

LPIC-2 / Examen 202 - Arranque del Sistema

202.2 Arranque del sistema

Teoría

El proceso de arranque de un sistema Linux es una secuencia compleja de pasos, donde cada etapa carga y prepara la siguiente. Entender esta secuencia es vital para diagnosticar problemas de arranque.

Secuencia del Proceso de Arranque:

1. Hardware (BIOS/UEFI):

- **POST (Power-On Self-Test):** La BIOS/UEFI inicializa y verifica el hardware básico (CPU, RAM, dispositivos esenciales).
- **Inicialización del Hardware:** La BIOS/UEFI configura el hardware para que el sistema operativo pueda tomar el control.
- **Búsqueda del Dispositivo de Arranque:** La BIOS/UEFI consulta su configuración para encontrar el dispositivo desde el cual arrancar (disco duro, CD/DVD, red).
- **Carga del Gestor de Arranque:** Lee el código del gestor de arranque desde el sector de arranque del dispositivo (MBR en BIOS heredada o una partición EFI en UEFI) y lo carga en la memoria RAM.

2. Gestor de Arranque (Bootloader):

- El gestor de arranque es el primer programa que se ejecuta en el disco. Su trabajo principal es cargar el kernel y el initramfs en la memoria y pasar el control al kernel.
- **GRUB 2 (GRand Unified Bootloader versión 2):** Es el gestor de arranque estándar en la mayoría de las distribuciones Linux modernas.
- **Ubicación de GRUB 2:** Se instala en el Master Boot Record (MBR) del disco (en sistemas BIOS) o en la partición del sistema EFI (ESP) en sistemas UEFI. Los archivos de configuración y módulos residen principalmente en `/boot/grub/` (BIOS) o `/boot/grub2/` o rutas específicas de EFI (UEFI).
- **Archivo de Configuración de GRUB 2:** `/boot/grub/grub.cfg` (Debian/Ubuntu) o `/boot/grub2/grub.cfg` (Red Hat/CentOS/Fedora con BIOS) o rutas bajo `/boot/efi/` (Red Hat/Fedora con UEFI). Este archivo **no debe editarse manualmente**; se genera automáticamente a partir de:
 - `/etc/default/grub`: Archivo de configuración principal con opciones globales (timeout, runlevel por defecto - aunque ahora es un target de systemd, kernel parameters).
 - `/etc/grub.d/`: Directorio con scripts que generan secciones del archivo `grub.cfg` (ej: listado de kernels disponibles, opciones de recuperación).
- **Generación de la Configuración de GRUB 2:**
 - **Rama Debian/Ubuntu:** `sudo update-grub` (un script que ejecuta `grub-mkconfig`).

- **Rama Red Hat/CentOS/Fedora:** `sudo grub2-mkconfig -o <ruta_a_grub.cfg>` (ej: `sudo grub2-mkconfig -o /boot/grub2/grub.cfg` para BIOS, la ruta para UEFI varía).
- **Función:** Presenta un menú con las opciones de arranque disponibles (diferentes kernels, sistemas operativos). El usuario selecciona una opción o se elige la predeterminada después de un tiempo de espera. El gestor de arranque luego carga la imagen del kernel (`vmlinux`) y el `initramfs` (`initrd.img/initramfs-`) asociados a la opción seleccionada en la memoria RAM.

3. El Kernel:

- El kernel se descomprime a sí mismo en memoria.
- Inicializa el hardware esencial y los controladores básicos.
- Carga el `Initramfs`.
- Ejecuta el primer proceso en el espacio de usuario, que es el programa `/init` dentro del `Initramfs`.

4. Initramfs (Initial RAM Filesystem): (Revisitado de 201.1)

- Es un sistema de archivos raíz temporal en memoria que contiene los módulos del kernel necesarios (ej: drivers para el controlador de disco, soporte RAID/LVM/cifrado) y utilidades básicas para montar el sistema de archivos raíz real.
- El script `/init` dentro del `initramfs` se encarga de:
 - Cargar los módulos de kernel requeridos.
 - Detectar el sistema de archivos raíz (basado en el parámetro `root=` pasado por el bootloader, a menudo usando UUIDs o LABELS).
 - Desbloquear volúmenes cifrados (si aplica, pidiendo la frase de paso).
 - Montar el sistema de archivos raíz real (ej: `/dev/sda1`) en el directorio `/sysroot` o similar dentro del `initramfs`.
 - Realizar una operación `pivot_root` o `chroot` para que el sistema de archivos real se convierta en el nuevo `/`.
 - Ejecutar el proceso `init` (PID 1) del sistema de archivos raíz real (`SysVinit`, `Upstart` o `systemd`).

5. Sistema de Inicio (Init System - PID 1): (SysVinit vs. systemd)

- **SysVinit:** Lee `/etc/inittab` para determinar el runlevel por defecto. Ejecuta los scripts de `/etc/init.d/` en los directorios `rcX.d` secuencialmente, primero los `K##` y luego los `S##`, para llevar el sistema al runlevel deseado.
- **systemd:** El sistema de inicio moderno, predominante. Lee su propia configuración (unidades, targets). Arranca los servicios y otros procesos de forma **paralela**, basándose en unidades (que representan servicios, puntos de montaje, dispositivos, etc.) y **targets** (que reemplazan conceptualmente a los runlevels).
 - **Targets de systemd:** `multi-user.target` (equivalente a runlevel 3), `graphical.target` (equivalente a runlevel 5, depende de `multi-user.target`), `rescue.target` (modo de rescate, similar a runlevel 1),

`emergency.target` (shell mínima para depuración), `reboot.target` (equivalente a `runlevel 6`), `poweroff.target` (equivalente a `runlevel 0`).

- El target por defecto al arrancar está definido por un enlace simbólico desde `/etc/systemd/system/default.target` (enlace estándar en **ambas ramas**) a uno de los targets disponibles en `/usr/lib/systemd/system/`.
- `systemctl isolate <target>`: Cambia el sistema a un target diferente (similar a `init <runlevel>`).

6. **Servicios y Deminios:** El sistema de inicio lanza los servicios y demonios configurados para el `runlevel/target` deseado (servidor SSH, servidor web, cron, etc.).

Mensajes del Proceso de Arranque:

- Los mensajes generados por el kernel durante el arranque se almacenan en el buffer del kernel y se pueden ver con `dmesg`.
- Los mensajes de los servicios y procesos iniciados por `systemd` se registran en el journal, accesibles con `journalctl -b`.
- En sistemas SysVinit antiguos, los mensajes de los scripts de inicio se enviaban a la consola o a archivos de log específicos.

Acceso a Modos de Recuperación (Single-User / Rescue Mode):

Fundamental para solucionar problemas de arranque, sistema de archivos o contraseñas. Se accede típicamente desde el menú del gestor de arranque (GRUB):

- **Desde el menú de GRUB:** Edita la entrada de arranque del kernel (presionando `e`).
- **Añadir parámetro al kernel:** Busca la línea que carga el kernel (empieza con `linux` o `linuxefi`). Al final de esa línea, añade `single` o `1` para modo monousuario (SysVinit) o `systemd.unit=rescue.target` para modo de rescate (`systemd`).
- **Shell de depuración temprana:** Añadir `init=/bin/bash` en los parámetros del kernel te dará una shell Bash muy temprana justo después de que el kernel y el `initramfs` se carguen y monten la raíz, antes de que se inicie el sistema de inicio principal. Útil para depurar problemas con el sistema de inicio.
- Después de editar los parámetros, presiona `Ctrl+X` o `F10` para arrancar con esos parámetros modificados.