

LPIC-2 / Examen 201 - Kernel de Linux

Este examen cubre los componentes, la compilación y la gestión de módulos y hardware relacionados con el kernel.

201.1 Componentes del Kernel

Teoría

El Kernel de Linux es el núcleo del sistema operativo. Es un programa que se carga al iniciar el ordenador y reside en la memoria principal, actuando como una interfaz entre el hardware de la computadora y los procesos de usuario (aplicaciones). Gestiona los recursos del sistema de bajo nivel y proporciona una plataforma para que las aplicaciones se ejecuten.

1. Rol Fundamental del Kernel:

- **Gestión de Procesos:** Controla la creación, planificación (scheduling) y terminación de procesos. Decide qué proceso se ejecuta en la CPU y cuándo (planificador de procesos).
- **Gestión de Memoria:** Administra la memoria RAM del sistema, asignando memoria a los procesos, gestionando la memoria virtual (uso de swap) y optimizando el uso de la caché.
- **Gestión de Hardware:** Proporciona una abstracción del hardware subyacente, permitiendo que las aplicaciones interactúen con dispositivos (CPU, memoria, discos, tarjetas de red, periféricos) a través de interfaces estandarizadas (llamadas al sistema).
- **Sistema de Archivos:** Implementa la capa del sistema de archivos (VFS - Virtual Filesystem Switch) que permite interactuar con diferentes tipos de sistemas de archivos (ext4, XFS, etc.) de manera uniforme.
- **Pila de Red (Network Stack):** Implementa los protocolos de red (TCP/IP, etc.) para la comunicación.
- **Controladores de Dispositivos (Device Drivers):** Código dentro del kernel (o cargado como módulos) que permite al kernel interactuar con hardware específico.

2. Espacio del Kernel vs. Espacio de Usuario:

- El kernel se ejecuta en un modo de privilegio especial de la CPU (modo kernel o anillo 0), con acceso total a todo el hardware y la memoria.
- Las aplicaciones de usuario se ejecutan en un modo sin privilegios (modo usuario o anillo 3). Cuando una aplicación necesita interactuar con el hardware o con otros recursos gestionados por el kernel, realiza una "llamada al sistema" (syscall), que es una solicitud para que el kernel realice la operación necesaria en su nombre.

3. Estructura y Componentes del Kernel (Conceptuales): El kernel es un código grande y complejo, pero se organiza en subsistemas:

- Subsistema de Gestión de Procesos (incluye el planificador).

- Subsistema de Gestión de Memoria.
 - Subsistema VFS (Virtual Filesystem Switch).
 - Subsistema de Red.
 - Subsistema de Gestión de Dispositivos / Controladores de Dispositivos.
 - Subsistema de Comunicación Entre Procesos (IPC).
4. **Versiones del Kernel:** Las versiones del kernel siguen un esquema numérico (ej: 5.15.0-105-generic).
- Mayor.Menor.Parche-Extra
 - Mayor: Cambios grandes, a veces incompatibles.
 - Menor: Nuevas funcionalidades, drivers.
 - Parche: Correcciones de errores, seguridad.
 - Extra: Sufijo específico de la distribución o compilación (indica distribución, arquitectura, tipo de kernel como "generic", "lowlatency", etc.).
5. **Imagen del Kernel y Initramfs:**
- El archivo ejecutable del kernel reside típicamente en el directorio `/boot/`. Su nombre suele empezar por `vmlinuz` - seguido de la versión del kernel (ej: `/boot/vmlinuz-5.15.0-105-generic`).
 - **Initramfs (Initial RAM Filesystem):** Es un pequeño sistema de archivos raíz inicial que se carga en RAM antes de que el kernel monte el sistema de archivos raíz real. Contiene controladores de dispositivos esenciales (para discos, controladores RAID/LVM) y scripts necesarios para montar el sistema de archivos raíz real. Es un archivo comprimido, también en `/boot/` (ej: `/boot/initrd.img-5.15.0-105-generic` o `/boot/initramfs-5.15.0-105-generic.img`).
6. **Interfaces del Kernel en `/proc` y `/sys`:**
- **/proc:** Un sistema de archivos virtual (en RAM) que proporciona información sobre procesos en ejecución (subdirectorios numéricos para PIDs) y sobre el estado y la configuración del kernel. Muchos archivos aquí reflejan parámetros del kernel (ej: `/proc/cpuinfo`, `/proc/meminfo`, `/proc/version`, `/proc/cmdline`).
 - **/sys:** Otro sistema de archivos virtual (en RAM) que proporciona una vista estructurada del hardware del sistema y sus dispositivos, así como de la configuración del kernel. Es una interfaz más organizada para interactuar con dispositivos (ej: `/sys/class/net/` para interfaces de red, `/sys/block/` para dispositivos de bloque).

Diferencias Debian vs. Red Hat (Kernel):

- **Nombres de Paquetes del Kernel:** Los paquetes principales del kernel y sus archivos asociados tienen nombres que siguen las convenciones de empaquetado de cada rama (ej: `linux-image-...` en Debian/Ubuntu, `kernel-...` en Red Hat/Fedora).

- **Nombres de Archivos en /boot:** La convención de nombres para `vmlinux - e`
`initrd.img/initramfs` - puede variar ligeramente en el sufijo específico de la
distribución y la versión exacta.
- **Ubicación y Contenido de Algunos Archivos de Configuración del Kernel:** Los archivos
de configuración compilados del kernel (si se guardan) pueden estar en
`/boot/config- . . .` o `/proc/config.gz`. Los archivos relacionados con los
módulos del kernel (`/etc/modules-load.d/`, `/etc/modprobe.d/`) son en su
mayoría estándar, pero algunos archivos de configuración por defecto pueden variar.