1º DAW
Computer systems

Unit 4

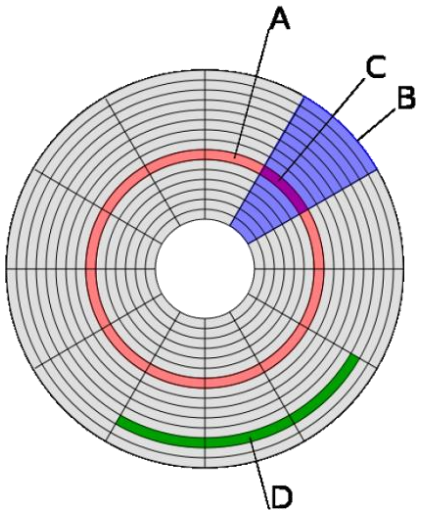
Installation of different operating systems in a computer

Index

1. Introduction to hard drives

Remember: Hard disk physical structure

- Head
- Tracks
- Cylinder
- Sectors
- Clusters

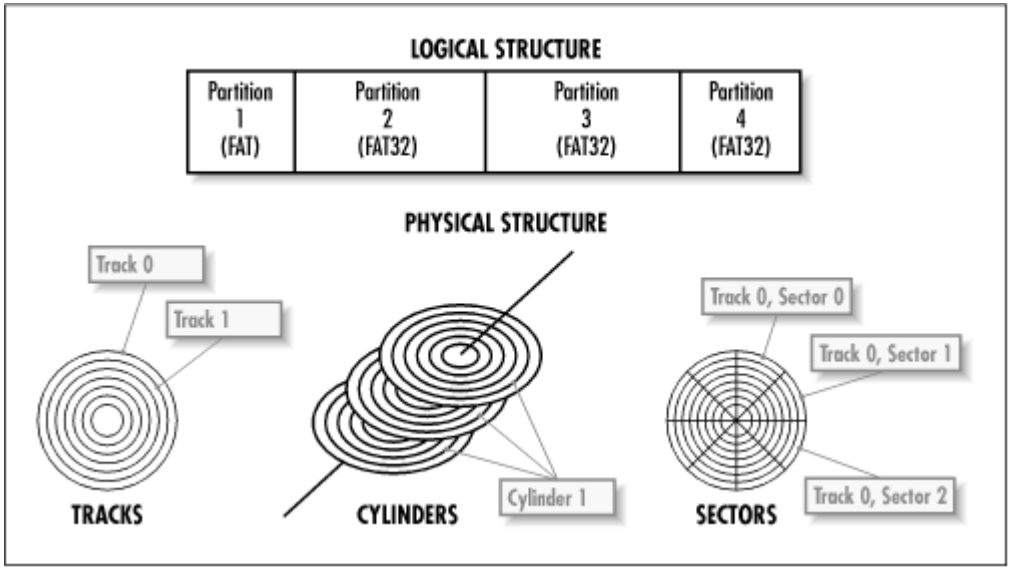


Hard Drive Structure:

A = track
B = sector
C = sector of a track
D = cluster

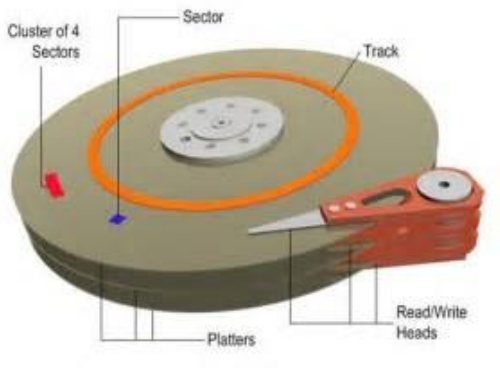
Hard disk logical structure

- Partitions



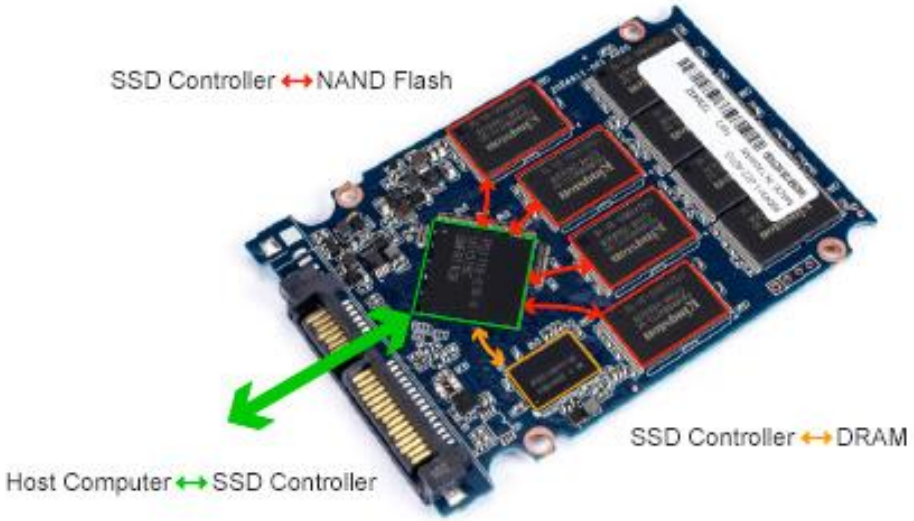
1. Introduction to hard drives

- Head: Every hard drive consists of platters and read-write heads. If a drive has four platters, it usually has eight read-write heads, one on the top and bottom of each platter.
- Tracks: Each platter is broken into thousands of tightly packed concentric circles, known as tracks. These tracks resemble the structure of annual rings of a tree.
- Cylinder: A cylinder is basically the set of all tracks that all the heads are currently located at.
- Sectors: Each track is further broken down into smaller units called sectors. As sector is the basic unit of data storage on a hard disk. A single track typically can have thousands of sectors and each sector can hold more than 512 bytes of data.
- Clusters: To reduce the overhead of managing on-disk data structures, the file system does not allocate individual disk sectors by default, but contiguous groups of sectors, called clusters.



1. Introduction to hard drives

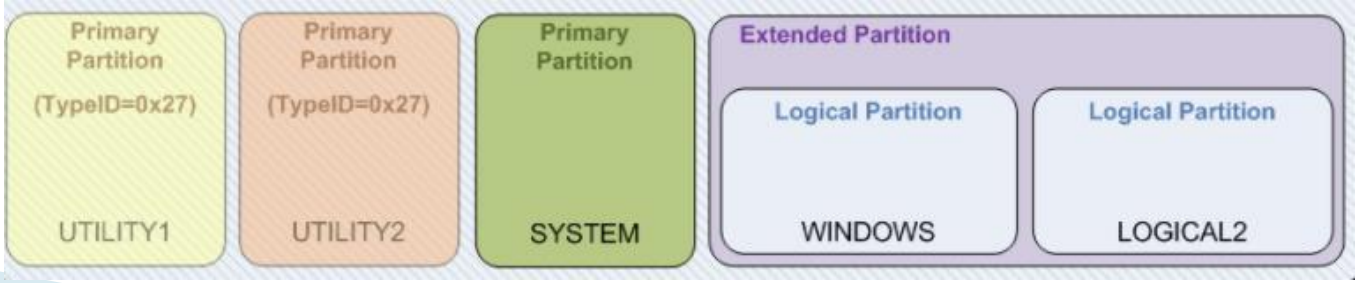
- Solid-State Drives (SSD) are more and more common nowadays.
- Unlike Hard Disks (HD), SSDs use a type of memory called “flash memory,” which is similar to RAM, but the data on an SSD persists even when it loses power.
- SSDs use a grid of electrical cells to quickly send and receive data. These grids are separated into sections called “pages,” and these pages are where data is stored. Pages are clumped together to form “blocks.”
- SSDs are called “solid-state” because they have no moving parts.



2. Partitions

Remember: Disk partitioning is the creation of one or more regions on a hard disk or other secondary storage, so that an operating system can manage information in each region separately.

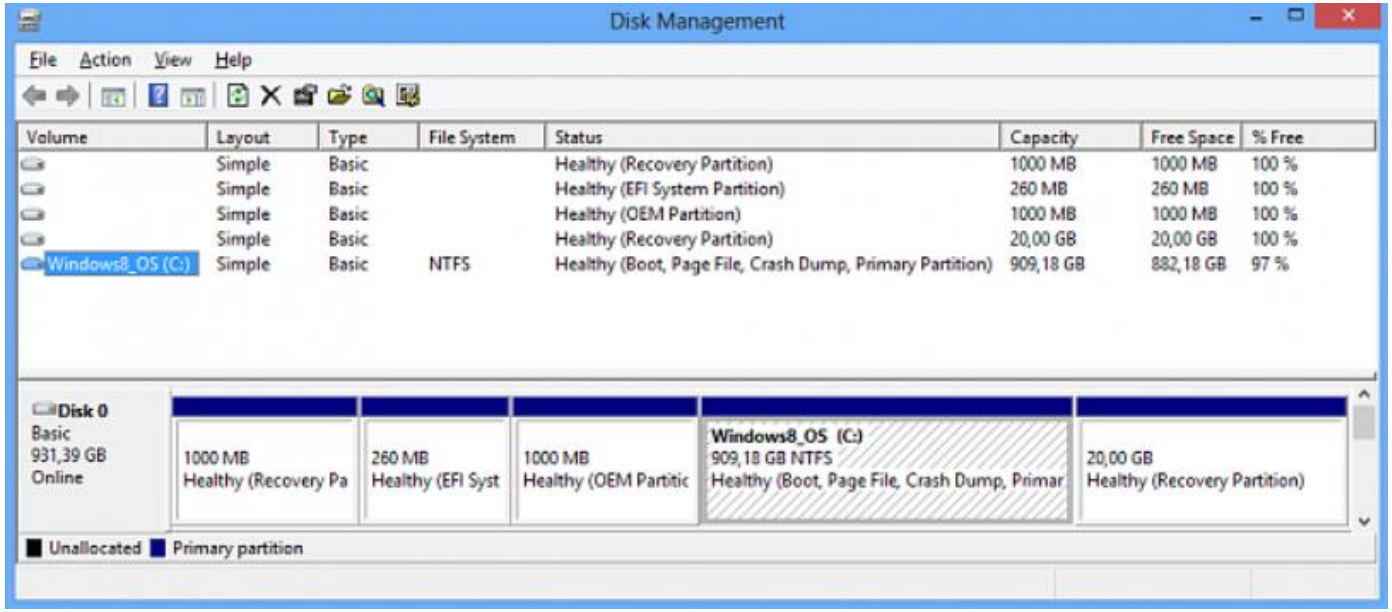
- Primary Partition is where both the OS and other data can be stored, and it is the only partition that can be set **active**. Whichever primary partition from which you load your OS at boot time becomes the active partition. You can designate only one partition active at a time.
- Extended Partition is a work-around used to extend the original method of partitioning a disk which only allowed dividing a disk into a maximum of four partitions. The extended partition can be subdivided into multiple logical partitions.
- Logical Partition is the hard disk partition created in extended partition. Like primary partition, a logical partition can be used to install an OS and any other types of files, but we are unable to set it active.



2. Partitions

Other partitions

- Hidden partition is a special section set aside on manufacturer computers to hold the information used to restore your computer back to its factory settings. In Windows, OEM Partition allows to store a system restore partition in the event you need to reinstall the OS.
- Recovery Partition is most likely for saving periodic system restore points and other backup data.
- Boot partition is a primary partition that contains the boot loader, a piece of software responsible for booting the operating system.



2. Partitions

Nomenclature

- Windows: Partitions can be identified by their names (such as "Windows7_OS") and their letters (such as "C:" or "D:").

- Linux: By convention, IDE and SATA drives will be given device names (/dev/hd for IDE Hard Drives or /dev/sd for SATA). SATA Hard Drive A (/dev/sda) is the first drive or SATA Hard Drive C (/dev/sdc) is the third one. Once a drive has been partitioned, the partitions will be represented as numbers on the end of the names. For example, the second partition on the second drive will be /dev/sdb2.

drive name	drive controller	drive number	partition number
/dev/sda1	SATA	1	1
/dev/sda2	SATA	1	2
/dev/sda3	SATA	1	3
/dev/sda4	SATA	1	4
/dev/hdb1	IDE	2	1
/dev/hdb2	IDE	2	2
/dev/hdb3	IDE	2	3
/dev/hdb4	IDE	2	4

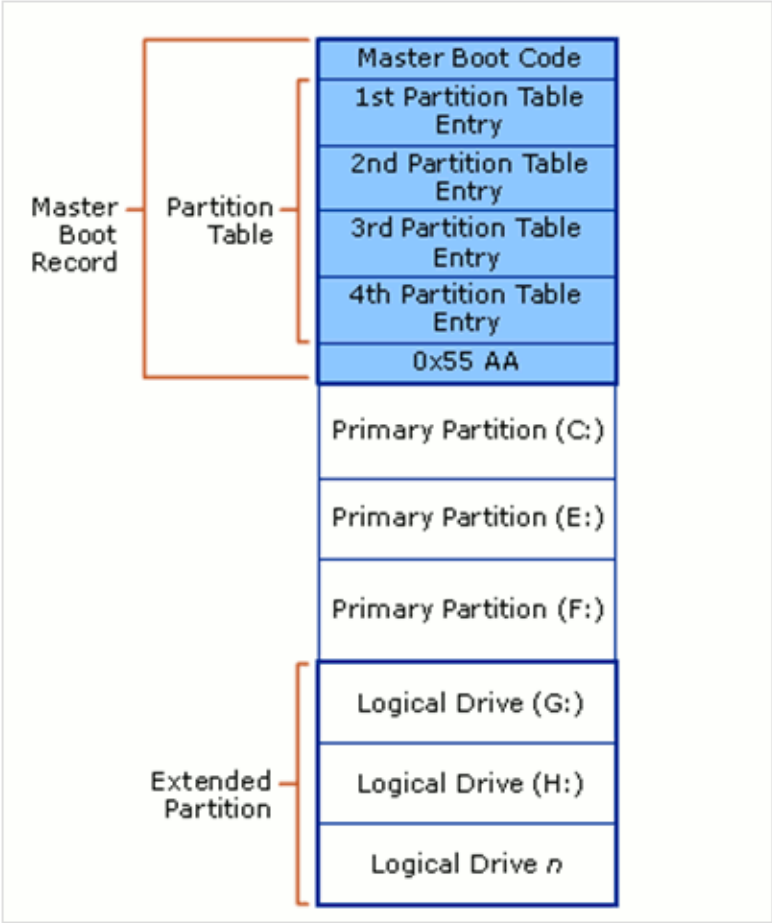
3. MBR vs GPT

Master Boot Record (MBR)

- MBR is the old standard for managing the partition in the hard disk, and it is still being used extensively by many people.
- The MBR resides at the very beginning of the hard disk and it holds the information on how the logical partitions are organized in the storage device.
- For a MBR disk, you can only have four primary partitions. To create more partitions, you can set the fourth partition as the extended partition and you will be able to create more sub-partitions (or logical drives) within it.
- As MBR uses 32-bit to record the partition, each partition can only go up to a maximum of 2TB in size
- The MBR is the only place that holds the partition information. If it ever get corrupted (and yes, it can get corrupted very easily), the entire hard disk is unreadable.

3. MBR vs GPT

Master Boot Record (MBR)



3. MBR vs GPT

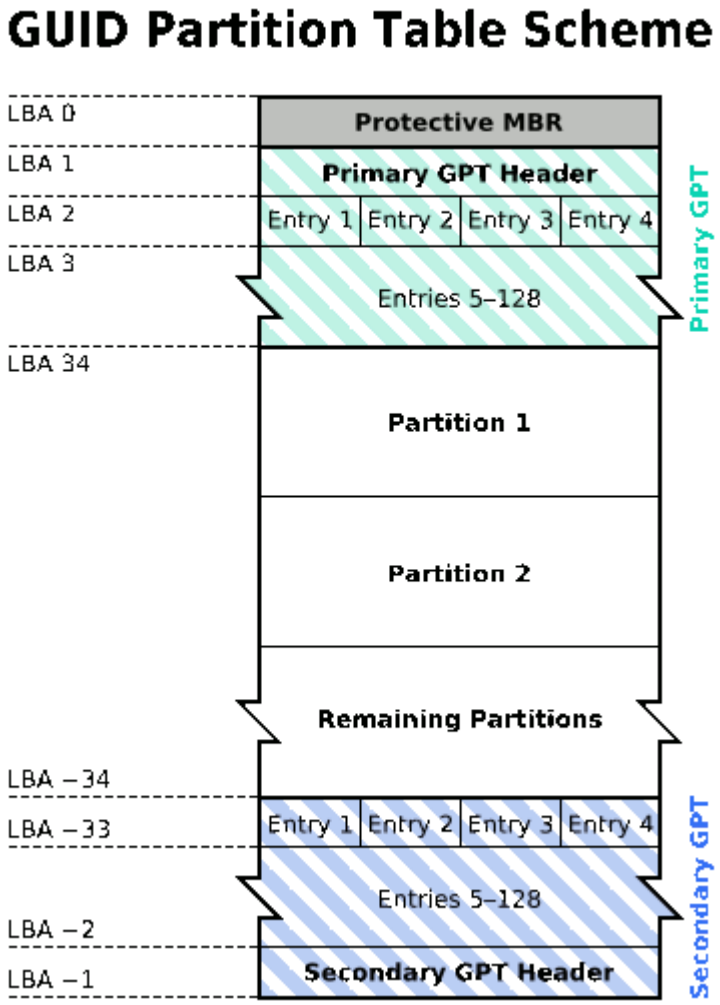
GUID Partition Table (GPT)

- GPT is the latest standard for laying out the partitions of a hard disk. It makes use of globally unique identifiers (GUID) to define the partition and it is part of the UEFI standard.
- This means that on a UEFI-based system (which is required for Windows 8 or Windows 10 Secure Boot feature), it is a must to use GPT.
- With GPT, you can create theoretically unlimited partitions on the hard disk, even though it is generally restricted to 128 partitions.
- Unlike MBR that limits each partition to only 2TB in size, each partition in GPT can hold up to 2^{64} blocks in length (as it is using 64-bit), which is equivalent to 9.44ZB for a 512-byte block (1 ZB is 1 billion terabytes).
- In Microsoft Windows, that size is limited to 256TB.

3. MBR vs GPT

GUID Partition Table (GPT)

- From the GPT Table Scheme diagram on the right, you can see that there is a primary GPT at the beginning of the hard disk and a secondary GPT at the end.
- This is what makes GPT more useful than MBR. GPT stores a backup header and partition table at the end of the disk so it can be recovered if the primary tables are corrupted.
- You can also see that there is a protective MBR at the first sector of the hard disk. Such hybrid setup is to allow a BIOS-based system to boot from a GPT disk using a boot loader stored in the protective MBR"s code area.



3. MBR vs GPT

Which one is better?

- Usually, MBR and BIOS (MBR + BIOS), and GPT and UEFI (GPT + UEFI) go hand in hand.
- This is compulsory for some systems (i.e. Windows), while optional for others (i.e. Linux).
- If a Windows computer that uses UEFI, it will only support GPT. So when you upgrade your computer, it may be necessary to convert MBR to GPT at first.
- A good way to convert MBR to GPT is using AOMEI Partition Assistant, among others.

4. Volumes

- A **volume** is a single accessible storage area with a single file system, typically (though not necessarily) resident on a single partition of a hard disk.
- A volume is not the same thing as a partition.
- For example, consider a 1 TB hard drive that has been divided into four 250 GB partitions. The first two partitions were formatted with standard Mac file systems; the third partition was formatted with a Windows file system; and the final partition was either never formatted, or was formatted with a file system that the Mac does not recognize.
- The Mac will see the two Mac partitions and the Windows partition (because the Mac can read Windows file systems), but it will not see the fourth partition. It is still a partition, but it is not a volume.

4. Volumes

- In Windows, the term volume is used to refer to the concept of a disk partition formatted with a valid file system, most commonly NTFS, that is used by the Windows operating system to store files (it is called basic disk). But, there is another type of disk when referring to storage types in this context: dynamic disks, which provide features that basic disks do not, such as the ability to create volumes that span multiple disks and the ability to create fault-tolerant volumes (mirrored and RAID-5 volumes).

- The term volume in Linux is related to the Logical Volume Manager (LVM), which can be used to manage mass storage devices. A physical volume is a storage device or partition. A logical volume created by the LVM is a logical storage device which can also span multiple physical volumes.

5. Bootloaders

- Alternatively referred to as bootstrap loader, a **bootloader** is a program that resides in the computer"s ROM or other non-volatile memory. It is automatically executed by the processor when turning on the computer.
- The bootloader reads the hard drives boot sector to continue the process of loading the computer"s operating system.
- It is stored in the master boot record (MBR) on the computer"s hard drive. When the computer is turned on or restarted, it first performs the power-on self-test, also known as POST. If the POST is successful and no issues are found, the bootstrap loader will load the operating system for the computer into memory. The computer will then be able to quickly access, load, and run the operating system.
- The bootstrap loader has been replaced in computers that have an Extensible Firmware Interface (**EFI**) and is now part of the EFI BIOS.

5. Bootloaders

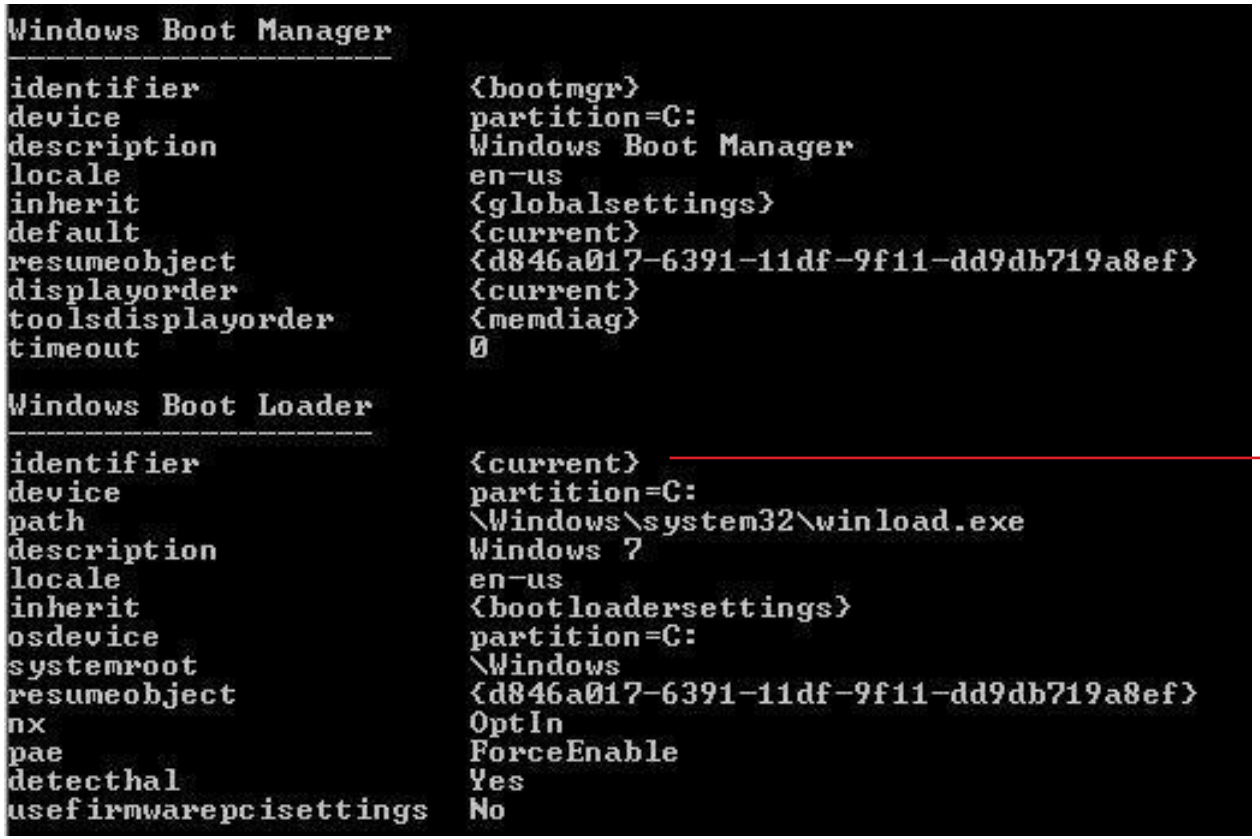
- Most common bootloaders:
  - **Windows XP and previous systems:** NTLDR and boot.ini configuration file
  - **Windows Vista and Windows 7:**
    1. Windows BCD (Boot Configuration Data).
    2. It allows to boot operating systems that do not use BIOS.
    3. Windows Boot Manager (bootmgr) is a database which replaces boot.ini.
    4. We can modify the bootloader with BCEDIT command.
  - **Windows 8, 8.1 and 10:** UEFI Secure Boot.
  - **Linux:**
    1. LILO
    2. GRUB (Grand Unified Bootloader). Configuration saved in /boot/grub/menu.lst and /boot/grub/grub.conf
    3. GRUB2 uses the files /etc/default/grub and /boot/grub/grub.cfg

5. Bootloaders

- When you boot your computer, the BIOS loads the boot loader from the hard drive and the boot loader boots the installed operating system.
- There"s no limit to the number of operating systems you can install. Having two operating systems installed and choosing between them at boot time is known as **dual-booting.**
- When you install a new operating system, the bootloader is replaced by the corresponding one of such a system.
- Some recommendations:
  - From oldest to newest.
  - First Windows systems, then Linux. Alike Windows BCD, GRUB or GRUB2 can detect Windows file systems.

5. Bootloaders (BCDEDIT)

- The BCD registry file replaces the Boot.ini files used in Windows XP and earlier versions of Windows to track operating system location



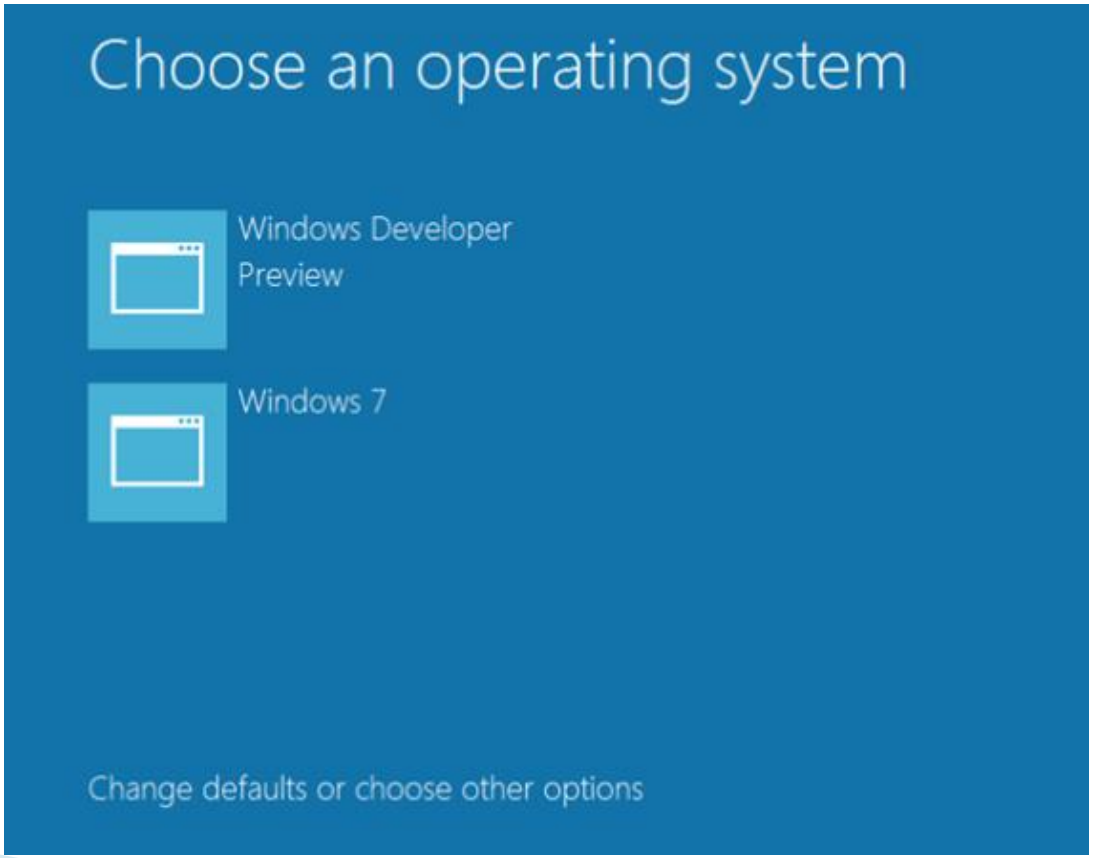
Default OS

5. Bootloaders (BCDEDIT)

- Some command examples:
  - **bcdedit /export C:\SAVEDBCD**. This will create a file C:\SAVEDBCD which is your boot entry backup.
  - **bcdedit /import c:\savedbcd**. To undo changes, we can import our backup.
  - **bcdedit /set {current} description "My edited Windows Boot Entry"**. This changes the title of the boot menu entry "{current}"
  - **bcdedit /set {ntldr} device partition="E:"** NOTE: This tells BCD that Windows XP partition is drive E:
  - **bcdedit /default {ntldr}** This places Windows XP as the default OS to boot first with.
  - **bcdedit /displayorder {33342343-3424-2342342342-2344} /addlast** This tells bcd that the boot entry with UUID 3334... should be the last entry on the menu.

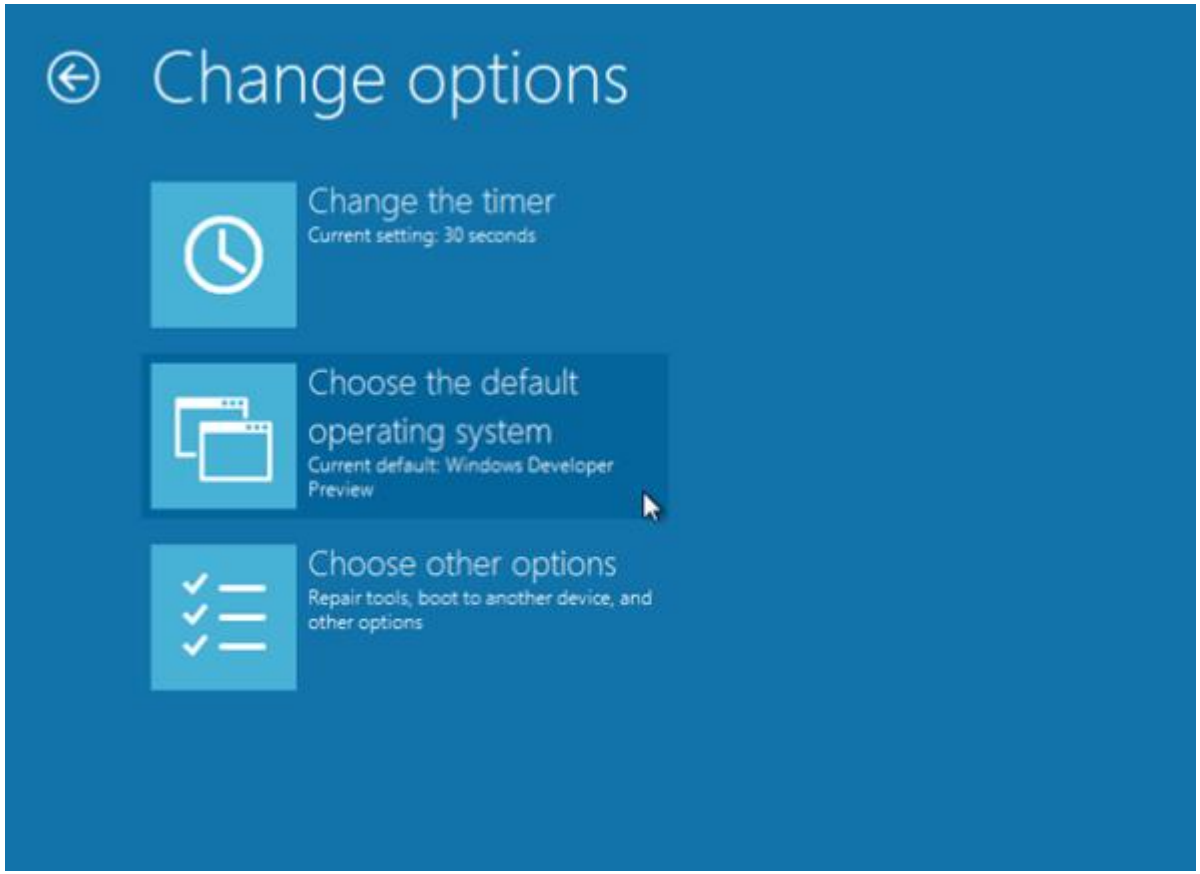
5. Bootloaders (BCDEDIT)

- Do you find BCDEDIT easy to deal with?
- Since Windows 8, we have a graphical option to manage the different operating systems when we are going to boot.



5. Bootloaders (BCDEDIT)

- We can change the default operating system and other booting options

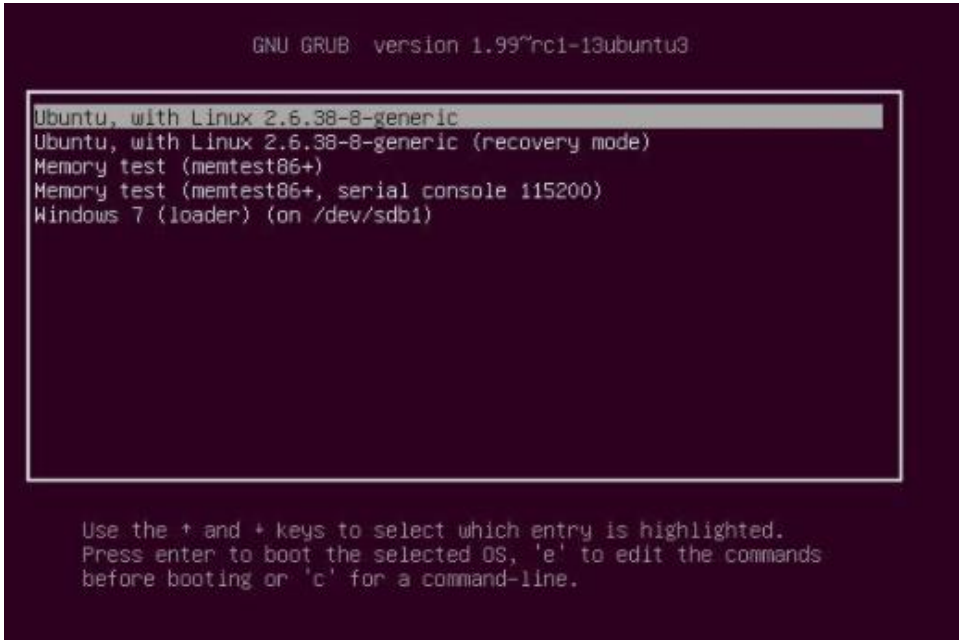


5. Bootloaders (GRUB2)

- To determine your version, use grub-install.
- GRUB 2 menu will display only if another operating system is detected.
- Configuration changes are normally made to the **/etc/default/grub** file and to the custom scripts located in /etc/grub.d
- No changes are made to the GRUB 2 menu until the **update-grub** command is run as root.
- This command runs the GRUB 2 configuration scripts and updates the **/boot/grub/grub.cfg** file (where changes in the bootloader are actually saved).
- This way, we do not update directly /boot/grub/grub.cfg
- The default entry is determined by the GRUB_DEFAULT= setting in /etc/default/grub; the first "menuentry" has a value of "0".

5. Bootloaders (GRUB2)

- The time the screen remains blank but available for display is determined by a setting in /etc/default/grub (GRUB_HIDDEN_TIMEOUT)
- GRUB 2 can display a countdown timer to provide visual feedback on the time remaining until the default selection is chosen. The timeout setting is enabled in /etc/default/grub (GRUB_HIDDEN_TIMEOUT_QUIET)
- Once the menu displays, the GRUB_TIMEOUT counter begins. Pressing "ESC" stops the countdown.



5. Bootloaders (GRUB2)

- **GRUB_DEFAULT=X.** Sets the default menu entry by menu position. The first "menuentry" in grub.cfg is 0, the second is 1, etc.
- **GRUB_TIMEOUT=X.** Sets the time period in seconds for the menu to be displayed before automatically booting unless the user intervenes.
- **GRUB_HIDDEN_TIMEOUT=0.** No menu is displayed. The system is immediately booted to the default OS. This is the default setting with only one identified operating system.
- **GRUB_HIDDEN_TIMEOUT=X**. X is a positive integer (e.g. 1, 5, 10, etc). The boot process will pause and display a blank screen or the designated splash image for X seconds. At the end of the time period, the system will boot. No menu will be displayed. While GRUB_HIDDEN_TIMEOUT is active, the menu can be displayed by pressing "ESC". To use this GRUB_HIDDEN_TIMEOUT, it is required to have a value of 0 for the GRUB_TIMEOUT parameter.
- **GRUB_HIDDEN_TIMEOUT_QUIET=true**. Determines whether a countdown timer is displayed on a blank screen when using the GRUB_HIDDEN_TIMEOUT feature.
  - true No countdown is displayed. The screen will be blank.
  - false A counter will display on a blank screen for the duration of the GRUB_HIDDEN_TIMEOUT value.

5. Bootloaders (GRUB2)

- **GRUB_HIDDEN_TIMEOUT** and **GRUB_HIDDEN_TIMEOUT_QUIET** are deprecated in favour of the less confusing **GRUB_TIMEOUT_STYLE**
- **GRUB_TIMEOUT_STYLE:** If not specified, the default value will be "menu".
- If this option is unset or set to menu, then GRUB will display the menu and then wait for the timeout set by "GRUB_TIMEOUT" to expire before booting the default entry. Pressing "ESC" interrupts the timeout.
- If this option is set to countdown or hidden, then, before displaying the menu, GRUB will wait for the timeout set by "GRUB_TIMEOUT" to expire. If ESC is pressed during that time, it will display the menu and wait for input. If the timeout expires before either of these happens, it will boot the default entry. In the countdown case, it will show a one-line indication of the remaining time.
- **GRUB_TIMEOUT_STYLE**=menu is equivalent to hide **GRUB_HIDDEN_TIMEOUT**.
- **GRUB_TIMEOUT_STYLE**=hidden is equivalent to **GRUB_HIDDEN_TIMEOUT**=X and **GRUB_HIDDEN_TIMEOUT_QUIET**=true
- **GRUB_TIMEOUT_STYLE**=countdown is equivalent to **GRUB_HIDDEN_TIMEOUT**=X and **GRUB_HIDDEN_TIMEOUT_QUIET**=false

5. Bootloaders (GRUB2)

- All the settings related to hide the GRUB menu are restricted by default.
- To allow this configuration you need to edit the file called **/boot/grub/grub.cfg**, which is not recommended.
- Once opened this file, it is required to comment using the character "#" some of the last lines.

```
set timeout_style=menu
if [ "${timeout}" = 0 ]; then
    set timeout=10
fi
```

- The main disadvantage is that for safety reasons you need to always manually comment these lines after running "update-grub".

- This only works **from Ubuntu 19.04**

5. Bootloaders (GRUB2)

In previous versions to Ubuntu 19.04.

- You can edit the file called **/etc/grub.d/30_os-prober**
- Once opened this file, it is required to comment the last line using **"# adjust_timeout"**
- In this case, unlike using Ubuntu 19.04, you do not need to modify the **/etc/grub.d/30_os-prober** again after running update-grub
- But it means that is a less safe alternative.

5. Bootloaders (GRUB2)

- For all the versions of Ubuntu, another option is to use the parameter "**GRUB_DISABLE_OS_PROBER=true"** in /etc/default/grub

- But, this parameter with value of TRUE is also set to avoid discovering other operating systems installed on the same system and generate appropriate menu entries for them. So, despite showing the menu, we will not show Windows entries if GRUB_DISABLE_OS_PROBER is activated. We can do it adding manually the entries in **/boot/grub/grub.cfg**, but it is not recommended.

- As a result, it is easier to comment the corresponding lines, either of **/boot/grub/grub.cfg** or **/etc/grub.d/30_os-prober**, depending on the operating system version you are using.