

Seminario 1:

Linux Kernel Module

Por: Antonio Martín León

Diego Lerena Gracia

Arturo Cortés Sánchez

ÍNDICE

1. *¿Qué son los LKM?*
2. *¿Qué estructura tienen?*
3. *¿Cómo se construyen?*
4. *¿Qué ordenes suministra el sistema para manejarlos?*
5. *Ventajas e inconvenientes de los módulos de carga dinámica.*

¿Que son los LKM?

Un módulo de núcleo (LKM) es un mecanismo para añadir código al kernel de Linux , o eliminar código del mismo en tiempo de ejecución. Son ideales para los controladores de dispositivos, lo que permite que el kernel se comuniquen con el hardware sin que tenga que saber como funciona. Si esto no existiera, habría que añadir el código de todos y cada uno de los drivers al kernel.

¿Qué estructura tienen?

El código fuente del Kernel de Linux se estructura en varios directorios:

1. **arch**. Este directorio contiene archivos referentes a arquitecturas específicas. Dentro del directorio *arch/* existen subdirectorios para diferentes tipos de arquitecturas como x86, ARM, Motorola 64K, MIPS o SPARC.
2. **block**. Contiene la implementación de algoritmos de planificación de E/S necesarios para el uso de dispositivos de almacenamiento por bloques
3. **crypto**. Contiene la implementación de operaciones de cifrado y la API criptológica que es usada por ejemplo por los dispositivos WiFi para implementar algoritmos de cifrado
4. **Documentation**. Este directorio contiene la descripción de muchos subsistemas así como información valiosa sobre el funcionamiento del Kernel
5. **drivers**. Este directorio contiene multitud de subdirectorios con los controladores de numerosos dispositivos separados por clases de dispositivo. Es el directorio con mayor contenido de largo
6. **fs**. Contiene la implementación de los diferentes sistemas de archivos como EXT3, EXT4, [resiserfs](#), HFS, XFS, NTFS, FAT y otros
7. **include**. Los ficheros de cabecera del Kernel residen en este subdirectorio.
8. **init**. Contiene código de alto nivel de inicio y arranque
9. **ipc**. Contiene el código de soporte para la Comunicación entre Procesos (IPC) donde se implementan mecanismos como las colas de mensajes, los semáforos, y la memoria compartida

10. **Kernel**. Las porciones del Kernel independientes de la arquitectura se encuentran en este directorio
11. **lib**. Contiene el código que implementa rutinas de librería
12. **mm**. En este directorio se encuentra la implementación de los mecanismos de gestión de memoria
13. **net**. Contiene la implementación de los protocolos de red.
14. **scripts**. Scripts usados durante la construcción del Kernel
15. **sound**. El subsistema de audio de Linux se encuentra en este subdirectorio
16. **usr**. Contiene la implementación del sistema **initramfs**

¿Cómo se construyen?

Para crear un LKM primero tenemos que tener unas consideraciones previas. A la hora de compilar un Kernel es importante comenzar con la carpeta de trabajo bien limpia por lo que es siempre recomendable comenzar haciendo un make clean en el directorio principal del Kernel. Una vez tenemos la carpeta limpia debemos proceder a la configuración del Kernel, si el Kernel está configurado actualmente para soportar el archivo `/proc/config.gz` es recomendable usarlo como plantilla de configuración.

Si no lo tienen existen dos alternativas:

-Buscar la información referente a la compilación y configuración del Kernel para dicha distribución en su documentación.

-Y como **último recurso** siempre se puede copiar la configuración estándar de `arch/x86/configs/[i386|x86_64]defconfig` y utilizarla como base.

Para crear un LKM debemos seguir los siguientes pasos:

1. Instalar y preparar las cabeceras del kernel.
2. Crear el código fuente y el makefile para el LKM.
3. Compilar el código: ir al directorio del proyecto y tipear "make".
4. Insertar el módulo compilado en el Kernel en ejecución: para ello, use "insmod".
5. Retirar el módulo cuando se haya terminado: para ello, usar "rmmod". Para asegurarse de que ha sido eliminado revisar el "syslog".

¿Qué órdenes suministra el sistema para manejarlos?

- instalar/desinstalar

Linux provee de herramientas para la carga y descarga de módulos, estas son *modprobe* y *rmmod*.

- módulos disponibles

Se puede ver que los módulos ya están cargados en el sistema mediante la orden "lsmod", que obtiene su información mediante la lectura de los archivos `/proc/modules`

- dependencias entre ellos

El Kernel automáticamente carga aquellos módulos de los subsistemas presentes en la máquina o dispositivos que no hayan sido reclamados por ningún controlador aún. "modprobe", además carga todas las dependencias del módulo que queremos añadir buscándolas en el archivo de dependencias del Kernel (*modules.dep*). Si queremos es listar las dependencias de un modulo concreto, hay que usar la orden "modprobe --show-depends"

-propiedades del modulo

La orden "modinfo" devuelve toda la informacion relacionada con un modulo, si buscamos su configuracion, habrá que usar el comando "modprobe -c" y filtrar los resultados en busca del modulo deseado. Con esta sencilla linea "modprobe -c | grep *nombre_del_módulo*" podemos realizar ambas acciones

Ventajas e inconvenientes de los módulos de carga dinámica

-Las desventajas son que los LKM son muy lentos al cargarse y su funcionalidad tiene que ir en el kernel base. Además, no todos los componentes del kernel se pueden configurar como módulos, como por ejemplo, el manejo de la memoria.

- Las ventajas son que un sistema operativo sin módulos cargables en el núcleo debería tener toda aquella funcionalidad que pueda necesitar precompilada en el núcleo base, lo que haría que la imagen del núcleo fuera mucho mayor y ocuparía un gran espacio de memoria. Además, los usuarios deberían recompilar y reiniciar el núcleo cada vez que quisieran agregarle alguna funcionalidad.

WEBGRAFIA:

<http://www.abbreviationfinder.org/es/acronyms/lkm.html>

<http://www.genbetadev.com/software-libre-y-licencias/programando-modulos-para-el-kernel-de-linux>

http://wiki.inf.utfsm.cl/index.php?title=Introducci%C3%B3n_a_programaci%C3%B3n_de_m%C3%B3dulos_de_kernel_Linux#Estructura_del_c%C3%B3digo_fuente_del_Kernel_de_Linux

[https://wiki.archlinux.org/index.php/Kernel_modules_\(Espa%C3%B1ol\)#Manejar_m%C3%B3dulos_manualmente](https://wiki.archlinux.org/index.php/Kernel_modules_(Espa%C3%B1ol)#Manejar_m%C3%B3dulos_manualmente)

https://es.wikipedia.org/wiki/M%C3%B3dulo_de_n%C3%A1cleo

<http://stackoverflow.com/questions/14973040/disadvantages-of-linux-kernel-module>