

LPIC-2 / Examen 201 - Kernel de Linux

201.3 Gestionar módulos del kernel y sus parámetros

Teoría

Los módulos del kernel son bloques de código objeto que pueden cargarse y descargarse del kernel en tiempo de ejecución según sea necesario. Permiten al kernel tener una arquitectura modular, lo que ofrece varias ventajas:

- **Tamaño Reducido del Kernel Base:** Las funcionalidades o drivers que no son esenciales para el arranque del sistema se compilan como módulos, manteniendo la imagen principal del kernel (`vmlinux`) más pequeña.
- **Flexibilidad:** Se puede añadir soporte para nuevo hardware o sistemas de archivos simplemente cargando el módulo apropiado, sin necesidad de recompilar o reiniciar el kernel.
- **Mantenimiento Simplificado:** Los drivers de dispositivos a menudo se desarrollan y mantienen como módulos separados.

Archivos de Módulos Compilados:

- Los archivos de módulos del kernel (`.ko` - Kernel Object) se almacenan en el sistema de archivos bajo `/lib/modules/<versión_kernel>/`.
- Dentro de este directorio, los módulos se organizan en subdirectorios (ej: `kernel/drivers/net/ethernet/intel/igb.ko` para un driver de red Intel).
- También hay archivos generados por `depmod` (ver abajo) en este directorio, como `modules.dep` (que lista las dependencias entre módulos) y mapas de símbolos.

Herramientas de Línea de Comandos para la Gestión de Módulos:

1. `lsmod`:

- Muestra una lista de los módulos del kernel que están actualmente cargados en la memoria.
- Columnas de salida:
 - `Module`: Nombre del módulo.
 - `Size`: Tamaño del módulo en bytes.
 - `Used by`: Número de veces que el módulo está siendo usado por otros módulos o subsistemas.
 - `Used by` (segunda columna): Nombres de otros módulos que dependen de este.
- Ejemplo: `lsmod | grep mi_modulo`

2. `modinfo <nombre_módulo>`:

- Muestra información detallada sobre un archivo de módulo específico (el archivo `.ko`), no sobre un módulo cargado.

- Busca el archivo `.ko` en el árbol de módulos cargado actualmente en `/lib/modules/$(uname -r)/`.
- Muestra campos como `filename`, `description`, `author`, `license`, `depends` (módulos de los que depende), `parm` (parámetros que acepta el módulo con su descripción).
- Ejemplo: `modinfo igb` (muestra info sobre el driver de red igb).

3. **modprobe <nombre_módulo>:**

- Carga un módulo del kernel y **automáticamente carga cualquier módulo del que dependa**.
- Busca el módulo en el árbol de módulos instalado (`/lib/modules/$(uname -r)/`), utilizando los archivos de dependencia generados por `depmod`.
- Requiere privilegios de root (`sudo`).
- Ejemplo: `sudo modprobe nouveau` (carga el driver de video Nouveau).

4. **rmmod <nombre_módulo>:**

- Descarga un módulo del kernel.
- Solo puede descargar un módulo si no está en uso (`Used by` es 0 en `lsmod`) y ningún otro módulo cargado depende de él.
- Requiere privilegios de root (`sudo`).
- Ejemplo: `sudo rmmod nouveau`.

5. **depmod:**

- Crea archivos de mapa de dependencias de módulos y otros archivos para el kernel especificado.
- Normalmente se ejecuta automáticamente después de instalar un nuevo kernel o nuevos módulos.
- `sudo depmod -a`: Vuelve a sondear todos los módulos instalados y regenera los archivos de dependencia. Es necesario ejecutarlo si instalas módulos manualmente en el directorio `/lib/modules/`.
- `modprobe` se basa en los archivos generados por `depmod`.

Configuración Persistente de Módulos:

Para asegurar que ciertos módulos se carguen o se configuren de una manera particular cada vez que el sistema arranca, se utilizan archivos de configuración:

1. **Archivos para Cargar Módulos al Inicio:**

- Define qué módulos deben cargarse durante el proceso de arranque.
- **Ubicación Estándar (Systemd):** Archivos `.conf` en `/etc/modules-load.d/`. Cada archivo contiene una lista de nombres de módulos (uno por línea) que se cargarán al arrancar.
- **Ubicaciones Antiguas/Alternativas:**

- **Rama Debian/Ubuntu:** El archivo `/etc/modules` (uno por línea) también es una ubicación común para cargar módulos al inicio en sistemas SysVinit o como una opción de compatibilidad en systemd.
- **Rama Red Hat/CentOS/Fedora:** Menos común usar un único archivo como `/etc/modules`. La configuración se maneja más a través de `/etc/modules-load.d/`.
- **Diferencias Debian vs. Red Hat (Carga al Inicio):** Ambas soportan `/etc/modules-load.d/`. Debian también mantiene `/etc/modules` por compatibilidad.

2. Archivos para Configurar Módulos:

- Define parámetros para los módulos o impide su carga (blacklist).
 - **Ubicación Estándar:** Archivos `.conf` en `/etc/modprobe.d/`.
 - **Directivas Comunes:**
 - `blacklist <nombre_módulo>`: Impide que el módulo se cargue automáticamente (por ejemplo, si un módulo causa problemas o si prefieres usar un driver alternativo).
 - `options <nombre_módulo> <parametro>=<valor> [<otro_parametro>=<valor>...]`: Establece parámetros para el módulo cuando se carga. Puedes ver los parámetros de un módulo con `modinfo <nombre_módulo>`.
 - `alias <nombre_alternativo> <nombre_módulo_real>`: Crea un alias para un módulo (menos común en la práctica administrativa básica).
 - **Ejemplo en `/etc/modprobe.d/no_pcspr.conf`:** `blacklist pcspr` (para desactivar el altavoz interno del PC).
 - **Ejemplo en `/etc/modprobe.d/options.conf`:** `options snd-hda-intel enable_msi=0` (establece un parámetro para el driver de audio Intel).
 - **Diferencias Debian vs. Red Hat (Configuración):** Ambas usan `/etc/modprobe.d/` y las directivas (`blacklist`, `options`) son las mismas.
-