



ugr

Universidad
de Granada

Escuela Técnica Superior de Ingenierías
Informática y de Telecomunicación

GRADO EN INGENIERÍA INFORMÁTICA

PERIFÉRICOS DE INTERFAZ HUMANA

Seminario 4: LKM

Autora:

DE LA VIEJA LAFUENTE, CLAUDIA

Curso:

2023-2024

Índice

1. Sistema LKM	2
1.1. Introducción	2
1.1.1. Diferencias entre módulos y aplicaciones	2
1.1.2. Módulos y controladores de dispositivos	2
1.1.3. Ventajas del uso de módulos	2
1.2. Modulo en C	3

1. Sistema LKM

1.1. Introducción

Un módulo cargable del kernel (loadable kernel module, LKM) es básicamente un archivo que tiene código objeto que puede enchufarse al kernel en funcionamiento (también conocido como kernel base) de un sistema operativo. La mayoría de los sistemas estilo Unix y también Microsoft Windows soportan estos módulos, aunque a veces les llaman de otra forma, como *.extensiones del kernel*.^{en} Mac OS X.

Los módulos cargables del kernel se usan generalmente para dar soporte a nuevos dispositivos de hardware y sistemas de archivos, además de añadir nuevas llamadas al sistema. Cuando ya no se necesita la funcionalidad que ofrece un módulo del kernel, normalmente se puede descargar, liberando la memoria que ocupaba.

Un ejemplo típico de un módulo cargable son los drivers de dispositivos.

1.1.1. Diferencias entre módulos y aplicaciones

Un módulo cargable es diferente de una aplicación en varias formas. Mientras que las aplicaciones pequeñas o medianas suelen hacer una tarea de principio a fin, un módulo cargable solo se registra en el kernel para manejar futuros requerimientos. El estilo de programación de un módulo es similar a la programación dirigida por eventos.

Otra diferencia importante es la tolerancia a fallos. Si una aplicación falla, normalmente no afecta al sistema operativo en general, pero si un módulo cargado en el kernel falla, puede dejar todo el sistema en un estado inestable o inutilizable.

Finalmente, los módulos se ejecutan en espacio de kernel, mientras que las aplicaciones se ejecutan en espacio de usuario.

1.1.2. Módulos y controladores de dispositivos

En el mundo de Linux, un uso común de los módulos cargables es para crear controladores de dispositivos de hardware.

Los controladores de dispositivos son súper importantes en el kernel de Linux. Básicamente, su chamba es ocultar los detalles técnicos de cómo funciona un dispositivo a nivel hardware. Esto permite que los usuarios interactúen con el hardware usando un conjunto estándar de comandos, sin importar qué controlador específico se esté usando detrás de escena.

La ventaja de desarrollar controladores para Linux es que se pueden hacer de forma independiente al resto del kernel y cargarlos solo cuando sean necesarios. Esta flexibilidad hace que escribir controladores de dispositivos para Linux sea mucho más sencillo.

1.1.3. Ventajas del uso de módulos

Un sistema operativo que no dispone de módulos cargables en el núcleo debe tener toda aquella funcionalidad que pueda llegar a ser necesitada precompilada dentro del núcleo base. El problema de este enfoque consiste en que en general la imagen del núcleo sería mucho mayor, ocupando un gran espacio memoria. Así mismo, resultaría necesario que los usuarios recompilaran y reiniciaran el núcleo base cada vez que se necesite agregar nueva funcionalidad al mismo.

1.2. Modulo en C

Este código define un módulo del kernel de Linux que muestra un mensaje cuando se carga (“Hola, mundo desde el kernel!”) y otro mensaje cuando se descarga (“Adiós, mundo desde el kernel!”). Estos mensajes se registran en los logs del sistema, y las funciones `init_module` y `cleanup_module` manejan la carga y descarga del módulo respectivamente.

```
PDIH > S_LKM > C miejemplo.c > MODULE_AUTHOR
1  /**
2   * @file miejemplo.c
3   * @author Claudia de la Vieja Lafuente
4   * @date 3 de junio de 2024
5   * @version 0.1
6   * @brief An introductory loadable kernel module (LKM).
7   */
8
9  #include <linux/module.h>
10 #include <linux/kernel.h>
11
12 int init_module(void) {
13     printk(KERN_INFO "Hola mundo desde el kernel!\n");
14     return 0; // 0 significa que está todo bien, cualquier otro valor indica error
15 }
16
17 void cleanup_module(void) {
18     printk(KERN_INFO "Adiós mundo desde el kernel!\n");
19 }
20
21 MODULE_LICENSE("GPL");
22 MODULE_DESCRIPTION("Un módulo sencillo de ejemplo");
23 MODULE_AUTHOR("Claudia de la Vieja");
24
```

Figura 1: Código

```
→ S_LKM git:(main) X make
make -C /lib/modules/6.5.0-35-generic/build/ M=/home/claudia/PDIH/S_LKM modules
make[1]: se entra en el directorio '/usr/src/linux-headers-6.5.0-35-generic'
warning: the compiler differs from the one used to build the kernel
The kernel was built by: x86_64-linux-gnu-gcc-12 (Ubuntu 12.3.0-1ubuntu1-22.04) 12.3.0
You are using: gcc-12 (Ubuntu 12.3.0-1ubuntu1-22.04) 12.3.0
CC [M] /home/claudia/PDIH/S_LKM/miejemplo.o
MODPOST /home/claudia/PDIH/S_LKM/Module.symvers
CC [M] /home/claudia/PDIH/S_LKM/miejemplo.mod.o
LD [M] /home/claudia/PDIH/S_LKM/miejemplo.ko
BTF [M] /home/claudia/PDIH/S_LKM/miejemplo.ko
Skipping BTF generation for /home/claudia/PDIH/S_LKM/miejemplo.ko due to unavailability of vmlinux
make[1]: se sale del directorio '/usr/src/linux-headers-6.5.0-35-generic'
```

Figura 2: Make

Despues de hacer make hay que cargar el nuevo modulo en el kernel con **sudo insmod miejemplo.ko**. Una vez cargado ejecutamos la orden **lsmod** para comprobar que se ha cargado correctamente.

```
→ S_LKM git:(main) X lsmod
Module              Size  Used by
miejemplo           12288  0
```

Figura 3: lsmod

```
→ S_LKM git:(main) X modinfo miejemplo.ko
filename:       /home/claudia/PDIH/S_LKM/miejemplo.ko
author:        Claudia de la Vieja
description:    Un módulo sencillo de ejemplo
license:       GPL
srcversion:    2E5E949DC8E608B44F59DD5
depends:
retpoline:     Y
name:          miejemplo
vermagic:      6.5.0-35-generic SMP preempt mod_unload modversions
```

Figura 4: modinfo para saber información sobre el módulo

Después de ver la información del modulo lo eliminamos usando el comando **usod rmmmod** y revisamos la salida de la función **printk()** en el registro de log del kernel

```
Jun  3 18:52:34 claudia-laptop kernel: [14837.969601] Hola, mundo desde el kernel!
Jun  3 18:53:41 claudia-laptop kernel: [14904.662386] Adiós, mundo desde el kernel!
```

Figura 5: Log del kernel