



Centro Oficial FP
Digital & Tech

0483 Sistemes informàtics

Procés d'arrencada d'un sistema operatiu

Procés d'arrencada d'un sistema operatiu

- Arranc inicial - POST
- MBR
- GPT - EFI
- Procés d'arrancada de Windows
 - Administrador d'arrencada de Windows
- Procés d'arrancada de Linux
- Errors d'arranc



Arrencada inicial POST

El procés de càrrega d'un sistema operatiu qualsevol es compon per un seguit de passes que s'inicien quan s'encén o es reinicia l'ordinador:

1. Quan es dona tensió a la font d'alimentació i una vegada que l'alimentació s'estabilitza, genera un senyal anomenat "**Power Good**" en un dels cables que va de la font d'alimentació a la placa base; aquest senyal és rebut en el joc de xips instal·lats a la placa, i al seu torn generen **un senyal de reinici (reset)** al processador. **La finalitat d'aquest procés és evitar que el processador arrenqui prematurament quan les tensions d'alimentació no són encara correctes**, el que podria produir un error de maquinari. **tema que s'utilitza per a un reinici en caler** bootó ma





Arrencada inicial POST

2. El processador arrenca quan es retira el senyal de reset. En aquest moment no existeix en memòria cap instrucció o dada, de manera que no pot fer absolutament res. Per salvar aquest obstacle, els fabricants inclouen en la circuiteria (maquinari) de la placa base un mecanisme especial. **El sistema es dirigeix a una adreça fixa de memòria ROM** (La FFFF0h en concret). Aquesta adreça, situada molt a prop del final de la memòria del sistema en els primers ordinadors compatibles, és el punt d'inici de la BIOS. en realitat aquest punt d'inici **conté una instrucció de salt (jump) que indica al processador a on ha de dirigir-se per trobar el punt on comença realment el programa de càrrega (bootstrap) de la BIOS**. Aquest programa contingut en aquesta direcció es porta a la CPU i s'executa.
3. La primera part d'aquest programa de la BIOS **inicia un procés de comprovació del maquinari denominat POST (Power On Self Test)**, en cas d'existir errors greus, el programa s'atura emetent una sèrie de xiulets que indiquen el tipus d'error trobat.



Arrencada inicial POST



4. A continuació la BIOS recorre la memòria a la **recerca de la possible existència d'altres programes en ROM** per veure si algun té BIOS, el que passa per exemple, amb els controladors de disc dur IDE / ATA, les BIOS dels quals es troben a l'adreça C8000h; altres elements que solen comptar amb les seves pròpies BIOS són les targetes de xarxa i les **controladores** SCSI. Aquests mòduls són carregats i executats.
5. A continuació, la BIOS mostra la seva pantalla principal (generalment amb els crèdits del fabricant nom de versió i data). Com hem vist, la BIOS realitza una espècie d'inventari del sistema i algunes proves per verificar que el seu funcionament és correcte, i en aquesta pantalla mostra un resum dels mateixos.

Phoenix Technologies, LTD System Configurations							
CPU Type	:	AMD Athlon(tm) XP	Base Memory	:	640K		
CPU ID	:	0681	Extended Memory	:	1047552K		
CPU Clock	:	2000MHz	L1 Cache Size	:	128K		
			L2 Cache Size	:	256K		
Diskette Drive A	:	1.44M, 3.5 in.	Display Type	:	EGA/UGA		
Pri. Master Disk	:	LBA,ATA 100,40822MB	Serial Port(s)	:	3F8 2F8		
Pri. Slave Disk	:	LBA,ATA 100,40862MB	Parallel Port(s)	:	378		
Pri. Master Disk	:	DVD,ATA 33	DDR DIMM at Rows	:	2 3 4 5		
Sec. Slave Disk	:	CHS,P10 4, 512MB					
PCI device listing ...							
Bus No.	Device No.	Func No.	Vendor/Device	Class	Device Class		IRQ
0	2	0	10DE 0067	0C03	USB 1.0/1.1 OHCI Controller		10
0	2	1	10DE 0067	0C03	USB 1.0/1.1 OHCI Controller		11
0	2	2	10DE 0068	0C03	USB 2.0 EHCI Controller		5
0	9	0	10DE 0065	0101	IDE Controller		14
0	13	0	10DE 006E	0C00	Serial Bus Controller		10
1	8	0	1106 3043	0200	Network Controller		11
1	9	0	1102 0002	0401	Multimedia Device		11

Arrencada inicial POST



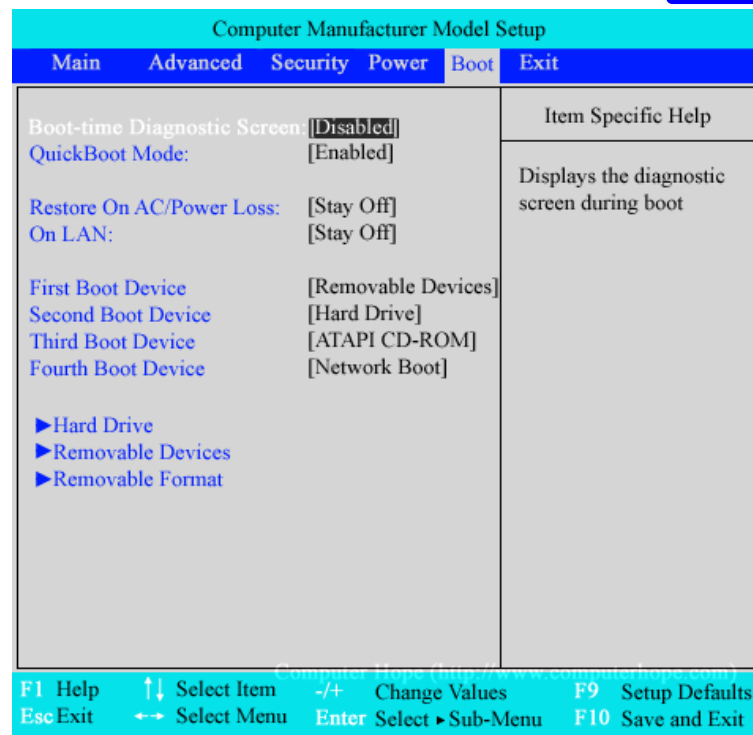
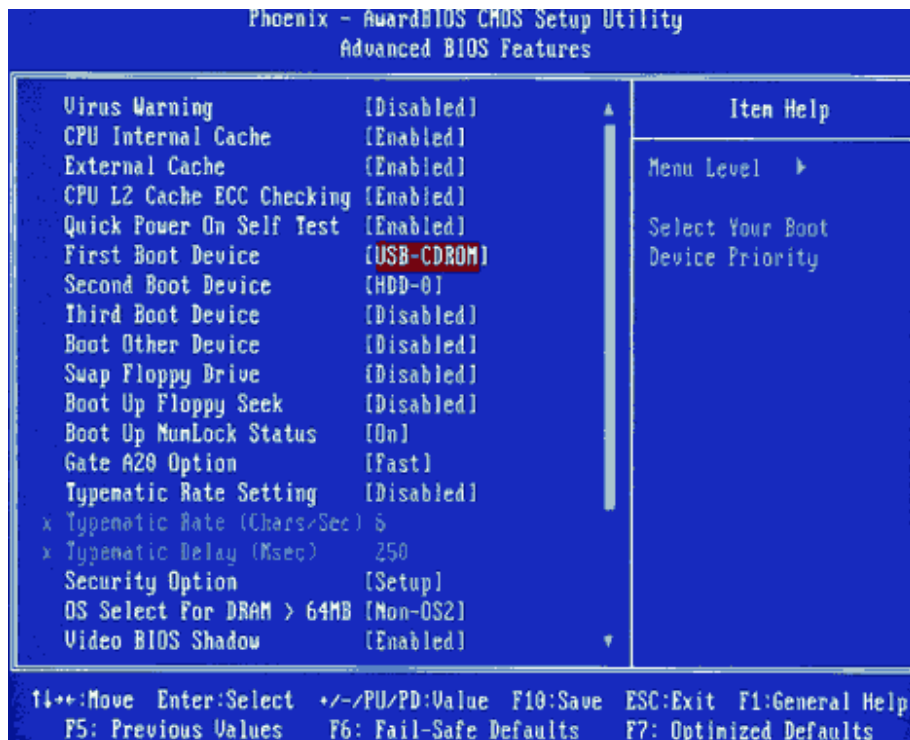
Anteriorment, la configuració del maquinari s'efectuava mitjançant interruptors ("Jumpers") situats a la placa-base. Avui dia s'utilitza l'estàndard PnP (Plug and Play), que és capaç per si sol de detectar i configurar els dispositius connectats, assignant-los els recursos necessaris i mostrant un missatge en pantalla per a cada un d'instal·lat.

L'última instrucció del programa POST s'encarrega de buscar un altre programa que pugui ser carregat en el processador perquè s'encarregui de seguir arrencant el sistema informàtic, normalment carregant ja un sistema operatiu.

Però a on buscarà el POST el programa a carregar? I en cas que hi hagi diversos sistemes operatius en diversos suports, quin d'ells serà l'escollit?

De moment, a la BIOS de gairebé tots els equips és possible trobar unes **opcions que indiquen quin és el suport d'informació des del qual s'ha d'arrencar el sistema (Boot)**.

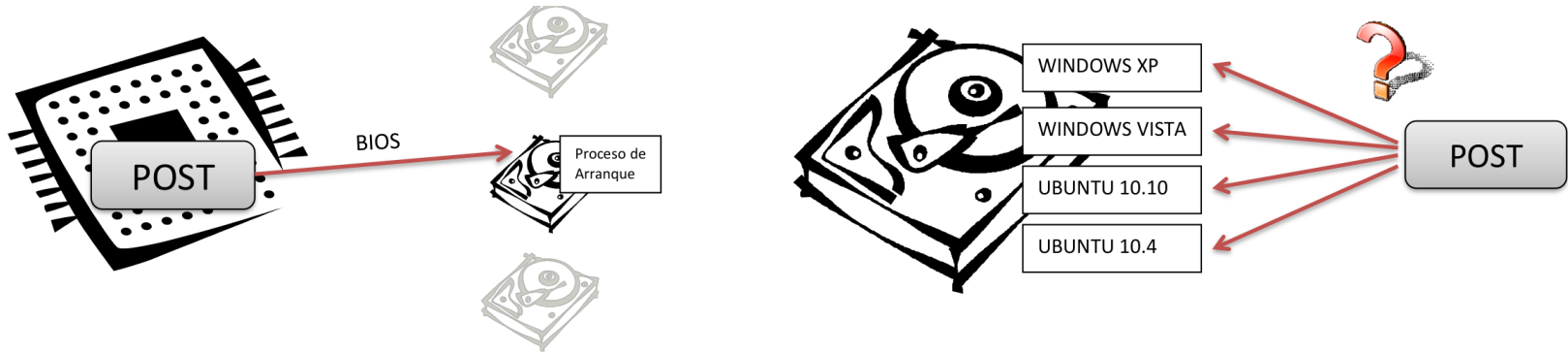
Arranc inicial POST



MBR



Si el sistema operatiu s'executa des de disquet o CD, no hi ha massa problemes, atès que en un disquet o en un CD només hi pot haver un únic procés d'arrencada per a un únic sistema operatiu.



Tanmateix, podem tenir diversos discos durs en el nostre sistema, i en cada disc podem tenir diversos sistemes operatius instal·lats. **Per a entendre el procés d'arrencada, hem de comprendre com està organitzat un disc dur.**

MBR



Les particions són divisions lògiques efectuades en un disc dur. Responen a una necessitat molt important en informàtica: compartir un mateix disc dur per a diversos sistemes operatius, per exemple. **Cada partició té l'estructura lògica corresponent al seu sistema operatiu**: una partició Windows 98 conté sector d'arrencada, FAT, directori arrel i àrea de dades, una partició de Windows amb NTFS té el seu sector d'arrencada i MFT (Master File Table), etc. Les dades d'una partició no es barregen amb les d'una altra.

En un disc dur podem tenir fins a 4 particions com a màxim. De les 4, solament una pot estar definida com a activa al mateix temps. **Aquesta partició activa serà la que carregui el sistema operatiu** quan iniciem el sistema informàtic.

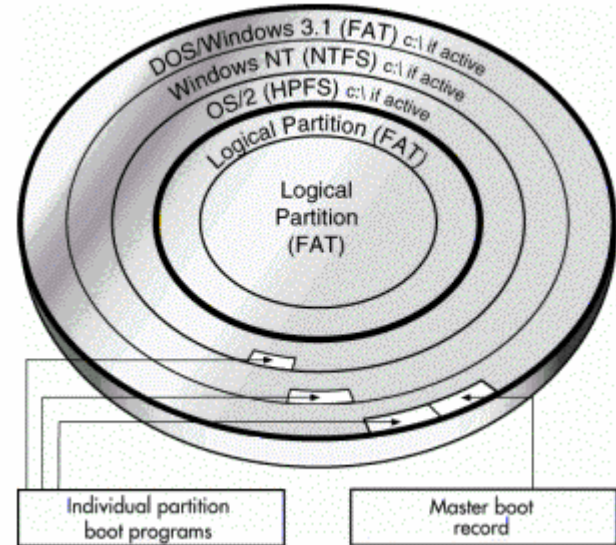
El primer sector de cadascuna d'aquestes particions es coneix com a sector d'arrencada, i en aquest sector (512 bytes) s'emmagatzema un programa especial que és l'encarregat d'arrencar el sistema operatiu de la partició.

MBR



En el primer sector del disc dur s'hi situa una taula de particions (Master Boot Record o MBR). Aquesta taula de particions inclou una taula on definim les 4 particions que poden estar presents en el nostre disc dur i la seva mida i un petit programa que permet localitzar la partició activa, llegir el seu sector d'arrencada i utilitzar-lo per engegar el nostre sistema informàtic.

Un programa MBR llegirà la taula de particions i escollirà de quina d'aquestes particions arrencarà el sistema operatiu. No ho farà com podria semblar lògic de la primera partició, sinó de la partició primària que estigui marcada com a activa. El MBR llegeix el primer sector d'aquesta partició i li cedeix el control de la CPU a aquest programa (Boot Sector o Sector d'Arrencada).

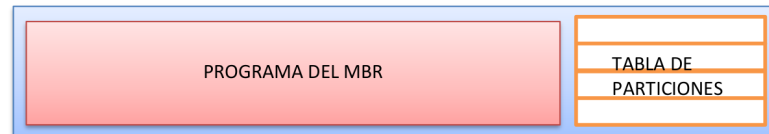


MBR

Contingut del MBR. Mida: 200h (512B). La identificació marca la posició del volum d'arranc.



Adreça	Contingut	Mida
+000h	Programa MBR	445B
+1BEh	1a entrada de la taula de particions	16B
+1CEh	2a entrada de la taula de particions	16B
+1DEh	3a entrada de la taula de particions	16B
+1EEh	4a entrada de la taula de particions	16B
+1FEh	Signatura (AA55h)	2B



MBR

Per a cada partició, s'emmagatzema el seu tipus amb 16B:



Adreça	Contingut	Mida
+00h	Estat de la partició (00h Inactiva; 80h activa)	1B
+01h	Cilindre, capçal i sector del primer sector a la partició	3B
+04h	Tipus de partició: 00h – Lliure; 01h – DOS amb l'antigaFAT de 12 bits; 02h – XENIX; 03h – XENIX; 04h – DOS amb FAT 16; 05h – Partició estesa; 06h – Partición DOS > 32 Megs; 0Bh – Windows FAT32; 0Ch – Windows FAT 32 LBA; 0Eh – VFAT; 16h – Hidden FAT 16 (Oculta); 63h – Unix; 65h – Novell Netware; etc.	1B
+05h	Cilindre, capçal i sector de l'últim sector a la partició	3B
+08h	Adreça del 1r sector de la partició (Sector d'arranc)	4B
+0Ch	Número de sectors en aquesta partició	4B

MBR

En un disc dur només pot haver-hi un **màxim de 4 particions**. No obstant això, **no una d'elles pot ser estesa**.



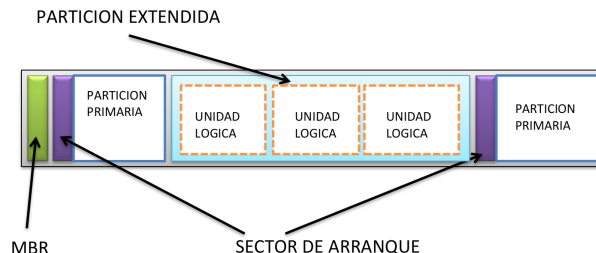
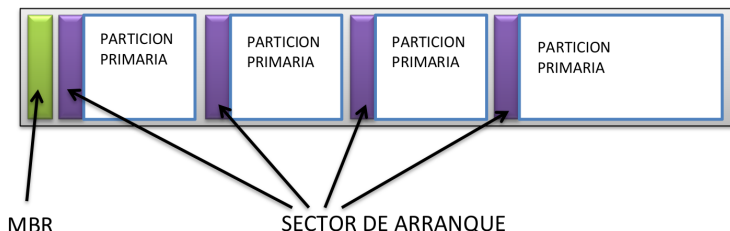
Tipus de particions:

- **Primària**

- Cada partició primària forma el seu propi volum de dades i té el seu propi sector d'arranc.

- **Estesa**

- No forma cap volum de dades **ni té un sector d'arranc** com a tal.
- És un contenidor d'unitats lògiques. Es poden crear tantes unitats lògiques com es vulguin (32 màx.). A termes pràctics, una unitat lògica es comporta com una primària.
- Utilitza el sector d'arranc per a controlar les mides, entre altres coses.





MBR

- **Només el sector d'arranc d'una primària és vàlid per a arrencar el sistema operatiu.** No podem marcar una estesa com a activa.
- **Gestió i recuperació de l'arranc:**
 - Windows
 - Reinstal·la el programa del MBR
 - FIXMBR
 - Gestiona la taula de particions
 - FDISK (Win9x)
 - Consola d'administració de discos (diskmgmt.msc)
 - diskpart.exe
 - programes de tercers
 - Linux
 - Gestiona MBR
 - testdisk, boot-repair,
 - Gestiona la taula de particions
 - fdisk
 - qtparted
 - parted

GPT (EFI)



Hem vist com funciona un disc dur amb una taula de particions MBR, que és l'opció més habitual amb què ens trobarem. No obstant això, des de fa un temps, s'està substituint les antigues BIOS per un sistema més modern conegut com a **EFI (Extensive Firmware Interface)**.

Aquest sistema, **totalment incompatible amb BIOS**, permet que en el disc dur ens oblidem de l'MBR i utilitzem un sistema molt més potent, conegut com a **GPT (GUID Partition Table)**, sent GUID acrònim de **Globally Unique Identifiers**.

GPT utilitza una moderna forma d'adreçament lògic (**LBA, logical block addressing**) en lloc del model cilindre-cap-sector (CHS) usat amb el MBR. La informació de MBR heretada està emmagatzemada en el "LBA 0" o bloc lògic 0, la capçalera GPT està en el LBA 1, i la taula de particions en si en els blocs successius. En els sistemes operatius Windows de 64-bits, 16.384B, o el que és el mateix, 32 sectors, estan reservats per a la GPT, deixant el bloc LBA 34 com el primer sector "usable" del disc.

GPT proporciona, a més, redundància. **La capçalera GPT i la taula de particions estan escrites**

tant al principi com al final del disc.

GPT (EFI)

Protective MBR

- Permet que programes i sistemes que estan preparats per a treballar únicament amb MBR puguin veure el disc com a vàlid.
- Està configurat com a una sola partició que ocupa tot el disc. L'EFI l'obvia.

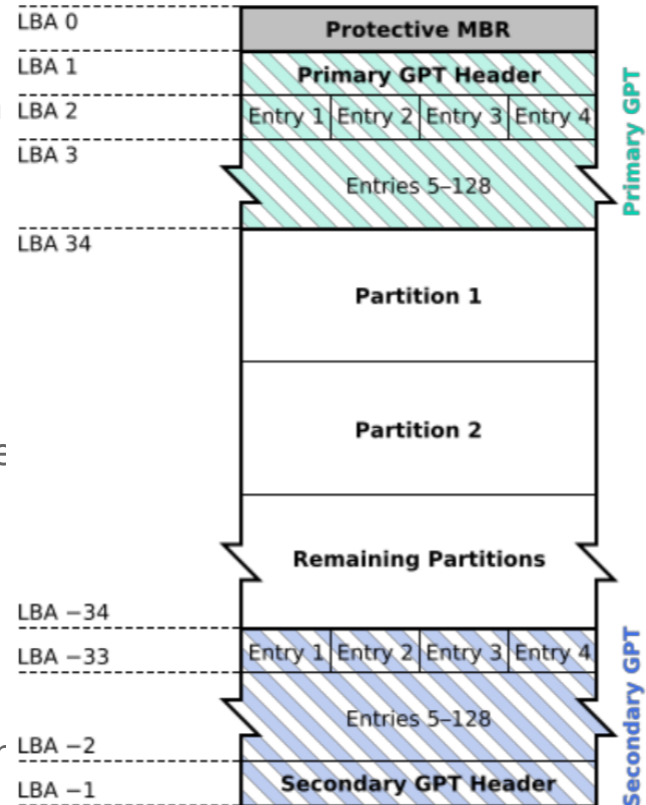
Primary GPT Header

- Defineix els blocs de disc “usables”. També defineix el número i mida de les entrades de partició de la taula de particions:
 - A Windows → 128 particions possibles.

GUID (Global Unique Identifier) del disc

- Al LBA1. Registra la seva pròpia mida i localització, i la mida i localització de la capçalera i la taula GPT secundària (a l'últim sector).

GUID Partition Table Scheme



GPT (EFI)



Desavantatges MBR

- **Només 4 particions primàries, o 3 de primàries i 1 estesa.**
- A l'MBR dins de la partició estesa, les metadades de les particions lògiques s'emmagatzemen en una estructura de llista enllaçada. Si un enllaç es perd, totes les particions lògiques es perden.
- MBR funciona amb codis de tipus de particions.
- **MBR emmagatzema la informació del sector de la partició amb valors LBA de 32b. Aquesta longitud, juntament amb els 512B de la mida del sector, limita la mida máxima de disc a 2TB.**

GPT (EFI)



Avantatges GPT

- Utilitza GUID per a identificar els tipus de particions.
- Proporciona un GUID únic de disc i un GUID únic de partició per a cada partició.
- **Nombre arbitrari de particions (depen de l'espai assignat per la taula de particions).**
 - **Fins a 128 particions a Windows. Es poden assignar encara més a Linux.**
- Utilitza 64b LBA per a emmagatzemar el nombre de sectors → mida màxima: 2ZetaBytes.
- **Redundància de l'encapçalament i de la taula de particions (a l'inici i final de disc). Facilita la recuperació en cas que els primers sectors estiguin malmesos.**
- Codi de reparació CRC32 per a detectar errors de la capçalera i taula de particions.

GPT (EFI)



Partició ESP

- La **EFI System Partition** és la partició on es guarda la informació per arrencar els diferents sistemes operatius instal·lats en un sistema.
- Es munta en iniciar l'arrencada d'un sistema amb UEFI.
- La configuració es guarda en sistemes .efi en diferents directoris.
- Exemple:
 - **EFI\ubuntu\shimx64.efi** (per Ubuntu 22.04)
 - **EFI\Microsoft\Boot\bootmgfw.efi** (per Windows 10).

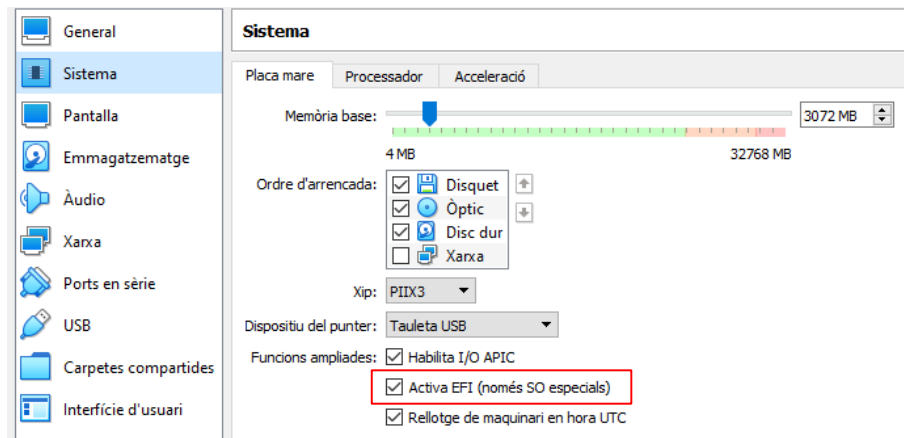
En sistemes Linux, amb la comanda **efibootmgr** es pot operar i canviar l'ordre d'inici per defecte del sistema operatiu.

GPT (EFI)



VirtualBox

- A VirtualBox, perquè un sistema utilitzi EFI en lloc de BIOS cal activar la característica “Activa EFI”:

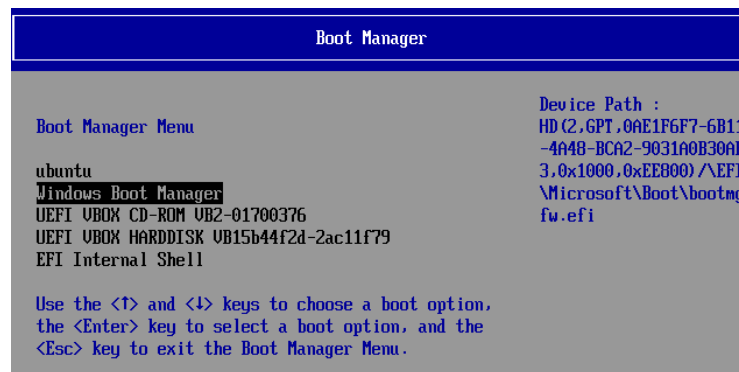
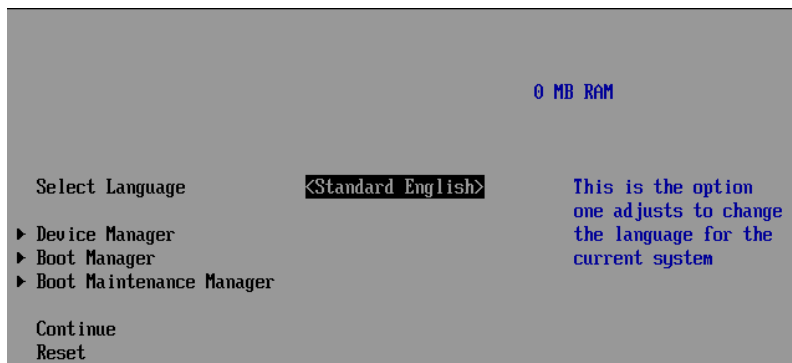


GPT (EFI)



VirtualBox

- Quan arranca el sistema, per a poder accedir al menú EFI, cal prémer la tecla **ESC**. En l'apartat “**Boot Manager**” es pot escollir quin sistema operatiu arrencar (en la imatge mostra un sistema amb Windows 10 i Ubuntu instal·lat):



Procés d'arrencada de Windows



Windows XP / 2000 / 2003

1. Es carrega i executa el **POST**
2. Es carrega l'**MBR** del disc dur
3. Es carrega el **sector d'arranc** de la partició activa
4. Es carrega el programa **NTLDR** (LoaDer de NT)
5. **NTLDR** ajusta el processador per a treballar en 32 o 64 bits.
6. **NTLDR** llegeix el fitxer **BOOT.INI** i mostra un menú si és necessari.
7. L'usuari selecciona un sistema operatiu del menú (o es carrega un per defecte)
8. **NTLDR** carrega **NTDETECT.COM**, el qual carrega la llista de hardware i retorna el control a **NTLDR**
9. **NTLDR** carrega **NTOSKRNL.EXE** (nucli o kernel del sistema), el qual llegeix el registre i procedeix a anar carregant el sistema completament.

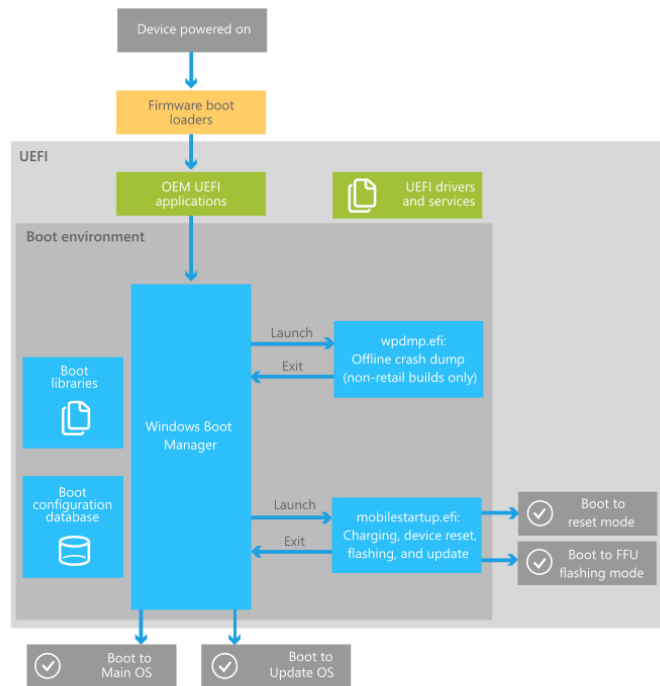
Procés d'arrencada de Windows



Windows 7 / Vista / 2008 / 8 / 10 (MBR)

1. Es carrega i executa el **POST**
2. Es carrega l'**MBR** del disc dur
3. Es carrega el **sector d'arranc** de la partició activa
4. Es carrega el programa **bootmgr**
5. **bootmgr** ajusta el processador per a treballar en 32 o 64 bits.
6. **bootmgr** llegeix la base de dades **BCD** i mostra un menú si és necessari (per a editar: **bcdedit**).
7. L'usuari selecciona un sistema operatiu del menú (o es carrega un per defecte)
8. **bootmgr** carrega **winload.exe**
9. L'arxiu **winload.exe** crea un **registre** de la configuració de maquinari. El registre és una llista de tots els ajustos, les opcions, el maquinari i el programari de l'ordinador. Winload.exe també utilitza la signatura de codi de la manera de nucli (**KMCS**, Kernel Mode Code Signing) per a **assegurar-se que tots els controladors estan signats digitalment**. Això garanteix que els controladors siguin segurs per a carregar-los a l'iniciar l'ordinador.
10. Després d'analitzar els controladors, winload.exe executa **ntoskrnl.exe**, que arrenca el nucli de Windows i configura la **HAL (Hardware Abstraction Layer)**. Finalment, la **Session Manager Subsystem (SMSS)** llegeix el registre per **crear l'entorn d'usuari, iniciar el servei de Winlogon i preparar l'escriptori de cada usuari a mesura que inicien sessió**.

Procés d'arrencada de Windows



Delivered by SoC vendor Delivered by OEM Delivered by Microsoft

Procés Actual d'arrencada amb UEFI

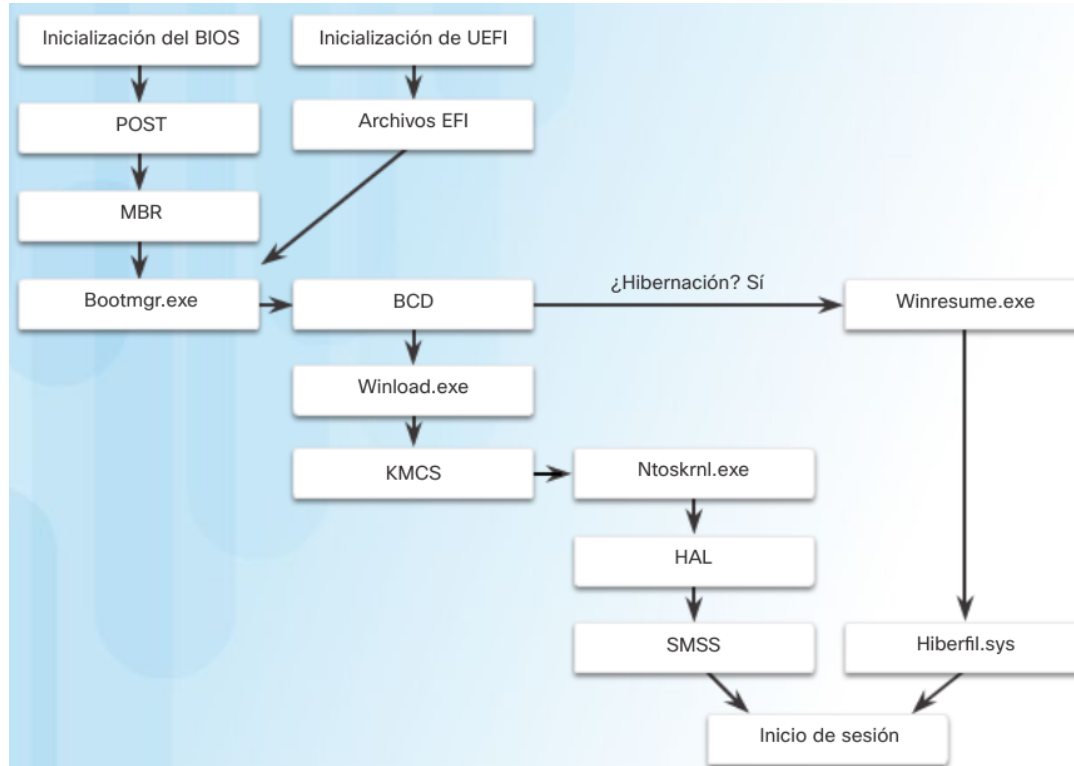
En aquest cas no hi intervé MBR

Nomenclatura:

SoC: System on Chip

OEM: Original Equipment Manufacturer.

Proces d'arrencada de Windows



Procés d'arrencada de Linux



GRUB

Linux no compta amb un sistema d'arrencada propi, sinó que permet utilitzar qualsevol gestor d'arrencada que desitgem. El que s'acostuma a incloure actualment en totes les versions de Linux és el GRUB.

El Grand Unified Bootloader (GRUB) és un gestor d'arrencada múltiple que s'utilitza habitualment per a iniciar dos o més sistemes operatius instal·lats en un mateix ordinador.

Altres gestors d'arrencada que s'utilitzen en Linux són el syslinux i el lilo.



Procés d'arrencada de Linux

Procés d'inici de GRUB (MBR)

1. La BIOS busca un dispositiu d'inici (com el disc dur) i passa el control al Màster Boot Record (MBR, els primers 512 bytes del disc dur).
2. El MBR conté la 1a fase del GRUB. Com que el MBR és petit (512 bytes), la fase 1 només s'encarrega de buscar i carregar la següent fase del GRUB (situat físicament a qualsevol part del disc dur).
3. La 2a fase del GRUB rep el control i presenta a l'usuari el menú d'inici de GRUB. Aquest menú es configura mitjançant un fitxer de text “menu.lst” o “grub.cfg” (a /boot/grub/).
4. GRUB carrega el kernel (nucli) seleccionat per l'usuari en la memòria i li passa el control per a què carregui la resta del sistema operatiu.

El **GRUB 2** substitueix el fitxer menu.lst (que s'ha d'editar manualment) per un procés modular, de manera que automàticament s'afegeixen els sistemes operatius i les opcions d'aquests.



Procés d'arrencada de Linux

Procés d'inici de GRUB (GPT)

1. La UEFI llegeix i munta la partició EFI on troba la informació dels diferents sistemes instal·lats.
2. En el cas d'Ubuntu, conté el fitxer **EFI\ubuntu\shimx64.efi**. Conté tota la informació per a poder saltar a la fase d'arrencada del GRUB.
3. A partir d'aquí GRUB rep el control i presenta a l'usuari el menú d'inici de GRUB. Aquest menú es configura “**grub.cfg**” (a **/boot/grub/**).
4. GRUB carrega el kernel (nucli) seleccionat per l'usuari en la memòria i li passa el control perquè carregui la resta del sistema operatiu.

GRUB 2 utilitza un procés modular, de manera que automàticament s'afegeixen els sistemes operatius i les opcions d'aquests.



Procés d'arrencada de Linux

FITXERS DE CONFIGURACIÓ DE GRUB

Per a configurar el gestor GRUB cal editar el següent fitxer:

- **/etc/default/grub**

I per a què els canvis s'apliquin, cal executar la comanda:

sudo update-grub

Els canvis es guarden al fitxer **/boot/grub/grub.cfg**

- **IMPORTANT:** Aquest fitxer es sobreescriu cada vegada que s'executa la comanda update-grub. Per tant, si s'hi fan canvis, es poden veure sobreescrits.



Errors d'arranc - causes

Errors hardware

A l'utilitzar un disc dur, sempre hi ha la possibilitat que es corrompin clústers d'aquest. Normalment, aquests errors no solen tenir massa importància, però si es dona la casualitat que es corromp **el primer clúster del disc dur, que és on se situa el sector del MBR i el primer sector d'arrencada de la primera partició**, ens trobarem en seriosos problemes. Normalment en aquests casos el millor és **canviar el disc dur** complet, i **intentar recuperar la informació** que existia en el disc dur.



Errors d'arranc - causes

Acció de malware (virus, troians, cucs...)

Aquestes amenaces poden esborrar el MBR i els sectors d'arrencada i antigament hi havia bastants virus que es dedicaven a realitzar aquestes accions. Avui en dia, i amb la “professionalització” dels desenvolupadors de malware, aquestes pràctiques han quedat relegades a l'oblit.

Culpa de l'usuari

La tercera causa, i la que acostuma a ser la culpable en el 99% dels casos, és que directament l'usuari faci malbé l'arrencada d'un sistema operatiu, simplement instal·lant-ne un segon.

Errors d'arranc - recuperació

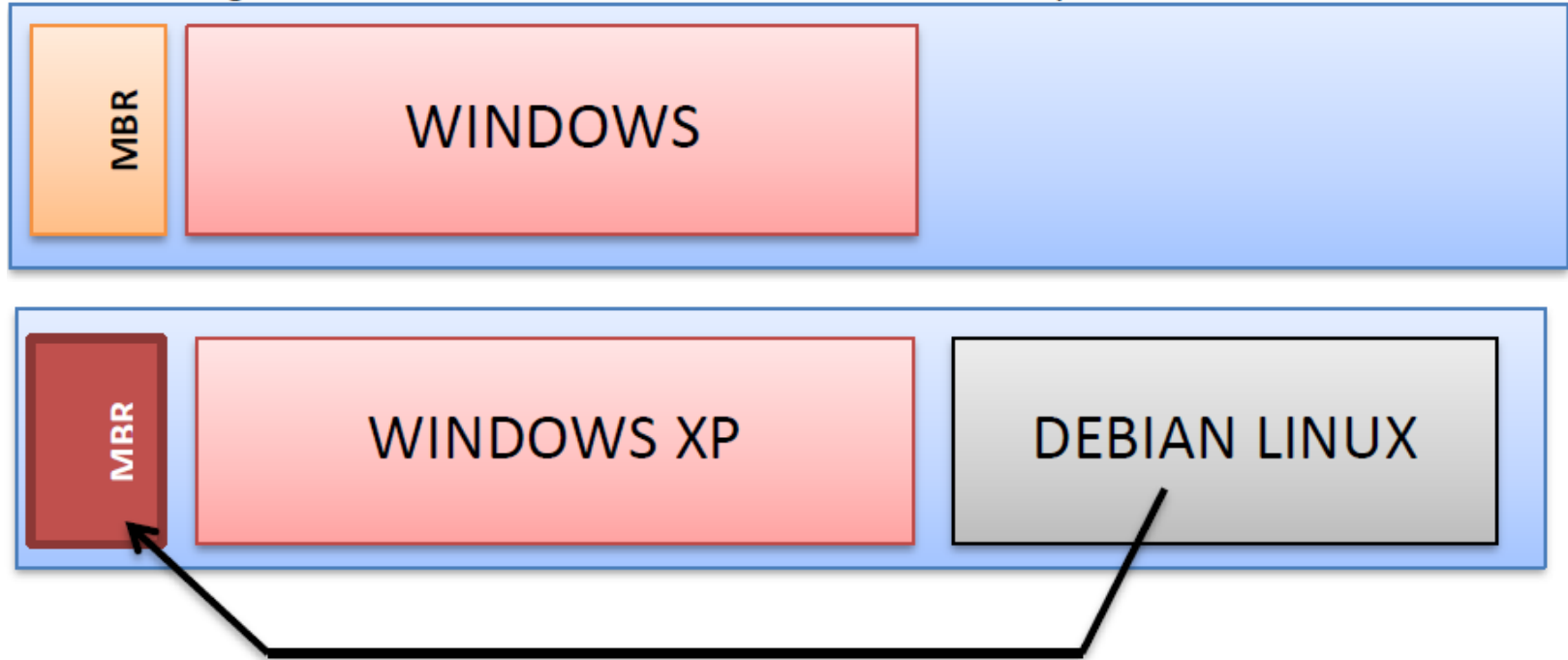


Hem vist com cada sistema operatiu disposa d'un programa per instal·lar a l'MBR, el seu propi programa per a instal·lar en el sector d'arrencada i també compten amb el seu propi gestor d'arrencada.

Està clar que si instal·lem en un mateix disc dur tres sistemes operatius diferents, cadascun d'ells haurà anat instal·lant el seu propi procés d'arrencada, però com **només pot existir un procés d'arrencada en un disc dur (només hi ha un MBR)** el procés d'arrencada que es quedi al final serà el de l'últim sistema operatiu, que “matxacarà” el procés d'arrencada del sistema operatiu anteriorment instal·lat, i així successivament.

Si 1r instal·lem Windows i després un Linux, com aquest contindrà el gestor GRUB, detectarà i permetrà iniciar ambdós sistemes.

Errors d'arranc - recuperació



Errors d'arranc - recuperació



1. El **Grub és capaç d'arrencar qualsevol sistema operatiu**, de manera que respectarà sempre (o almenys ho intentarà) qualsevol sistema operatiu que hi hagués al disc dur abans que s'instal·lés el nou sistema d'arrencada.
2. **Els gestors d'arrencada de Windows mai respectaran els Linux.** De fet, el gestor d'arrencada de Windows només és capaç d'arrencar automàticament els sistemes Windows, sent molt complicat aconseguir arrencar-ne d'altres que no siguin de Microsoft.
3. **Els gestors d'arrencada de Windows respecten els sistemes operatius Windows però només els anteriors a l'esmentat Windows.** És a dir, Windows 8 reconeix i respecta el Windows 7, però no al contrari, ja que quan es va crear el gestor d'arrencada del 7, el sistema operatiu Windows 8 no existia i, per tant, aquest gestor d'arrencada no el reconeixerà com un SO legítim i es negarà a arrencar-lo de forma automàtica.

Errors d'arranc - recuperació



Linux

En aquest cas, es pot iniciar el sistema des d'un CD especial per a recuperació del grub. També podem recuperar el sistema arrencant des d'un CD d'una distribució "live" i utilitzar alguna de les moltes eines per a recuperar el GRUB.

Windows XP

En el cas de Windows XP, cal iniciar el sistema des del CD original d'instal·lació de Windows XP. En el procés d'instal·lació que s'executarà, cal arribar fins al punt en què ens permet executar la "*consola de recuperación*". En aquesta consola podrem executar des de la línia d'ordres les següents ordres:

FIXMBR - Instal·la el carregador d'arrencada de XP al MBR.

FIXBOOT - Recupera el sector d'arrencada de Windows XP.

També hi ordres per recuperar la llista de sistemes operatius que apareixen al

Errors d'arranc - recuperació



Windows 7 / 8 / 10

Hem d'iniciar el sistema des del CD original d'instal·lació de Windows. Arribarà un moment en què el mateix programa d'instal·lació ens donarà l'opció de realitzar una reparació automàtica de l'inici de Windows. Triem aquesta opció i comprovem si el sistema és capaç de reparar automàticament. Si aquest procés falla (cosa bastant probable) tornem a iniciar el sistema des del CD, però aquesta vegada, des del menú avançat, escollim l'opció de consola de recuperació o línia d'ordres. Des d'allà, podem executar les següents ordres:

Bootrec.exe /fixmbr -Instal·la el carregador d'arrencada de Windows en el MBR.

Bootrec.exe /fixboot - Recupera el sector d'arrencada de Windows.

Errors d'arranc - recuperació



Problemes d'arranc amb UEFI i BIOS

UEFI (Unified Extensible Firmware Interface) és una interfície de firmware estàndard per a PC, dissenyada per a substituir BIOS (Basic I/O System). És un estàndard creat per més de 140 companyies tecnològiques que formen part del consorci UEFI, en el qual s'inclou Microsoft. S'ha **dissenyat per millorar la interoperabilitat del programari i solucionar les limitacions del BIOS**. Algunes dels avantatges que ofereix el firmware UEFI són:

- Ajudar a protegir el procés previ a l'inici davant d'atacs de bootkit.
- Temps d'inici i represa des de la hibernació més ràpids
- Compatibilitat amb unitats de disc dur amb particions de més de 2,2 TB.
- Compatibilitat amb controladors de dispositius de 64 bits.
- Capacitat per utilitzar Secure Boot.

Errors d'arranc - recuperació



Problemes d'arranc amb UEFI i BIOS

UEFI és el firmware que ara mateix podem trobar en els PC comercials, és molt estrany trobar un equip que no ho suporti. En BIOS de tipus **UEFI únicament podem instal·lar els sistemes de 64 bits**. Els de 32 bits no es poden instal·lar en mode UEFI. **La UEFI és un BIOS molt més amigable que la clàssica BIOS** amb pantalla blava, suporta un entorn gràfic de més qualitat, multilinguatge, recollida prèvia d'aplicacions o gestió de LAN, entre moltes altres opcions.

El gran problema que ens podem trobar en l'arrencada, és que actualment podem configurar la BIOS bé en **mode compatibilitat (legacy) o en mode UEFI pur**. Un sistema operatiu que s'hagi instal·lat en mode UEFI no podrà ser carregat per un sistema que es passi a BIOS, i viceversa. Això fa que moltes vegades ens trobem amb sistemes que no arrenquen, simplement perquè no s'ha escollit la versió correcta.



Esquemes particions Windows (GPT i MBR)

Nomenclatura:

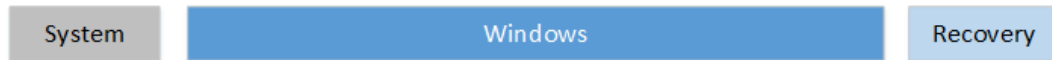
MSR: Partició reservada de Microsoft

Disk 0 default partition layout (UEFI-based PCs)



Device	Start	End	Sectors	Size	Type
/dev/sda1	2048	206847	204800	100M	EFI System
/dev/sda2	206848	239615	32768	16M	Microsoft reserved
/dev/sda3	239616	103817332	103577717	49,4G	Microsoft basic data
/dev/sda4	103819264	104853503	1034240	505M	Windows recovery environment

Disk 0 default partition layout (BIOS-based PCs)



Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sda1	*	2048	104447	102400	50M	7	HPFS/NTFS/exFAT
/dev/sda2		104448	66068596	65964149	31.5G	7	HPFS/NTFS/exFAT
/dev/sda3		66070528	67104767	1034240	505M	27	Hidden NTFS WinRE

Esquemes particions Linux (GPT MBR)

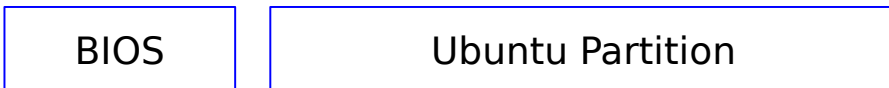


Linux UEFI - GPT



Device	Start	End	Sectors	Size	Type
/dev/sda1	2048	4095	2048	1M	BIOS boot
/dev/sda2	4096	1054719	1050624	513M	EFI System
/dev/sda3	1054720	20969471	19914752	9,5G	Linux filesystem

Linux BIOS - MBR



Dispositiu	Arrencada	Start	Final	Sectors	Size	Id	Tipus
/dev/sda1	*	2048	1050623	1048576	512M	b	W95 FAT32
/dev/sda2		1052670	41940991	40888322	19,5G	5	Estesa
/dev/sda5		1052672	41940991	40888320	19,5G	83	Linux

Actualització i manteniment del sistema i de les aplicacions

- Actualitzacions
 - Windows
 - Linux
- Instal·lació/desinstal·lació de programari
 - Windows
 - Linux

Actualització de Windows



- Utilitzarem un sistema similar per a totes les versions de Windows, tant client com servidor (Windows Update).
- Les actualitzacions normalment les descarreguem d'Internet i Windows les divideix en 2 grups: crítiques i recomanades.
- Per fer l'actualització del nostre sistema Windows cal indicar-li quan i com volem que s'actualitzi:
 - Instal·lar actualitzacions automàticament (recomanat)
 - Descarregar actualitzacions, però preguntar si les vull instal·lar
 - Buscar actualitzacions, però preguntar si les vull descarregar i instal·lar
 - No buscar mai actualitzacions (no recomanat)

Actualització de Linux (Ubuntu)



Interfície gràfica



- Quan existeixen noves actualitzacions de seguretat i noves funcionalitats, Ubuntu ens avisa a través d'un missatge emergent a través del qual podem iniciar la instal·lació (això sí, sempre com a root).
- També podem executar directament el Gestor d'actualitzacions que ens permet comprovar les actualitzacions disponibles i escollir fàcilment quines volem instal·lar.
- Ubuntu, a més, si reconeix una versió del sistema operatiu anterior, ens pot oferir actualitzar tot el nucli (kernel) del sistema de cop i no paquet a paquet.

Actualització de Linux (Ubuntu)



Línia de comandes (línia d'ordres)

- Utilitzarem l'eina **apt-get**:
 - **update** → Actualitzar els **repositoris** del sistema
 - Repositori: Arxius que contenen informació sobre el programari
 - Gestió amb interfície gràfica
 - Gestió amb línia de comandes
 - **upgrade** → Un cop actualitzats els repositoris, procedim a actualitzar els arxius disponibles del programari. L'upgrade compara les versions instal·lades en el sistema amb la dels repositoris i, si aquestes últimes són més actuals, les descarrega i les instal·la.

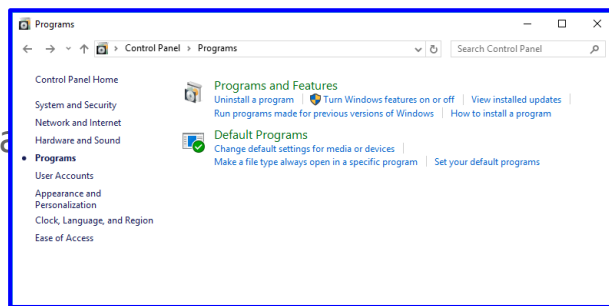


Instal·lació/desinstal·lació de programari a Windows

- La majoria d'aplicacions incorporen un Install o Setup per a poder realitzar directament la seva instal·lació. Així mateix, un cop instal·lada també incorporen un Uninstall per realitzar la desinstal·lació.
- Tot i això, a Windows el gestor d'instal·lació / desinstal·lació anomenat **“Agregar o quitar programas”** o **“Programas y características”**.

Aquestes són les seves opcions principals:

- Canviar o treure programes
- Afegir o treure components de Windows
- Configurar accés i programes predeterminats
- Instal·lar programes des de la xarxa





Instal·lació/desinstal·lació de programari a **Linux (Ubuntu)**

- Els sistemes Linux treballen amb paquets d'aplicacions, que normalment ens arriben en **formats comprimits** tipus zip, gzip, rpm, tar...
- Per a executar l'aplicació, normalment només cal descomprimir l'aplicació empaquetada en el directori que desitgem i llençar l'executable.
- Podem veure com, a diferència de Windows, realment no s'instal·len programes sinó que es descomprimeixen paquets, algun o alguns dels quals són executables que fan funcionar l'aplicació.
- Això sí, alguns paquets i/o arxius requereixen ser copiats a un determinat directori per tal del seu correcte funcionament posterior.
- A més, cal diferenciar **dos tipus de paquets** principals:
 - els **binaris**
 - els de **codi font**



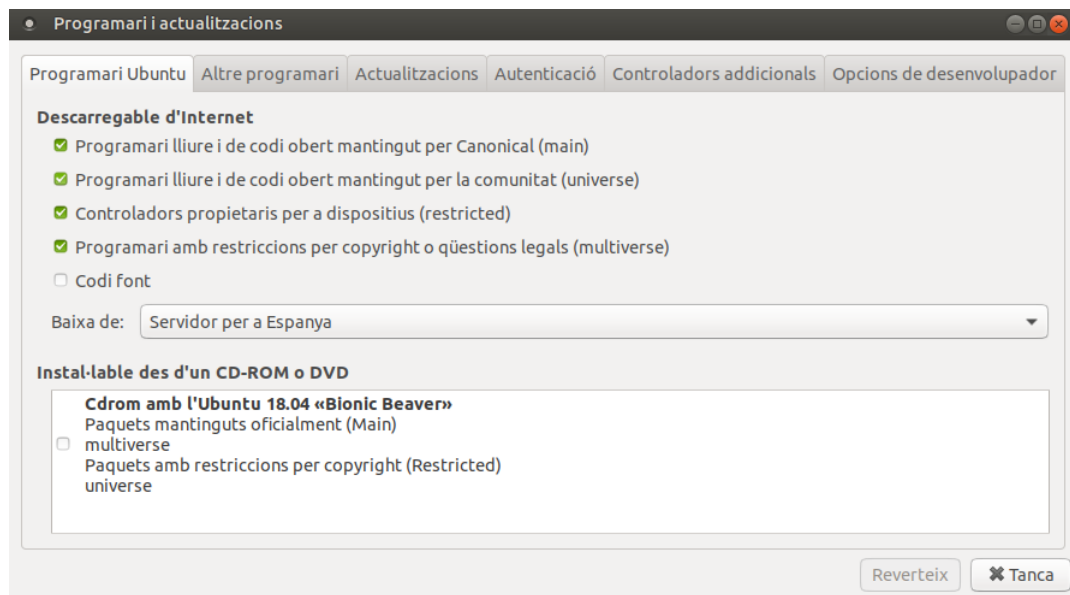
Instal·lació/desinstal·lació de programari a Linux (Ubuntu)

- Els **binaris** es caracteritzen perquè contenen codi màquina i, per tant, són dependents de l'arquitectura. A més, no tots els tipus de paquets poden ser utilitzats en totes les distribucions de Linux. Algunes extensions d'aquest tipus de paquets són rpm, deb, tgz, ebuild o src.
- Els de **codi font** contenen, en canvi, el codi font del programa, és a dir, cal compilar-los i després instal·lar-los. L'avantatge d'aquesta compilació manual és l'optimització del programa a la màquina on s'ha compilat. Per realitzar aquest procés normalment s'acompanya un arxiu README o INSTALL.
- L'últim aspecte a tenir en compte a l'hora d'instal·lar manualment paquets, és que alguns d'aquests tenen **dependències** d'altres paquets que cal tenir instal·lats prèviament. Aquestes dependències es solen indicar en algun arxiu de text inclòs al paquet i cal fer una revisió d'aquestes necessitats.



Instal·lació/desinstal·lació de programari a Linux (Ubuntu)

Interfície gràfica: Centre de Software d'Ubuntu





Instal·lació/desinstal·lació de programari a Linux (Ubuntu)

Línia de comandes: apt-get (o apt) i aptitude; dpkg

Aquests dos gestors de programari són molt similars i podem utilitzar-ne un o altre indistintivament. Ara bé, cal considerar que aptitude té una millor gestió (per exemple, pel que fa a les dependències).

- Instal·lar un paquet → `apt-get install nom_paquet`
- Instal·lar diversos paquets → `apt-get install nom_paquet1 nom_paquet2 nom_paquet3...`
- Desinstal·lar paquets → `apt-get remove nom_paquet`
- Desinstal·lar paquets i arxius de configuració → `apt-get remove --purge nom_paquet`
- Instal·lar un paquet “.deb” → `dpkg -i nom_paquet.deb`
- Desinstal·lar un paquet “.deb” → `dpkg -r nom_paquet.deb`
- Executar un programa → Situar-nos en el seu directori i `./nom_programa`