

CURSO DE GNU/Linux

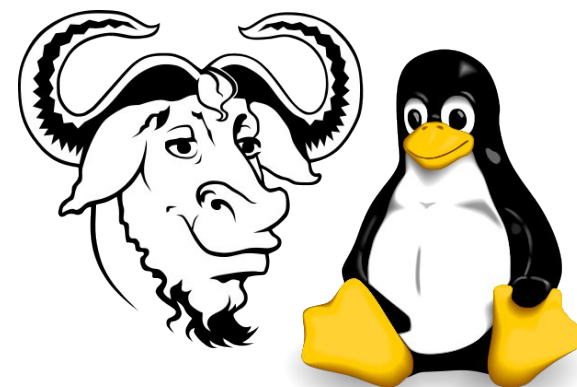
Proceso de Arranque





Introducción

Proceso de Arranque



Resumen Proceso de Inicio



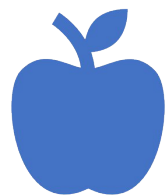
Secuencia de Arranque PC



Arranque Linux

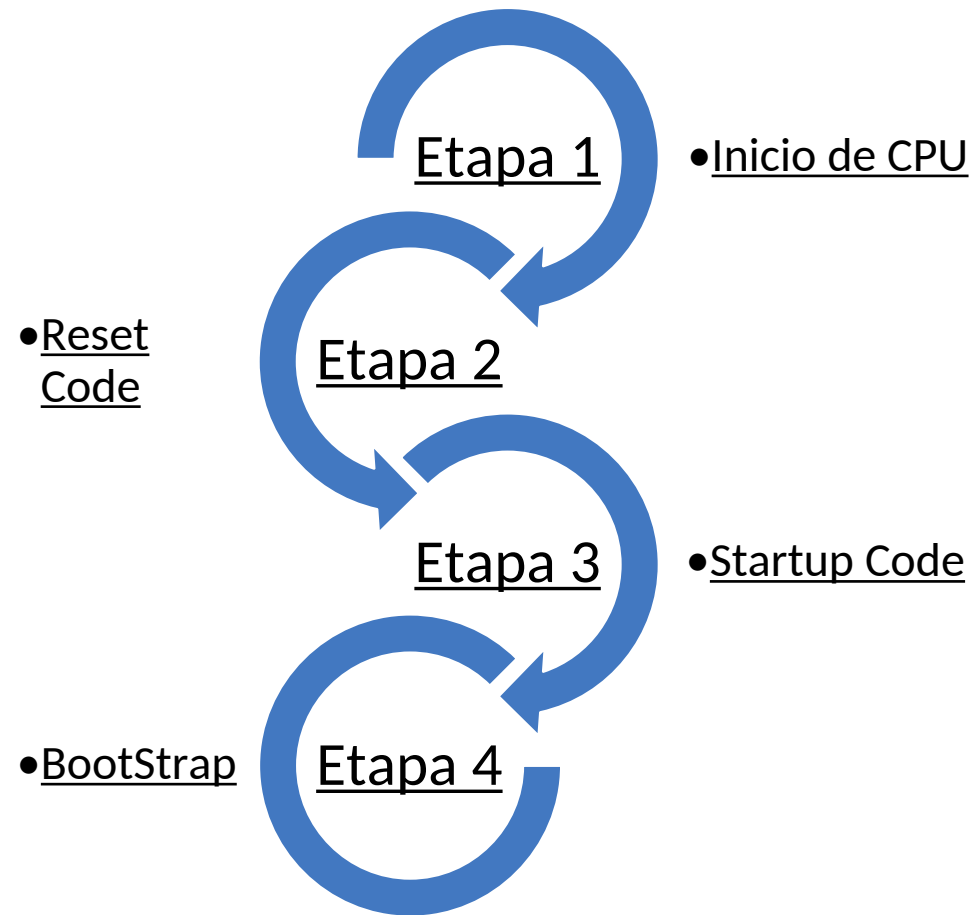


Arranque Windows



Secuencia de Arranque Mac

Secuencia de Arranque General



Secuencia de Arranque General

E1: Inicio CPU

- RTC + Oscilador (RAM, CPU, Buses,...)
- Botón de Encendido -> Señal PowerGood
- Registros (como Code Segment -> FFFF0h)

E2: Reset Code

- Programa de memo persistente
- Rutina de inicio
- Complementos HW para CPU
- PIC (Interrupciones)

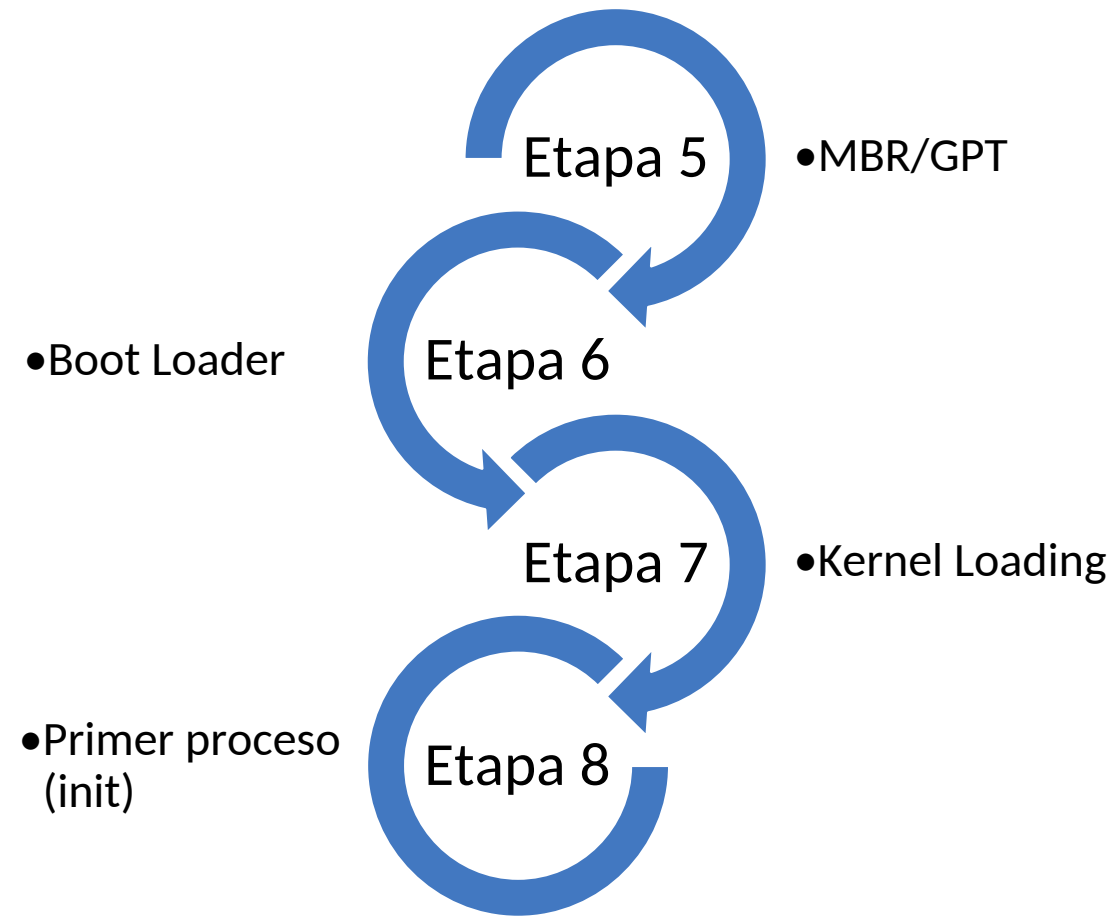
E3: StarUp Code

- Rutina desde memo no volátil: POST

E4: BootStrap

- Desde un dispositivo de arranque (HD, SSD, USB, ...), el BootLoader ejecuta instrucciones de arranque para que SO -> RAM

Secuencia de Arranque Linux



Secuencia de Arranque en Linux

E5: MBR/GPT

- **BIOS Legacy o CSM.**

Activa la rutina IPL en los primeros 512 bytes del primer sector del HD O MBR (446 Programas+64 Tabla Particiones+2 magic number AA55).

Linux MBR vacío => inicia mediante BootLoader

- **EFI/UEFI.**

Arranca desde GUID Partition Table. Guarda los ficheros de firmware en EFI System Partition

E6: BootLoader (desde /boot)

- **Stage 1:** Carga en RAM primeros 512 B de MEM y ejecuta mediante CPU

- **Stage 1.5:** Busca el S2 en el FS

- **Stage 2:** Carga GRUB2. El BL apunta al kernel y monta el initramfs en /dev/ram0

E7: Kernel Loading

- **Kernel Loading Stage**

Carga un FS en la RAM (initrd/initramfs) donde descomprime el Kernel para poder cargar los recursos físicos del SO

- **Kernel Init Stage**

Mediante swapper o proceso 0 (startup) maneja la RAM, CPU, IRQ, syscall start_kernel(), monta ramdisk y controladores no compilados (udev)

- **pivot_root()**

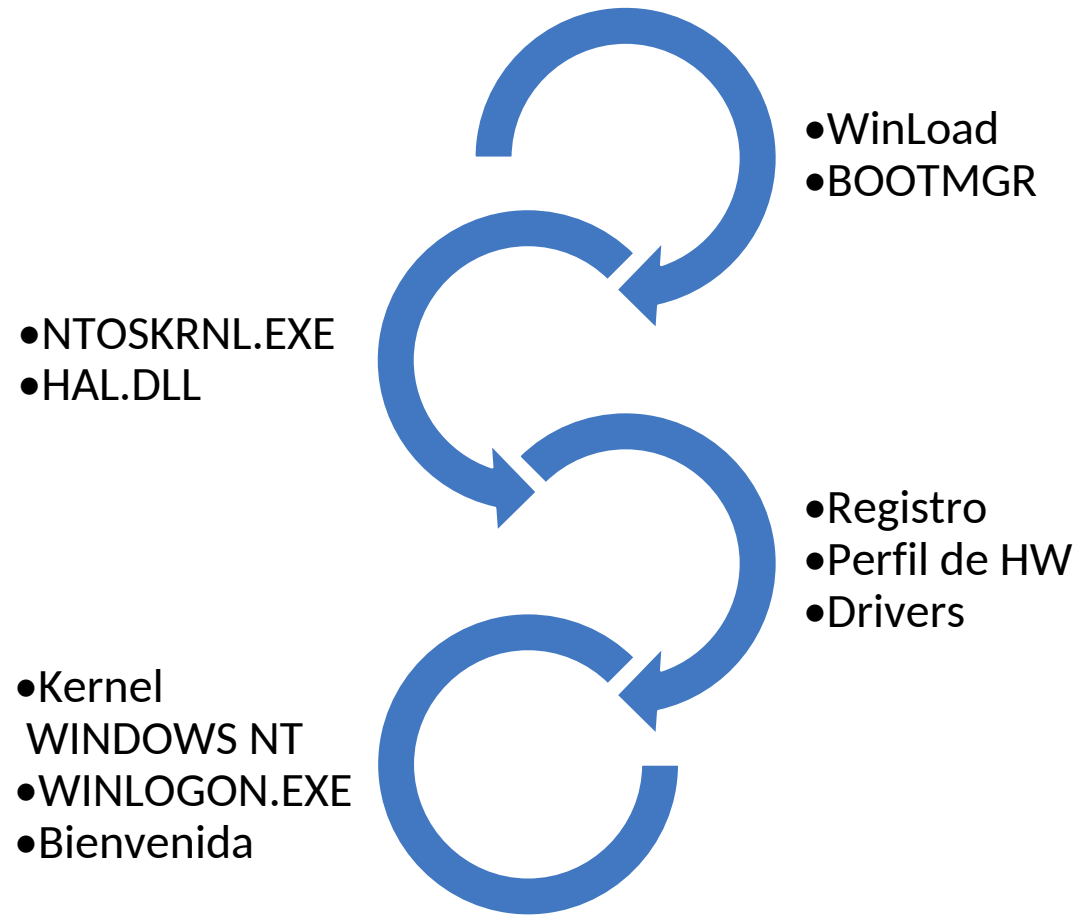
Demonta el FS y carga el proceso 1 o PID=1

E8: Primer Proceso (init)

- PID=1 suele ser asumido por **systemd**

- PID=1 inicia scripts para sesión gráfica, mono/multisusuario...

Secuencia de Arranque Windows



Secuencia de Arranque en Windows

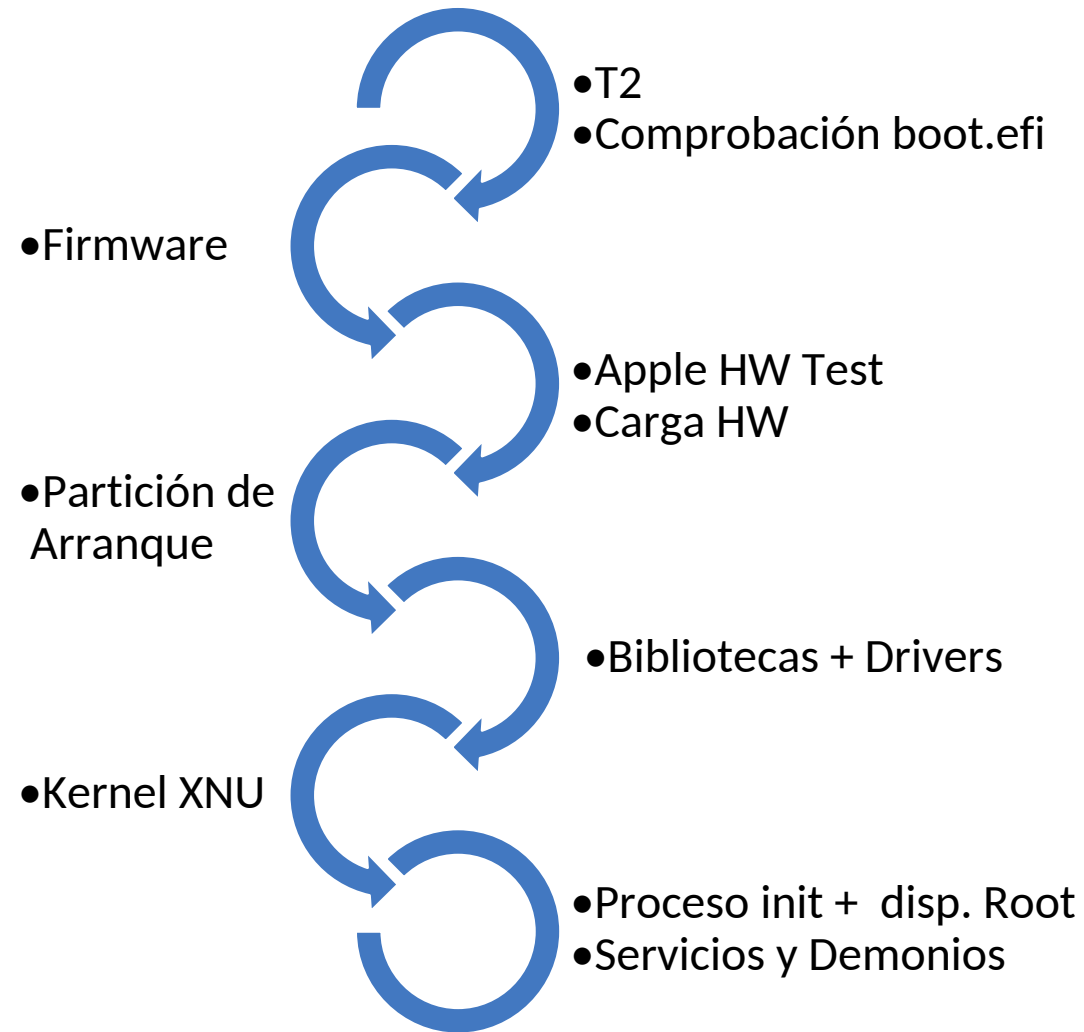
- Carga el Windows Loader (WinLoad) y usa la ruta BOOTMGR para montar la partición de arranque

- Carga NTOSKRNL.EXE para la gestión del Kernel
- Carga HAL.DLL para la gestión del HW

- WinLoad lee los **archivos del Registro** de Windows y selecciona un perfil de hardware, cargando todos los drivers necesarios.

- Carga el kernel Windows NT
- Inicia el archivo de inicio de sesión WINLOGON.EXE
- Muestra la pantalla de bienvenida

Secuencia de Arranque Mac



Secuencia de Arranque en Mac

- Botón de encendido

- [T2+firmware evalúa la firma del boot.efi]

- OpenFirmware (PowerPC), el EFI (Intel x86) o el firmware para los ARM

- Apple Hardware Test y se inicializa el hardware presente

- Selecciona partición de arranque

- /System/Library/CoreServices/BootX del bootload
er
- Carga drivers de la lista
/usr/sbin/ke
xtcache.

- Carga el kernel XNU, que toma el control

- EL kernel ejecuta el proceso init
- Se determina el dispositivo root

- Inicia servicios y demonios como BSD

- Inicia sesión

Esta presentación se difunde únicamente con fines docentes.

Las imágenes empleadas pueden pertenecer a terceros y, por tanto, son propiedad de sus autores.

