

Arrancada del sistema

Apunts d'Administració de Sistemes i Aplicacions

Jordi Mateo

Francesc Solsona

Quan engegum un ordinador, s'inicien una sèrie de processos que permeten que el sistema operatiu es carregui i estigui llest per al seu ús. Aquest conjunt de processos és el que es coneix com a arrencada del sistema (*booting*). A continuació, es detalla aquest procés pas a pas:

- **Càrrega del BIOS o UEFI.** Quan encenem l'ordinador, el primer programa que s'executa està emmagatzemat en una memòria no volàtil (NVRAM). Aquest programa pot ser el BIOS (Basic Input/Output System) en sistemes més antics o l'UEFI (Unified Extensible Firmware Interface) en sistemes més moderns. Aquest firmware és el responsable de gestionar les funcions bàsiques del sistema abans de carregar el sistema operatiu.
- **Power-On Self Test (POST).** Un cop carregat el BIOS o UEFI, es realitza una sèrie de proves conegudes com a POST (Power-On Self Test). Aquestes proves verifiquen que tots els components de maquinari essencials (com la memòria, el processador i els dispositius de xarxa) funcionin correctament. Si es detecta algun problema, el sistema pot emetre una sèrie de senyals acústics o mostrar un missatge d'error per ajudar a identificar la causa.
- **Selecció del dispositiu d'arrencada.** Un cop superat el POST, el BIOS o UEFI procedeix a seleccionar el dispositiu d'arrencada, que pot ser un disc dur, una unitat USB, un CD-ROM, o altres. L'ordre d'arrencada es pot configurar prèviament en el BIOS/UEFI o bé es pot seleccionar manualment. Un cop seleccionat el dispositiu, es busca la partició d'arrencada on es troba el gestor d'arrencada.
- **Identificació de la partició d'arrencada.** El BIOS o UEFI localitza la partició d'arrencada del dispositiu seleccionat. En sistemes amb BIOS, es fa servir el Master Boot Record (MBR), mentre que en sistemes amb UEFI es fa servir la taula de particions GUID (GPT) per identificar la partició correcta (normalment, anomenada EFI). Aquesta partició conté el gestor d'arrencada i altres fitxers necessaris per continuar el procés d'arrencada.
- **Càrrega del gestor d'arrencada.** Un cop localitzada la partició d'arrencada, el BIOS o UEFI carrega el gestor d'arrencada. Aquest programa és el responsable de carregar el sistema operatiu. Els gestors d'arrencada més comuns en sistemes Linux són GRUB (Grand Unified Bootloader) o LILO (Linux Loader). El gestor d'arrencada permet seleccionar entre diferents sistemes operatius instal·lats o diferents versions del nucli. Addicionalment, també permet configurar opcions d'arrencada i passar paràmetres al nucli del sistema operatiu.
- **Càrrega del kernel.** Un cop seleccionat el sistema operatiu, el gestor d'arrencada carrega el nucli (kernel) del sistema operatiu. El kernel és el component central del sistema operatiu que es comunica directament amb el maquinari. Per fer-ho, necessita descomprimir-se i carregar-se a la memòria RAM.
- **Muntatge del Sistema de Fitxers Inicial i Real:** El nucli primer munta un sistema de fitxers virtual (initramfs o initrd), que proporciona els recursos necessaris per muntar el sistema de fitxers real del sistema operatiu. Un cop muntat el sistema de fitxers real, el sistema està preparat per continuar amb la seva inicialització.
- **Sistema d'Inicialització:** Després de muntar el sistema de fitxers real, el kernel arrenca el sistema d'inicialització. En la majoria de distribucions Linux modernes, aquest sistema és **systemd**, però en obsoletes pot ser **initd**. El procés d'inicialització (assignat com PID 1) és el responsable de gestionar l'inici de tots els altres processos del sistema, incloent la càrrega dels serveis i daemons essencials. Aquest és el primer procés que s'executa en l'espai d'usuari i és responsable de mantenir el sistema en funcionament.
- **Execució dels targets o runlevels:** El sistema d'inicialització carrega els diferents targets o runlevels, que defineixen l'estat del sistema en un moment determinat. Aquests runlevels poden ser diferents nivells d'arrencada, com arrencada en mode gràfic o mode de recuperació. Cada runlevel pot tenir diferents serveis i daemons activats o desactivats.
- **Execució dels scripts d'arrencada:** El sistema d'inicialització executa els scripts d'arrencada definits per a cada runlevel. Aquests scripts són responsables de configurar els serveis i daemons del sistema, com ara la xarxa, els dispositius de maquinari, els sistemes de fitxers, entre altres.
- **Inici de la Sessió d'Usuari:** Finalment, el sistema està preparat per a l'usuari. Si el sistema està configurat per a un entorn gràfic, es carrega el gestor de finestres o l'entorn d'escriptori. En sistemes sense entorn gràfic, es presenta una línia de comandes. En aquest punt, l'usuari pot iniciar sessió i començar a utilitzar el sistema.

- **Execució dels scripts d'inici de sessió:** Un cop iniciada la sessió, s'executen els scripts d'inici de sessió de l'usuari. Aquests scripts poden configurar variables d'entorn, carregar programes o configurar preferències de l'usuari. Això permet personalitzar l'experiència de l'usuari i preparar l'entorn de treball.

La figura següent mostra un diagrama simplificat del procés d'arrencada del sistema:

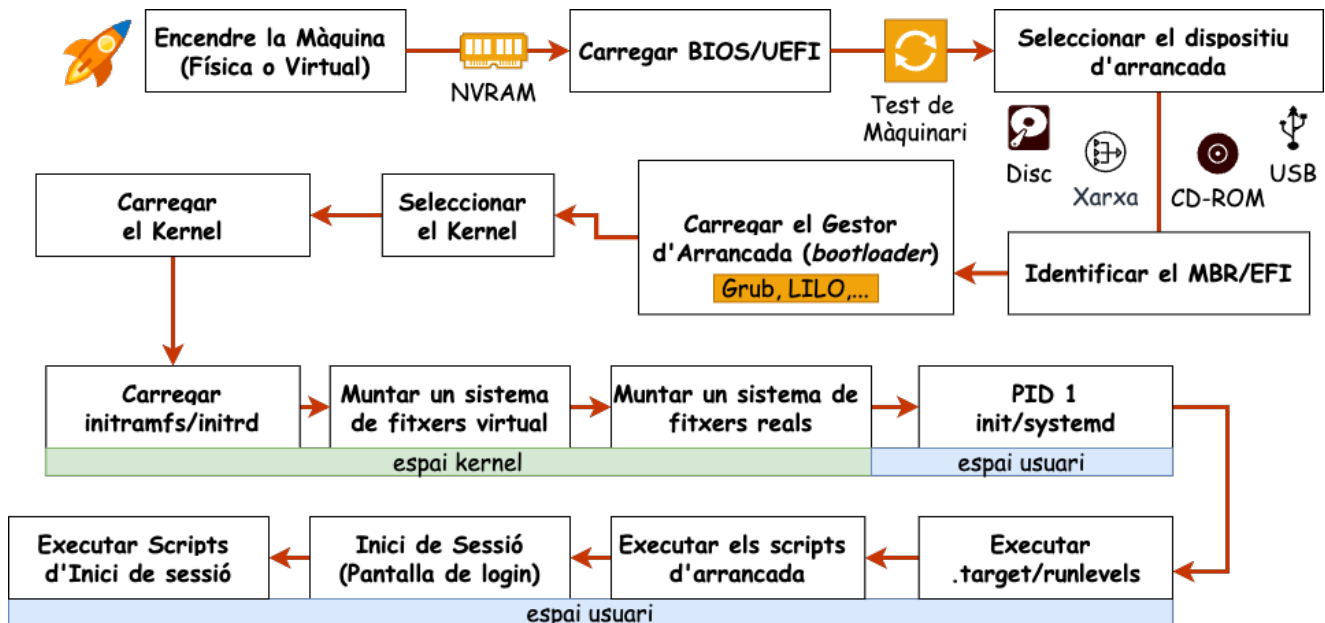


Figure 1: Arrencada del sistema

Nota:

És important destacar que, en l'actualitat, molts sistemes es troben en entorns de núvol i no segueixen un procés d'arrencada físic com el descrit anteriorment. En aquests casos, els servidors es despleguen en entorns virtuals, on la càrrega del sistema operatiu es gestiona a través de la infraestructura del núvol, sense la necessitat de BIOS o UEFI tradicionals. El procés d'arrencada en aquests entorns es pot gestionar mitjançant l'API de la plataforma de núvol o amb eines de Infraestructura com a Codi (IaC), que automatitzen la configuració i la posada en marxa dels serveis. Aquesta metodologia permet una major flexibilitat i escalabilitat, adaptant-se a les necessitats dinàmiques dels entorns moderns. Tot i això, aquests serveis virtuals tenen associades servidors físics que segueixen el procés d'arrencada tradicional.

BIOS vs. UEFI

Tradicionalment, els sistemes informàtics han utilitzat el BIOS (Basic Input/Output System) com a firmware per gestionar les funcions bàsiques del maquinari abans de carregar el sistema operatiu. El BIOS requereix un procés d'arrencada específic, basat en la taula de particions MBR (Master Boot Record).

Taula de Particions MBR

La taula de particions MBR és una estructura de dades que ocupa el primer sector d'un dispositiu d'arrencada, amb una mida total de 512 bytes. Aquesta estructura es divideix en diverses parts:

- Els primers (446 bytes): Conté el codi d'arrencada, conegut com el *bootloader* de primera etapa, que és responsable de començar el procés d'arrencada.
- Els següents 64 bytes: Emmagatzemen la taula de particions, on cada entrada ocupa 16 bytes i descriu una partició. Això limita la taula de particions MBR a un màxim de quatre particions primàries.
- Els últims 2 bytes: Contenen la signatura (0x55, 0xAA) que identifica el sector com un sector d'arrencada vàlid.

La figura [fig:MBR] mostra un esquema de l'estructura MBR que acabem de descriure.

Limitacions del BIOS i MBR

La primera limitació és l'espai dedicat al *bootloader*, que només té 446 bytes. Per tant, requereix habitualment un *bootloader* de segona etapa per carregar el sistema operatiu.

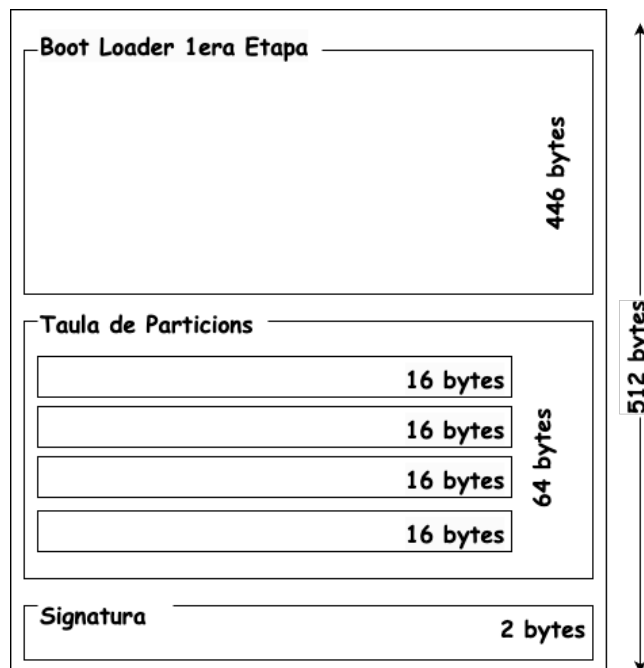


Figure 2: Esquema de l'estructura MBR

La segona limitació és que el *bootloader* de primera etapa no és prou sofisticat per llegir sistemes de fitxers estàndard. Per tant, s'ha de col·locar el *bootloader* de segona etapa en un lloc fàcilment accessible. Per exemple:

- El *bootloader* de segona etapa pot residir en l'espai no utilitzat entre l'MBR i la primera partició del disc, conegut com a “zona morta”.
- Alternativament, el *bootloader* de segona etapa pot residir en una partició marcada com a “activa”, on el *bootloader* de primera etapa llegeix i executa el *bootloader* de segona etapa des del començament d'aquesta partició.

Per garantir un arrencada correcta s'ha de garantir que tots els components BIOS, MBR i *bootloader* de primera i segona etapa estiguin correctament instal·lats i siguin compatibles.

Els *bootloaders* de segona etapa solen ser capaços de suportar múltiples sistemes operatius i sistemes de fitxers, i inclouen opcions de configuració per personalitzar el procés d'arrencada segons les necessitats de l'usuari o el sistema. Ho veurem amb més detall en les properes seccions.

UEFI i GPT

Amb l'evolució de la tecnologia, s'ha introduït l'UEFI (Unified Extensible Firmware Interface) com a reemplaçament del BIOS. L'UEFI és un firmware més modern i flexible que ofereix una sèrie d'avantatges respecte al BIOS.

Una de les millores més significatives és l'esquema de particionament de disc conegut com a GPT (GUID Partition Table). A diferència del MBR (Master Boot Record), aquest permet un nombre il·limitat de particions i suporta discos de mida superior a 2 TB i inferior a 9,4 ZB (zettabytes). A més, GPT inclou redundància per protegir les dades de partició i permet l'ús de taula de particions de reserva per a còpies de seguretat. Per facilitar la interoperabilitat amb sistemes BIOS, UEFI manté un registre compatible amb MBR al començament de cada disc. Això permet que els sistemes BIOS reconguin el disc com a formatat, tot i que no poden veure la taula de particions completa en estil GPT. És important tenir cura de no utilitzar eines administratives específiques de MBR en discos GPT, ja que podrien malinterpretar el disseny del disc.

Durant el procés d'arrencada, l'UEFI consulta la taula de particions GPT per identificar la partició d'arrencada, anomenada ESP (EFI System Partition). Aquesta partició conté el *bootloader* i altres fitxers necessaris per carregar el sistema operatiu.

A més, l'UEFI és compatible amb els sistemes de fitxers FAT (File Allocation Table), que permeten que la partició ESP sigui llegida i mantinguda per qualsevol sistema operatiu.

En un sistema UEFI, tècnicament no es requereix cap *bootloader*. L'objectiu d'arrencada UEFI pot ser un nucli UNIX o Linux configurat per a la càrrega directa UEFI, eliminant la necessitat d'un carregador de boot. Tot i així, la majoria de sistemes encara utilitzen un carregador de boot per mantenir la compatibilitat amb BIOS

antics. UEFI guarda el camí per carregar des de l'ESP com a paràmetre de configuració. Sense configuració, busca un camí estàndard, normalment `/efi/boot/bootx64.efi` en arquitectures x86_64, `efi/boot/bootaa64.efi` en arquitectures ARM.

A més, UEFI defineix APIs estàndard per accedir al maquinari del sistema, actuant com un petit sistema operatiu per si mateix. Proporciona controladors de dispositius a nivell UEFI, escrits en un llenguatge independent del processador i emmagatzemats a l'ESP. Els sistemes operatius poden utilitzar la interfície UEFI o prendre el control directe del maquinari.

Aquestes característiques fan que UEFI sigui una opció molt més avançada i flexible comparada amb el BIOS tradicional, oferint una millor gestió de l'emmagatzematge, compatibilitat amb sistemes moderns i una arrencada més segura i ràpida. Podeu consultar totes les especificacions a la documentació oficial de la darrera versió, en el moment de redacció UEFI v2.10

En resum, les principals avantatges de UEFI davant de BIOS són:

Característica	BIOS	UEFI
Capacitat de disc	Suporta fins a 2,2 TB	Suporta fins a 9,4 ZB
Interfície	Basada en text	Basada en gràfics
Seguretat	No té funcions de seguretat avançades	Suporta Secure Boot, que evita l'arrencada de codi maliciós
Compatibilitat	Limitada amb sistemes moderns	Compatible amb sistemes moderns i anteriors
Velocitat d'arrencada	Més lenta	Més ràpida

GRUB

GRUB (GRand Unified Bootloader) és el bootloader utilitzat per defecte en la majoria de distribucions Linux actuals. És un programa essencial en el procés d'arrencada, ja que carrega el nucli del sistema operatiu des del dispositiu d'emmagatzematge a la memòria RAM, permetent que el sistema operatiu s'executi.

Funcionament i Configuració de GRUB

Una de les característiques més destacades de GRUB és la seva capacitat per entendre diversos sistemes de fitxers, com ext4, XFS, Btrfs, entre altres. Aquesta habilitat permet a GRUB localitzar la partició arrel del sistema de fitxers (root filesystem) i llegir la seva configuració directament des d'un fitxer de text regular, anomenat `grub.cfg`. Aquest fitxer conté totes les instruccions necessàries per al procés d'arrencada, incloent quins nuclis estan disponibles, les opcions de línia de comandes per al nucli, i altres paràmetres de configuració.

El fitxer `grub.cfg` sol estar situat en el directori `/boot/grub`, o `/boot/grub2`. Aquest directori també emmagatzema altres recursos i mòduls de codi que GRUB pot necessitar durant l'arrencada.

Modificació i Generació del Fitxer de Configuració

Modificar la configuració d'arrencada de GRUB és tan senzill com editar el fitxer `grub.cfg`. Tanmateix, en la pràctica, no és habitual que els usuaris modifiquin aquest fitxer manualment. Això es deu al fet que moltes distribucions Linux proporcionen eines per generar automàticament el fitxer `grub.cfg` basant-se en una configuració predeterminada o personalitzada.

La utilitat més comuna per generar aquest fitxer és `grub-mkconfig`, coneguda com `grub2-mkconfig` en sistemes Red Hat, i `update-grub` en distribucions basades en Debian. Aquestes eines recopilen informació del sistema i actualitzen `grub.cfg` per reflectir els canvis en els nuclis instal·lats o altres paràmetres d'arrencada.

Precaucions a l'hora de modificar GRUB

És important tenir en compte que, en la majoria de distribucions, el fitxer `grub.cfg` es pot regenerar automàticament després d'una actualització del sistema. Això significa que qualsevol modificació manual que s'hagi fet al fitxer podria perdre's si no es prenen mesures per evitar-ho.

Per tant, si cal personalitzar la configuració d'arrencada, es recomana fer-ho a través dels fitxers de configuració que utilitzen les eines de generació automàtica, com ara `/etc/default/grub` o scripts personalitzats que es troben en `/etc/grub.d/`. D'aquesta manera, els canvis es mantenen de manera segura, fins i tot després de les actualitzacions del sistema.

Parametres de configuració

El fitxer `/etc/default/grub` ens permet definir diferents parametres en forma de variables d'entorn per configurar diferents opcions d'arrencada.

Variable	Descripció
GRUB_BACKGROUND	Defineix la imatge de fons que es mostrarà al menú d'arrencada. Especifica la ruta a una imatge.
GRUB_TIMEOUT	Temps en segons que GRUB esperarà abans de carregar l'entrada predeterminada automàticament.
GRUB_DEFAULT	Entrada per defecte que es carregarà. Pot ser un índex numèric o el nom de l'entrada.
GRUB_CMDLINE_LINUX	Opcions de línia de comandes que es passen al nucli en arrencar el sistema.
GRUB_DISABLE_RECOVERY	Si es defineix com true , desactiva les opcions de mode de recuperació en el menú d'arrencada.
GRUB_DISTRIBUTOR	Defineix el nom de la distribució que es mostrarà al menú d'arrencada.
GRUB_DISABLE_OS_PROBER	Si es defineix com true , impedeix que GRUB busqui altres sistemes operatius instal·lats.
GRUB_PRELOAD_MODULES	Llista de mòduls GRUB que es carregaran abans de mostrar el menú d'arrencada.

Aquests són només alguns exemples de les opcions de configuració disponibles. Per obtenir més informació sobre les opcions de configuració de GRUB, es pot consultar la documentació oficial o les pàgines de manual.

A continuació, es mostra un exemple de configuració de GRUB amb algunes de les opcions de configuració més comunes:

```
GRUB_DEFAULT=0
GRUB_TIMEOUT=5
GRUB_CMDLINE_LINUX="quiet splash"
GRUB_DISABLE_RECOVERY=true
GRUB_DISTRIBUTOR="Debian"
```

En aquest exemple, la configuració predeterminada és carregar la primera entrada del menú (`GRUB_DEFAULT=0`) després d'una espera de 5 segons (`GRUB_TIMEOUT=5`). S'han definit algunes opcions de línia de comandes per al nucli (`GRUB_CMDLINE_LINUX="quiet splash"`) aquestes opcions activen el mode silenciós (`quiet`) i mostren una pantalla de càrrega (`splash`). A més, s'ha especificat que no es mostrarà l'opció de mode de recuperació (`GRUB_DISABLE_RECOVERY=true`). Finalment, s'ha definit el nom de la distribució (`GRUB_DISTRIBUTOR="Debian"`).

GRUB no és l'únic *bootloader* disponible per a sistemes Linux. Altres opcions populars inclouen LILO (Linux Loader) i systemd-boot, entre altres. Cada *bootloader* té les seves pròpies característiques i avantatges, i la seva elecció depèn de les necessitats específiques del sistema.

Initramfs

L'Initramfs (**Initial RAM Filesystem**) és un petit sistema de fitxers integrat a la imatge del nucli Linux. Proporciona els fitxers necessaris perquè el kernel pugui muntar el sistema de fitxers arrel durant l'arrencada. A diferència de l'antic *Initrd*, que s'emmagatzemava en un disc separat, l'Initramfs es carrega completament a la memòria RAM com una imatge comprimida.

Nota:

L'Initramfs no sempre està present, pot estar buit o omès si el sistema no necessita un espai RAM inicial (per exemple, en sistemes simples o compilacions estàtiques del nucli).

Funció i Components de l'Initramfs

L'Initramfs proporciona una estructura mínima per al sistema fins que es pugui muntar el sistema de fitxers arrel complet. Els components principals són:

1. **Fitxers Executables:** Com BusyBox, que encapsula moltes eines Unix bàsiques en un únic executable, proporcionant shells i utilitats com `cp`, `ls`, `mount`, entre altres. A més, pot contenir llibreria dinàmiques i qualsevol programari que vulguem executar durant l'arrencada.
2. **Mòduls del Kernel:** Inclou controladors de maquinari necessaris per accedir a dispositius com discos, xarxes, sistemes RAID o LVM. Aquests mòduls es carreguen des de l'Initramfs si no estan compilats directament dins del nucli.

3. **Fitxers de Dispositiu i Sistemes Especials:** El directori `/dev` conté fitxers de dispositiu com `dev/tty` o `dev/null`. La utilitat `mdev` o `udev` ajuda a gestionar dinàmicament aquests dispositius.

Tots aquests components es troben definits en un fitxer d'inici (`/init`) que s'executa durant l'arrencada. Inclou la lògica per muntar el sistema de fitxers arrel, carregar mòduls del kernel, configurar la xarxa i altres tasques d'inicialització.

Execució de l'Init i Switchroot

Durant l'arrencada, l'script d'inici de l'Initramfs executa el fitxer `/init`. Aquest fitxer pot estar escrit en un llenguatge com *sh* o *Perl*, sempre que l'interpret estigui disponible a l'Initramfs.

Una vegada que s'ha carregat el sistema mínim, es fa servir la comanda `switch_root` (actual) o `pivot_root` (antic) per muntar el sistema de fitxers definitiu i transferir el control al procés d'inicialització principal (PID 1), que normalment és `systemd` o `SysVinit`.

Configuració de l'Initramfs

Es configura durant la compilació del kernel, mitjançant diferents opcions de configuració. Per accedir a aquestes opcions, es pot utilitzar la comanda `make menuconfig` o `make nconfig` durant la compilació del kernel. Algunes de les opcions més rellevants són:

- `CONFIG_BLK_DEV_INITRD`: Habilita la càrrega de l'Initramfs durant l'arrencada.
- `CONFIG_INITRAMFS_SOURCE`: Especifica els fitxers a incloure que poden ser un fitxer CPIO, un directori o un fitxer d'especificació.

Per exemple, un fitxer d'especificació pot contenir llistes de fitxers i directoris a incloure, així com permisos i propietats. A continuació, es mostra un exemple d'un fitxer d'especificació per a l'Initramfs:

```
dir /etc 0755 0 0
dir /dev 0755 0 0
...
```

Nota:

El format `cpio` és més eficient per al kernel comparat amb `tar`, ja que té una estructura més simple i directa per llistats de fitxers, enllaços simbòlics i permisos, reduint així la complexitat durant l'arrencada. Els fitxers `cpio` es generen fàcilment mitjançant eines com:

```
find . | cpio -o -H newc > initramfs.cpio
```

Initramfs vs. Initrd

És important diferenciar entre l'Initramfs i l'Initrd (Initial RAM Disk), ja que són conceptes diferents que s'acostumen a confondre.

- **Initrd:** L'Initrd (Initial RAM Disk) és un arxiu separat carregat pel kernel des de la partició de boot pel *bootloader*.
- **Initramfs:** L'Initramfs és part de la imatge del kernel i es carrega directament a la memòria del kernel.

En distribucions modernes, com Ubuntu, Debian o Fedora, s'utilitza l'Initramfs per la seva flexibilitat i eficiència. L'Initrd encara es pot utilitzar en sistemes més antics o en configuracions específiques, però l'Initramfs és l'opció més comuna en la majoria de sistemes Linux actuals.

Procés d'Arrencada amb Initramfs

1. **Estructura i compressió:** La `initramfs` és un fitxer comprimit (habitualment amb formats com *gzip*, *xz*, *lzma* o *bzip2*) que conté un sistema de fitxers mínim, similar al sistema de fitxers arrel complet que es trobaria en qualsevol sistema Linux. Aquest arxiu s'organitza com un arxiu `cpio`, el qual encapsula fitxers i directoris en un únic arxiu que es descomprimeix a la memòria RAM durant l'arrencada.
2. **Execució de l'Init:** Un cop muntada la imatge, el kernel executa `/init`. Aquest script pot muntar discos, desxifrar particions (LUKS), activar LVM o configurar la xarxa. Un cop el sistema de fitxers definitiu està llest, el procés PID 1 es transfereix mitjançant `switch_root`.
3. **Iniciació d'altres components:** La `initramfs` s'encarrega de carregar mòduls del nucli necessaris, muntar dispositius d'emmagatzematge o configurar la xarxa. En configuracions avançades com *RAID*, *LVM* o *xifratge de discos*, la `initramfs` és essencial per garantir que el sistema pugui arrencar correctament.

Quan s'ha d'actualitzar la initramfs?

La majoria de distribucions Linux actualitzen la initramfs automàticament quan es realitzen canvis en el nucli o en la configuració del sistema. Per exemple, en la configuració de sistemes de fitxers avançats, en configuracions de RAID o LVM, o en la instal·lació de controladors de dispositius específics. També caldrà modificar-la si es canvia el dispositiu d'arrencada o els sistemes de fitxers que es volen muntar a l'arrencada.

Com s'ha de modificar la initramfs?

La modificació de la initramfs es pot fer manualment o mitjançant eines específiques de la distribució. A continuació, es mostren algunes situacions en les quals es podria necessitar modificar la initramfs:

- Actualització de la configuració del sistema: Si es realitzen canvis en la configuració del sistema que afecten la inicialització, com canvis en la configuració de xarxa, mòduls del nucli o dispositius d'emmagatzematge, és necessari actualitzar la initramfs. Això es pot fer amb eines com `update-initramfs` en sistemes Debian o Ubuntu, o `dracut` en sistemes Red Hat.
- Generació d'una nova initramfs: En alguns casos, es pot requerir la creació d'una nova initramfs per a una versió específica del nucli. Això es pot fer amb la comanda `update-initramfs -c -k <versió-del-nucli>` en sistemes Debian o Ubuntu. O bé, amb `dracut -f` en sistemes Red Hat.

Casos pràctics

Arrencada Remota a través de HTTP o NFS

Aquest escenari és útil quan es desitja arrencar un sistema operatiu completament remot o en entorns de xarxa sense disc local.

1. Durant l'arrencada, el nucli carrega la imatge de l'Initramfs, que conté els binaris necessaris (com busybox) i scripts de configuració de xarxa.
2. Mitjançant scripts en l'Initramfs, es configura la interfície de xarxa (per exemple, mitjançant DHCP).
3. Un cop la xarxa està configurada, s'utilitzen comandes com `mount -t nfs` per muntar el sistema de fitxers remot, o es fa ús de clients HTTP per descarregar arxius o imatges.

Un exemple de configuració d'un script init per a un arrencada via NFS podria ser:

```
#!/bin/busybox sh
# Configuració de la interfície de xarxa

ip link set eth0 up
udhcpc -i eth0 # Obtenció de l'adreça IP via DHCP

# Muntar el sistema remot NFS
mount -t nfs ip:/path/to/nfs /mnt

# Canviar l'arrel a la muntada i iniciar el sistema
exec switch_root /mnt /sbin/init
```

LUKS i LVM

Imaginem un escenari on tenim 3 discs durs, on només la partició /boot està fora de LVM i sense xifrar. Però volem que les particions restants estiguin xifrades amb LUKS i gestionades per LVM.

- Disc 1: /dev/sda (EFI i /boot) i sense xifrar.
- Disc 2: /dev/sdb (LUKS xifrat) i gestionat per LVM.
- Disc 3: /dev/sdc (LUKS xifrat) i gestionat per LVM.

En aquest cas, l'Initramfs és essencial per desxifrar els discos i activar els volums LVM abans de muntar el sistema de fitxers arrel. Això es pot fer mitjançant scripts d'inici personalitzats. Per exemple:

```
#!/bin/busybox sh
# Muntar /proc i /sys
mount -t proc proc /proc
mount -t sysfs sys /sys

# Desxifrar els discos amb LUKS
```

```
cryptsetup luksOpen /dev/sdb crypted1
cryptsetup luksOpen /dev/sdc crypted2
```

```
# Activar LVM
```

```
vgchange -a y
```

```
# Muntar el sistema de fitxers arrel
```

```
mount /dev/mapper/vg-root /mnt
```

```
# Canviar l'arrel i iniciar el sistema
```

```
exec switch_root /mnt /sbin/init
```

USB amb clau d'encryptació

Un altre cas pràctic inclou la introducció de seguretat addicional, com la utilització d'un USB que conté una clau d'encryptació per desxifrar discos LUKS. Un cop llegida la clau, es pot exigir que l'usuari retiri el dispositiu USB abans de continuar.

- En primer lloc, es carrega la imatge de l'Initramfs, que conté els binaris necessaris i scripts per gestionar la clau d'encryptació.
- Es demana a l'usuari que connecti el dispositiu USB amb la clau d'encryptació.
- Un cop llegida la clau, es desxifra el disc i es munta el sistema de fitxers arrel.
- Es demana a l'usuari que retiri el dispositiu USB abans de continuar amb l'arrencada.

Un exemple de script d'inici per a aquest escenari podria ser:

```
#!/bin/busybox sh
```

```
# Montar dispositius essencials
```

```
mount -t proc proc /proc
```

```
mount -t sysfs sys /sys
```

```
# Buscar i muntar el dispositiu USB amb la clau
```

```
mount /dev/sdb1 /mnt
```

```
KEYFILE=/mnt/keyfile
```

```
# Desxifrar el disc amb la clau del USB
```

```
cryptsetup luksOpen /dev/sda1 crypted --key-file $KEYFILE
```

```
# Demanar que es retiri el USB abans de continuar
```

```
echo "Retira el dispositiu USB i prem Enter per continuar."
```

```
read
```

```
# Continuar el procés normal d'arrencada
```

```
vgchange -a y
```

```
mount /dev/mapper/vg-root /mnt
```

```
exec switch_root /mnt /sbin/init
```

Processos d'Inici del Sistema

Referències

- GRUB Manual
- UEFI Specifications
- Linux Kernel Documentation
- Debian GRUB Configuration
- Red Hat GRUB Configuration
- Linux Boot Process
- Understanding the Linux Boot Process
- BIOS vs. UEFI
- Understanding the Linux Kernel
- Linux Kernel Documentation
- Linux Boot Process
- Understanding the Linux Boot Process
- BIOS vs. UEFI

- Understanding the Linux Kernel