

## Examen 101 - Arquitectura del Sistema

Este examen se centra en la comprensión de la arquitectura fundamental de un sistema Linux, el proceso de arranque, la gestión de hardware y los niveles de ejecución.

### 101.1 Determinar y configurar ajustes hardware

#### Teoría Profunda:

El sistema operativo Linux necesita interactuar con el hardware subyacente para funcionar. Esta interacción se realiza a través de controladores de dispositivo (drivers), que son módulos de código dentro del kernel de Linux. La detección de hardware ocurre principalmente durante el arranque del sistema (boot).

#### 1. Detección de Hardware:

- Durante el arranque, el kernel sondea el bus de hardware (como PCI, USB) para identificar los dispositivos conectados.
- Utiliza identificadores únicos de hardware (Vendor ID, Product ID) para determinar el tipo de dispositivo.
- Una vez identificado, el kernel carga el módulo (driver) apropiado que sabe cómo comunicarse con ese hardware específico.
- El directorio `/dev/` es crucial aquí. No contiene archivos "reales" en el sentido tradicional, sino que proporciona interfaces (nodos de dispositivo) para interactuar con el hardware. Por ejemplo, `/dev/sda` representa el primer disco SATA, `/dev/ttyS0` el primer puerto serie, etc. Estos nodos pueden ser de "carácter" (para dispositivos que se leen/escriben byte a byte, como terminales) o de "bloque" (para dispositivos que manejan datos en bloques, como discos).

#### 2. Información del Hardware en Tiempo de Ejecución:

- El **sistema de archivos virtual `/proc`** es una mina de oro de información del sistema en tiempo de ejecución, *incluyendo hardware*.
  - `/proc/cpuinfo`: Detalles sobre el(los) procesador(es).
  - `/proc/meminfo`: Información sobre la memoria RAM (total, libre, buffers, caché, swap).
  - `/proc/devices`: Lista los dispositivos de caracteres y bloque mayores asignados y sus nombres internos.
  - `/proc/interrupts`, `/proc/ioports`, `/proc/dma`: Muestran qué dispositivos están usando qué recursos del sistema (interrupciones, puertos I/O, canales DMA).
- El **sistema de archivos virtual `/sys`** proporciona una vista jerárquica y más estructurada de los dispositivos del sistema, los buses, los controladores, etc. Es una interfaz más moderna y programática al kernel que `/proc`. Permite a los programas y scripts interactuar con el hardware de una manera más organizada.

- **Herramientas de usuario:** Varias herramientas de línea de comandos interactúan con `/proc` y `/sys` (o directamente con las interfaces del kernel) para mostrar información legible:
  - `lspci`: Lista dispositivos PCI (tarjetas gráficas, de red, sonido, etc.). Lee información del directorio `/sys/bus/pci/devices/`.
  - `lsusb`: Lista dispositivos USB. Interactúa con el subsistema USB del kernel, a menudo a través de `/sys`.
  - `lshw`: Lista información detallada sobre todo el hardware. Agrega información de varias fuentes (`/proc`, `/sys`, DMI/BIOS).
  - `dmidecode`: Reporta información de hardware desde la DMI/BIOS.
  - `lsblk`: Lista dispositivos de bloques (discos, particiones, LVM). Lee información de `/sys/block`.
  - `fdisk`, `parted`: Herramientas para gestionar particiones en dispositivos de bloque.

### 3. Gestión de Módulos del Kernel (Drivers):

- Los drivers suelen estar compilados como módulos (`.ko` files) que pueden cargarse o descargarse del kernel en tiempo de ejecución. Esto permite que el kernel sea más pequeño y cargue solo los drivers necesarios.
- Comandos clave:
  - `lsmod`: Lista los módulos del kernel cargados actualmente. Muestra dependencias entre módulos.
  - `modinfo <module_name>`: Muestra información detallada sobre un módulo específico (autor, descripción, dependencias, parámetros).
  - `modprobe <module_name>`: Carga un módulo del kernel (y sus dependencias).
  - `rmmod <module_name>`: Descarga un módulo del kernel (si no está en uso y no tiene dependencias cargadas que lo requieran).
- Los archivos de configuración para `modprobe` se encuentran típicamente en `/etc/modprobe.d/`. Esto permite configurar alias para módulos, opciones de carga, o prohibir la carga de ciertos módulos (blacklist).

### 4. Configuración de Hardware:

- Aunque gran parte de la configuración de hardware es automática, a veces se necesita intervención manual:
  - Configurar parámetros del kernel al arrancar (ver 101.2).
  - Usar archivos en `/etc/modprobe.d/` para pasar opciones a los módulos o manejar conflictos.
  - Configurar dispositivos a través de herramientas específicas de usuario (ej: `ip` para redes, `hdparm` para discos, `xrandr` para pantallas).
  - Editar archivos de configuración en `/etc/udev/rules.d/` para personalizar la forma en que `udev` (el demonio que gestiona los dispositivos

en `/dev`) nombra o configura dispositivos al detectarlos (ej: asegurar que una unidad USB particular siempre se monte en un punto específico).

**Conceptos Clave:** Kernel Modules, `/dev`, `/proc`, `/sys`, `udev`, PCI, USB, IRQ, DMA, I/O Ports.