24

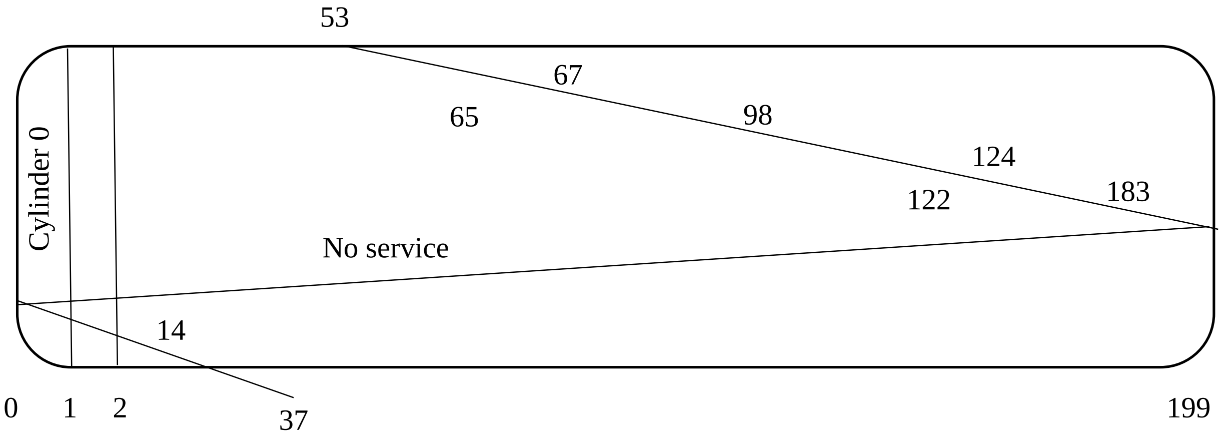
122~124=2

124~183=59

⇨ 238 cylinders

Shortest Seek Time First provides better performance than FCFS. The read/write head moves to the request, which is the nearest to its current position. The read/write head gives service to all the request found in the queue. It causes a low number of swings.

### c. C-Scan



Disk movements:

53~65=12

65~67=2

67~98=31

98~122=24

122~124=2

124~183=59

183~199=16

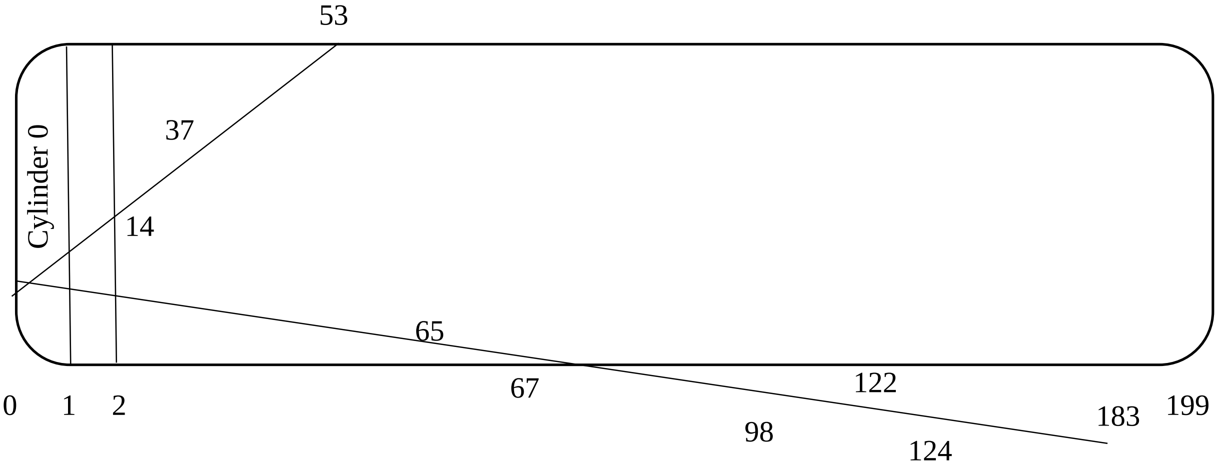
0~14=14

14~37=23

⇨ 183 cylinders

C-Scan disk scheduling algorithm is one of the efficient disk scheduling algorithms. It provides better performance than the FCFS and the SSTF disk scheduling algorithm. It treats the HD as a circular disk. The read/write head always travels from one end (0th cylinder) to other end (the last cylinder). The algorithm is really an efficient algorithm because it causes a low number cylinder movements and low number of swings. The read/write head gives service to the requests found on its way to the 199th cylinder. Whilst, the read write does not give service to the requests found on its way to 0th cylinder.

### d. Scan



Disk movements:

53~37=16

37~14=23

14~0=14

0~65=65

65~67=2

67~98=31

98~122=24

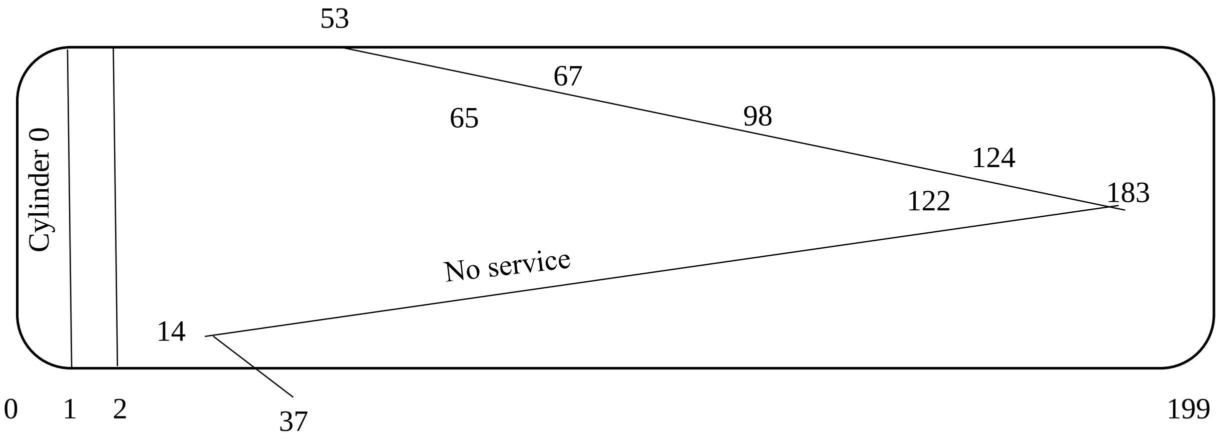
122~124=2

124~183=59

⇨ 236 cylinders

This disk scheduling algorithm provides better performances than the FCFS and SSTF. Nevertheless, it is not an efficient disk scheduler algorithm. It behaves like an elevator. The read/write head travels from one of the disk (0th cylinder) to other end of the disk (last cylinder). The read/write head gives service to the requests found on its way to the 0th cylinder to the last cylinder. When the read/write head travels towards the 0th cylinder, it travels up to the 0th cylinder. When the read/write head travels towards the last cylinder (end of disk) it travels up to the last cylinder.

## e. C-Look



Disk movements:

53~65=12

65~67=2

67~98=31

98~122=24

122~124=2

124~183=59

14~37=23

- ⇨ 153 cylinders

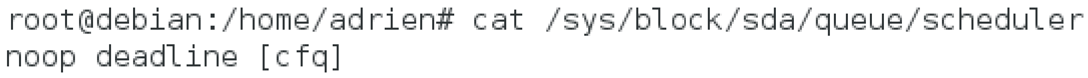
It is one of the efficient disk scheduling algorithm. It provides the best performance among the disk scheduling algorithms (FCFS, SSTF, C-scan, Scan) like (C-scan, Scan) the write/write head travels from one end of the disk to other end of the disk. But, when the read/write head travels towards the last cylinder (end of disk), it gives service to the last request found on its way. The read/write head does not travel up to the last cylinder (199th). It stops at the last request on its way to the end of disk, changes its direction and travels towards the beginning of the disk.

### f. Linux Disk Scheduling algorithms

Since Debian uses a Linux kernel, it offers three different disk scheduling algorithms.

- Noop, which is a simple disk scheduling algorithm, using the first in, first out principle to insert all requests in a queue. It implements requests merging.
- Deadline scheduler, which guarantees (or at least tries to) a start time for each request
- Completely fair queuing is the default disk scheduling algorithm. It tries to provide a fair share of disk service for each process.

The Anticipatory scheduling algorithm has been removed from Linux kernel since version 2.6.33.



```
root@debian:/home/adrien# cat /sys/block/sda/queue/scheduler
noop deadline [cfq]
```

*The cfq is the default disk scheduling algorithm on a fresh Debian installation*

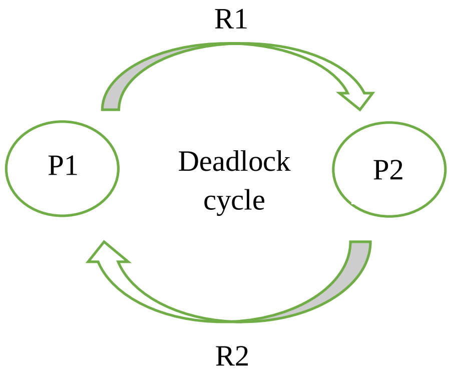
## 5. Deadlock and help

### a. Deadlock Management

#### i. Deadlock cycle

A deadlock is an unexpected situation that occurs in multi user networking environment.

R1, R2 are resources (printer, CD driver…)



P1, P2… are running programs

R1, R2… are resources like CD drive, printer, fax…

For example, the process P1 needs the resource P1, which P2 currently uses it while the process P2 needs the resource R2 which the process P1 currently uses.

Both processes will enter into a long-waiting state that causes a deadlock.

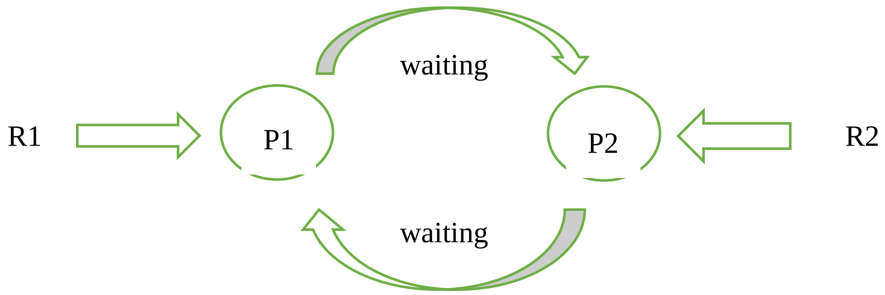
A deadlock situation may happen if the following conditions are met simultaneously:

- Circular wait
- Hold and wait
- No preemption
- Mutual exclusion

They are known as the "Coffman condition" and are described below.

No OS is good at handling deadlocks.

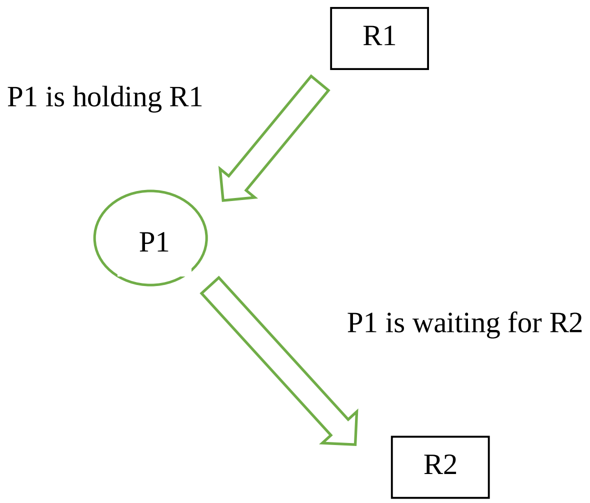
#### ii. Circular wait



This circular wait state may occur, when there are n numbers of processes and m number of resources in a multi user networking environment. The diagram shows that the process P1 badly in need of the resources R1 being by the process P2. The process P2 badly in need of the resources R2, which is currently being the process P3. The process Pn badly in need of the resources Rn which is currently being used by the process P1.

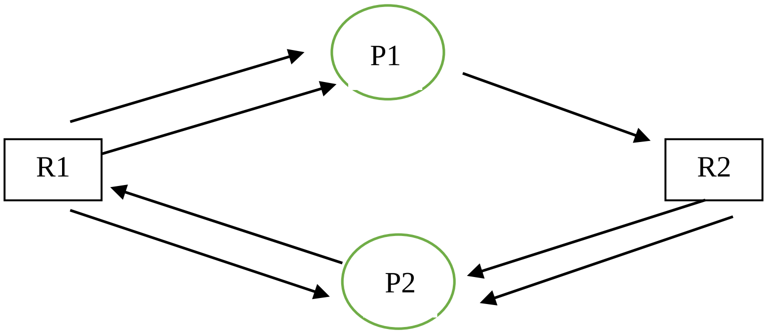
If this long-waiting state continues, thee waiting process will cause a "circular deadlock".

### iii. Hold and wait



This situation may occur when a process waits to get additional resources which is currently being held by another process. The diagram shows that the process P1 is currently holding 2 resources (printer and file). Instead of using two resources, the process is waiting to get an additional resource, fax machine, which is currently being held by the process P2.

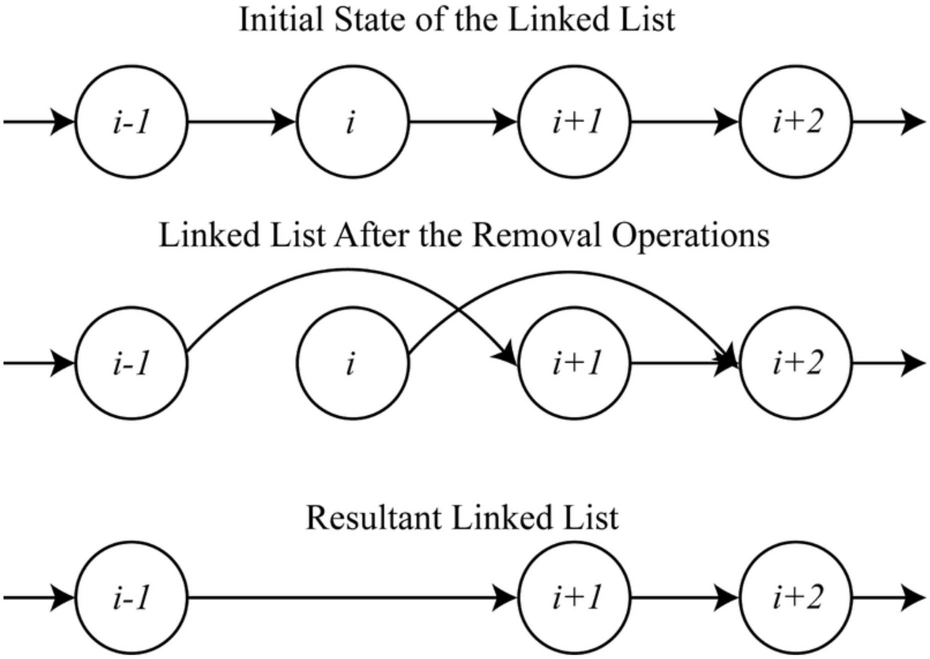
### iv. No preemption



This situation/state may occur when the following situation takes place. For example, the process P1 got a privilege to hold the resources CPU for a period of time. No other processes including the CPU can force the process P1 to exit before the allocated time is expired.

The CPU scheduling algorithm FIFO behaves like this strategy.

### v. Mutual exclusion


Initial State of the Linked List
i-1
i
i+1
i+2
Linked List After the Removal Operations
i-1
i
i+1
i+2
Resultant Linked List
i-1
i+1
i+2

*Two nodes are removed simultaneously (i and i+1)*

*In the end, only the node i is really removed*

*Source: Wikipedia, see the References section below*

This situation may occur when a valuable resource is shared by more than one process.

For example, the printer is currently being used by the process P1. The process P1 uses the printers on “Mutual exclusion” mode. It means no other processes can use this printer until the printer is released by the process P1.

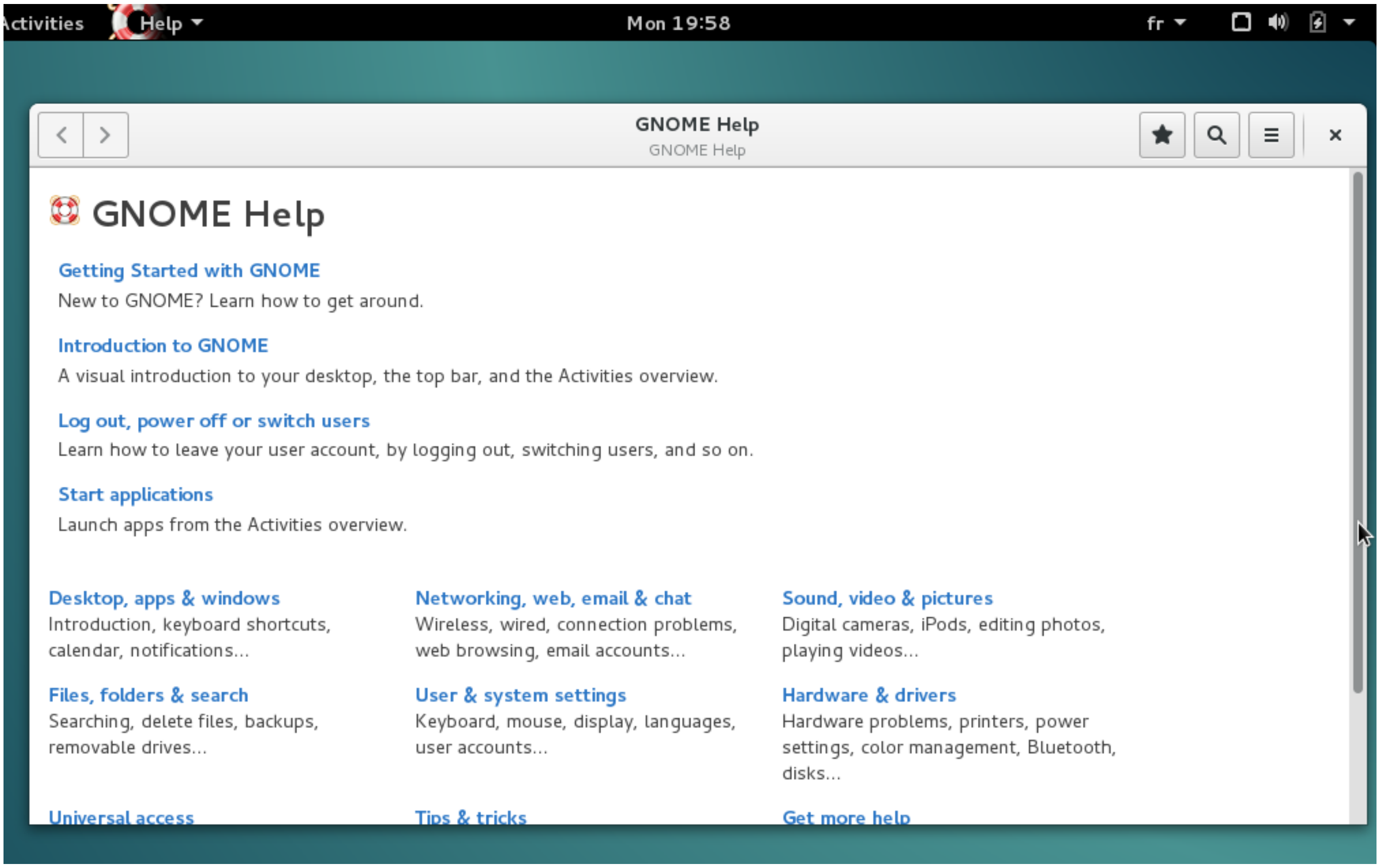
### vi. Deadlocks in Linux

As all the other operating systems, Linux kernel is not good at handling deadlocks. However, if there is no deadlock prevention for user applications or threads, it does take care about its own deadlocks.

For developers, “lockdep” is a tool allowing deadlock prevention.

## b. Help and Support

GNOME offers a “Help” feature accessible through the desktop:



GNOME Help

Getting Started with GNOME
New to GNOME? Learn how to get around.

Introduction to GNOME
A visual introduction to your desktop, the top bar, and the Activities overview.

Log out, power off or switch users
Learn how to leave your user account, by logging out, switching users, and so on.

Start applications
Launch apps from the Activities overview.

Desktop, apps & windows
Introduction, keyboard shortcuts, calendar, notifications...

Networking, web, email & chat
Wireless, wired, connection problems, web browsing, email accounts...

Sound, video & pictures
Digital cameras, iPods, editing photos, playing videos...

Files, folders & search
Searching, delete files, backups, removable drives...

User & system settings
Keyboard, mouse, display, languages, user accounts...

Hardware & drivers
Hardware problems, printers, power settings, color management, Bluetooth, disks...

Universal access

Tips & tricks

Get more help

## 6. Conclusion

Debian is a free Linux-based operating system, free and open-source. It is very popular for its stability and its ungreediness. It is easy to install thanks to his GUI installer and his GUI desktop, GNOME.

Every operating system having a CPU scheduling algorithm, Debian uses the completely fair scheduler. It is very efficient, despite having its own weakness and strengths like every CPU scheduling algorithm. It does not cause the “starvation” problem and is a preemptive CPU scheduling algorithm.

Debian uses the compaction technique to avoid having free RAM memory being wasted. To handle page replacement, a “Page Frame Reclaiming Algorithm” is used, which is basically a Least Recently Used algorithm.

Debian uses an optimistic memory allocation strategy, it means that the memory allocation procedure will always succeed. However, if the memory is already full, one or more processes will be killed.

The Linux kernel Debian relies on offers three different disk scheduling algorithms. The completely fair scheduling is the algorithm used by default; it tries to provide a fair share of disk service for each process.

No Operating System is good at handling deadlocks, which are caused by four simultaneous conditions: circular wait, hold and wait, no preemption and mutual exclusion. Nevertheless, if it does not provide support for user’s processes, Linux’s kernel takes care of its own deadlocks.

Finally, the GNOME desktop offers a GUI interface where a help interface as well as a process manager are available. If need be, the terminal is available.

## 7. References

3.4. Meeting Minimum Hardware Requirements. 2016. *3.4. Meeting Minimum Hardware Requirements*. [ONLINE] Available at: https://www.debian.org/releases/wheezy/amd64/ch03s04.html.en. [Accessed 30 May 2016].

Wikipedia. 2016. *Completely Fair Scheduler - Wikipedia, the free encyclopedia*. [ONLINE] Available at: https://en.wikipedia.org/wiki/Completely_Fair_Scheduler. [Accessed 30 May 2016].

malloc(3) - Linux manual page. 2016. *malloc(3) - Linux manual page*. [ONLINE] Available at: http://man7.org/linux/man-pages/man3/malloc.3.html. [Accessed 30 May 2016].

c++ - Linux optimistic malloc: will new always throw when out of memory? - Stack Overflow. 2016. *c++ - Linux optimistic malloc: will new always throw when out of memory? - Stack Overflow*. [ONLINE] Available at: http://stackoverflow.com/questions/1655650/linux-optimistic-malloc-will-new-always-throw-when-out-of-memory. [Accessed 30 May 2016].

Wikipedia. 2016. *CFQ - Wikipedia, the free encyclopedia*. [ONLINE] Available at: https://en.wikipedia.org/wiki/CFQ. [Accessed 30 May 2016].

Linux Change The I/O Scheduler For A Hard Disk. 2016. *Linux Change The I/O Scheduler For A Hard Disk*. [ONLINE] Available at: http://www.cyberciti.biz/faq/linux-change-io-scheduler-for-harddisk/. [Accessed 30 May 2016].

Wikipedia. 2016. *Mutual exclusion - Wikipedia, the free encyclopedia*. [ONLINE] Available at: https://en.wikipedia.org/wiki/Mutual_exclusion. [Accessed 30 May 2016].

Columbia University. 2010. *Deadlock Avoidance*. [ONLINE] Available at: https://www.cs.columbia.edu/~smb/classes/s06-4118/l10.pdf. [Accessed 30 May 2016].

concurrency - how does the linux kernel avoid deadlocks? - Stack Overflow. 2016. *concurrency - how does the linux kernel avoid deadlocks? - Stack Overflow*. [ONLINE] Available at: http://stackoverflow.com/questions/22170943/how-does-the-linux-kernel-avoid-deadlocks. [Accessed 30 May 2016].

| No | Student Name | Research and Investigation (30) | Installation Process (20) | Documentation (10) | Referencing (10) | Analysis (15) | Presentation (15) | Total (100) |
|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | |