

Seminario 4. Loadable kernel module (LKM)

Alberto Socas Mendoza – Periféricos y Dispositivos de Interfaz Humana

Para comenzar a trabajar con los módulos Kernel, hay que tener instaladas ciertas cabeceras de Linux, comencemos por instalarlas:

```
albertosocas@albertosocas-VirtualBox:~$ sudo apt-get install linux-headers-$(uname -r)
[sudo] contraseña para albertosocas:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
linux-headers-5.8.0-48-generic ya está en su versión más reciente (5.8.0-48.54~20.04.1).
fijado linux-headers-5.8.0-48-generic como instalado manualmente.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 55 no actualizados
.
```

Ilustración 1

Con la línea de código **sudo apt-get install linux-headers-\$(uname -r)** instalamos los módulos de Linux necesarios para empezar a trabajar con ellos.

Ahora continuaremos mirando el código que vamos a compilar, en este caso **hello.c**:

```
albertosocas@albertosocas-VirtualBox:~/Escritorio/S4ejemplos$ cat hello.c
/**
```

Ilustración 2

Cuando se cargue este código, se ejecutará la primera función **helloBBB_init()**, y cuando se descargue se ejecutará la función **helloBBB_exit()**.

Para compilar el código usaremos **Makefile**, lo que hace el proceso de compilación muy sencillo:

```
albertosocas@albertosocas-VirtualBox:~/Escritorio/S4ejemplos$ make
make -C /lib/modules/5.8.0-48-generic/build/ M=/home/albertosocas/Escritorio/S4ejemplos modules
make[1]: se entra en el directorio '/usr/src/linux-headers-5.8.0-48-generic'
CC [M] /home/albertosocas/Escritorio/S4ejemplos/hello.o
MODPOST /home/albertosocas/Escritorio/S4ejemplos/Module.symvers
CC [M] /home/albertosocas/Escritorio/S4ejemplos/hello.mod.o
LD [M] /home/albertosocas/Escritorio/S4ejemplos/hello.ko
make[1]: se sale del directorio '/usr/src/linux-headers-5.8.0-48-generic'
```

Ilustración 3

Tras ejecutar el makefile nos encontraremos en nuestro directorio un LKM llamado **hello.ko**, como vemos a continuación:

```
albertosocas@albertosocas-VirtualBox:~/Escritorio/S4ejemplos$ ls -l
total 36
-rwxrwxrwx 1 albertosocas albertosocas 2424 may 11 2020 hello.c
-rw-rw-r-- 1 albertosocas albertosocas 5704 abr 22 12:19 hello.ko
-rw-rw-r-- 1 albertosocas albertosocas 50 abr 22 12:19 hello.mod
-rw-rw-r-- 1 albertosocas albertosocas 801 abr 22 12:19 hello.mod.c
-rw-rw-r-- 1 albertosocas albertosocas 3192 abr 22 12:19 hello.mod.o
-rw-rw-r-- 1 albertosocas albertosocas 3424 abr 22 12:19 hello.o
-rwxrwxrwx 1 albertosocas albertosocas 154 may 11 2020 Makefile
-rw-rw-r-- 1 albertosocas albertosocas 50 abr 22 12:19 modules.order
-rw-rw-r-- 1 albertosocas albertosocas 0 abr 22 12:19 Module.symvers
albertosocas@albertosocas-VirtualBox:~/Escritorio/S4ejemplos$
```

Ilustración 4

A continuación, comenzaremos a probar el módulo LKM con ciertas herramientas.

La primera herramienta que vamos a utilizar es (**insmod**), que insertará el nuevo módulo en kernel:

```
albertosocas@albertosocas-VirtualBox:~/Escritorio/S4ejemplos$ sudo insmod hello.ko
[sudo] contraseña para albertosocas:
albertosocas@albertosocas-VirtualBox:~/Escritorio/S4ejemplos$ lsmod
Module                  Size  Used by
hello                   16384  0
btrfs                  1290240  0
blake2b_generic         20480  0
```

Ilustración 5

La segunda herramienta será la (**modinfo**), que nos dará información sobre el módulo, como comprobaremos en la siguiente imagen:

```
albertosocas@albertosocas-VirtualBox:~/Escritorio/S4ejemplos$ modinfo hello.ko
filename:               /home/albertosocas/Escritorio/S4ejemplos/hello.ko
version:                 0.1
description:             A simple Linux driver for the BBB.
author:                 Derek Molloy
license:                 GPL
srcversion:              0DD9FE0DE42157F9221E608
depends:
retpoline:              Y
name:                   hello
vermagic:               5.8.0-48-generic SMP mod_unload modversions
parm:                   name:The name to display in /var/log/kern.log (charp)
```

Ilustración 6

Finalmente, lo eliminamos del kernel con la herramienta (**rmmod**) y revisaremos la salida de la función **printk()**:

```
albertosocas@albertosocas-VirtualBox:~/Escritorio/S4ejemplos$ sudo su -  
root@albertosocas-VirtualBox:~# cd /var/log  
root@albertosocas-VirtualBox:/var/log# tail -f kern.log  
Apr 22 12:22:47 albertosocas-VirtualBox kernel: [14559.670724] EBB: Hello wo  
rld from the BBB LKM!
```

Ilustración 7

Vemos en la Ilustración 7, como se ejecutó la función **HelloBBB_ini()**, imprimiendo en pantalla *“Hello world from the BBB LKM!”*

Finalmente, cuando se descarga, se ejecutaba la función **helloBBB_exit()**, la que mostraba en pantalla *“Goodbye world from the BBB LKM!”*

```
Apr 22 12:24:47 albertosocas-VirtualBox kernel: [14679.700340] EBB: Goodbye  
world from the BBB LKM!
```

Ilustración 8