

📁 LPIC-2 / 🔄 Examen 202 - Arranque del Sistema - Ejercicios

Nota: Estos ejercicios se centran en explorar la estructura de SysVinit. Si tu sistema usa `systemd` (lo más probable), notarás que `inittab` es ignorado y los directorios `rcX.d` contienen enlaces que son gestionados de forma diferente a como lo hacía `init` directamente.

Ejercicio 2.1.1: Identificando el Proceso Init (PID 1)

- **Objetivo:** Determinar qué sistema de inicio está realmente corriendo como PID 1.
- **Requisitos:** Acceso a la línea de comandos.
- **Desarrollo Paso a Paso:**
 1. Abre una terminal.
 2. **Verifica el nombre del proceso con PID 1:** Ejecuta `ps -p 1 -o comm=`.
 3. **Interpretación:** Si la salida es `init` (o a veces `systemd`), es probable que `init` sea un enlace simbólico al binario real del sistema de inicio. Verifica: `ls -l /sbin/init`. Si apunta a `/lib/systemd/systemd`, tu sistema usa `systemd`. Si apunta a otro binario (raro en sistemas modernos), podría ser SysVinit o Upstart. En la gran mayoría de sistemas modernos LPIC-2, será `systemd`.

Ejercicio 2.1.2: Explorando `/etc/inittab`

- **Objetivo:** Ver el archivo de configuración principal de SysVinit y si está en uso.
- **Requisitos:** Acceso a la línea de comandos.
- **Desarrollo Paso a Paso:**
 1. Abre una terminal.
 2. **Visualiza el contenido de `/etc/inittab`:** Ejecuta `cat /etc/inittab`.
 3. **Busca la línea `initdefault`:** Identifica el runlevel por defecto (un número entre 2 y 5).
 4. **Busca comentarios:** Lee los comentarios al principio del archivo. A menudo indican si el archivo es ignorado por el sistema de inicio actual (`systemd`).
 5. **(Conclusión):** Si tu sistema usa `systemd` (ver Ej. 2.1.1), este archivo es mayormente informativo o solo se usa la línea `initdefault` para compatibilidad con el objetivo `multi-user.target` o `graphical.target` de `systemd`.

Ejercicio 2.1.3: Explorando el Directorio de Scripts Init (`/etc/init.d/`)

- **Objetivo:** Identificar los scripts de inicio tradicionales disponibles en el sistema.
- **Requisitos:** Acceso a la línea de comandos.
- **Desarrollo Paso a Paso:**
 1. Abre una terminal.
 2. **Lista el contenido de `/etc/init.d/`:** Ejecuta `ls -l /etc/init.d/`. Verás los scripts para varios servicios (`apache2`, `ssh`, `networking`, etc.).
 3. **Verifica los permisos:** Observa que son archivos ejecutables (tienen la `x`).
 4. **Visualiza el contenido de un script (ej: `ssh`):** Ejecuta `less /etc/init.d/ssh`. Observa la estructura del script (definiciones de funciones

`start()`, `stop()`, `restart()`, `status()`, y el `case` principal que maneja los argumentos pasados).

Ejercicio 2.1.4: Explorando los Directorios de Runlevels (/etc/rcX.d/)

- **Objetivo:** Ver cómo los scripts de `/etc/init.d/` están enlazados a los runlevels.
- **Requisitos:** Acceso a la línea de comandos.
- **Desarrollo Paso a Paso:**
 1. Abre una terminal.
 2. **Lista el contenido del directorio del runlevel 3 (servidor multiusuario con red):** Ejecuta `ls -l /etc/rc3.d/`. Verás enlaces simbólicos que empiezan con S o K y dos dígitos, apuntando a scripts en `/etc/init.d/`.
 3. **Identifica los scripts de inicio (S##) y de parada (K##):** Los S## se ejecutarían con el argumento `start` al entrar en el runlevel 3 en un sistema SysVinit puro. Los K## se ejecutarían con el argumento `stop` al salir de un runlevel superior o al entrar en un runlevel inferior donde deban detenerse.
 4. **Compara con el runlevel 5 (escritorio gráfico):** Ejecuta `ls -l /etc/rc5.d/`. Probablemente verás muchos de los mismos enlaces S## que en el runlevel 3, más enlaces para servicios gráficos como el gestor de pantalla (S##gdm3, S##lightdm, etc.).
 5. **Compara con el runlevel 0 (halt):** Ejecuta `ls -l /etc/rc0.d/`. Aquí verás casi todos los enlaces empezando con K## para detener los servicios antes de apagar.
 6. **(Conclusión con systemd):** En un sistema systemd, estos directorios a menudo existen por compatibilidad (especialmente en Debian/Ubuntu), y systemd puede convertirlos internamente a unidades de servicio o "targets". Sin embargo, la lógica de dependencia y paralelismo de systemd es diferente a la ejecución secuencial y simple de SysVinit de estos scripts.

Ejercicio 2.1.5: Usando el Comando `runlevel`

- **Objetivo:** Ver los runlevels actual y anterior.
- **Requisitos:** Acceso a la línea de comandos.
- **Desarrollo Paso a Paso:**
 1. Abre una terminal.
 2. **Ejecuta el comando:** Ejecuta `runlevel`.
 3. **Interpretación:** La salida mostrará dos valores. El primero es el runlevel anterior (a menudo N si no ha habido cambio desde el arranque). El segundo es el runlevel actual (ej: 3 o 5).
 4. **(Nota con systemd):** En sistemas systemd, `runlevel` aún puede funcionar por compatibilidad, mapeando los targets de systemd a runlevels SysVinit (ej: `multi-user.target` mapea a 3, `graphical.target` mapea a 5).

Ejercicio 2.1.6: (Conceptual) Cambiando de Runlevel con `init`

- **Objetivo:** Entender el comando para cambiar de runlevel, sin ejecutarlo a menos que sea seguro.
- **Requisitos:** Acceso a la línea de comandos. Privilegios de superusuario (sudo). **Entorno de VM de prueba SIN TRABAJO CRÍTICO.**
- **Desarrollo Paso a Paso:**
 1. Abre una terminal.
 2. **Comprende el comando `sudo init 1`:** Esto intentaría cambiar al modo monousuario. Cerraría la mayoría de los servicios y usuarios. **NO LO EJECUTES EN UN SISTEMA REAL.** En systemd, esto podría mapearse al target `rescue.target`.
 3. **Comprende el comando `sudo init 0`:** Esto apagaría el sistema. **NO LO EJECUTES ACCIDENTALMENTE.** En systemd, se mapea al target `poweroff.target`.
 4. **Comprende el comando `sudo init 6`:** Esto reiniciaría el sistema. **NO LO EJECUTES ACCIDENTALMENTE.** En systemd, se mapea al target `reboot.target`.
 5. **Si estás en una VM de prueba y quieres ver el efecto (¡Cuidado!):** Puedes intentar cambiar a un runlevel multiusuario diferente si tu sistema no está ya en el 3 o 5. Por ejemplo, si tu sistema usa runlevel 5 por defecto, puedes intentar `sudo init 3`. Esto detendría los servicios asociados al runlevel 5 (como el gestor de pantalla) y te llevaría a una consola de texto en runlevel 3. Luego podrías volver con `sudo init 5` (o `sudo telinit 5`). En systemd, esto mapearía a `multi-user.target` y `graphical.target`.