



ugr

Universidad
de Granada

PDIH

PERIFÉRICOS Y DISPOSITIVOS DE INTERFAZ HUMANA.

Seminario: Módulos cargables del kernel (LKM).

Autora: Cristina María Crespo Arco

Correo: cmcrespo@correo.ugr.es

Profesor: Pedro A. Castillo Valdivieso



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

—
Curso 2021 - 2022

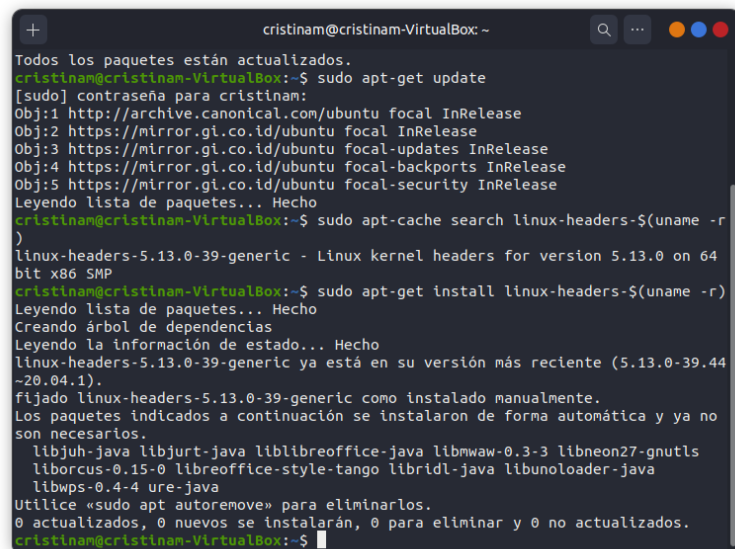
Índice

1. Preparar el sistema para construir un LKM.	2
2. Código del módulo creado	2
3. Probar el módulo LKM	4

1. Preparar el sistema para construir un LKM.

Lo primero que hay que hacer es instalar las cabeceras de Linux, esto se realiza mediante los tres comando siguientes:

```
$ sudo apt-get update
$ sudo apt-cache search linux-headers-$(uname -r)
$ sudo apt-get install linux-headers-$(uname -r)
```



```
cristinam@cristinam-VirtualBox: ~
Todos los paquetes están actualizados.
cristinam@cristinam-VirtualBox:~$ sudo apt-get update
[sudo] contraseña para cristinam:
Obj:1 http://archive.canonical.com/ubuntu focal InRelease
Obj:2 https://mirror.gi.co.id/ubuntu focal InRelease
Obj:3 https://mirror.gi.co.id/ubuntu focal-updates InRelease
Obj:4 https://mirror.gi.co.id/ubuntu focal-backports InRelease
Obj:5 https://mirror.gi.co.id/ubuntu focal-security InRelease
Leyendo lista de paquetes... Hecho
cristinam@cristinam-VirtualBox:~$ sudo apt-cache search linux-headers-$(uname -r)
)
linux-headers-5.13.0-39-generic - Linux kernel headers for version 5.13.0 on 64
bit x86 SMP
cristinam@cristinam-VirtualBox:~$ sudo apt-get install linux-headers-$(uname -r)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
linux-headers-5.13.0-39-generic ya está en su versión más reciente (5.13.0-39.44
~20.04.1).
Fijado linux-headers-5.13.0-39-generic como instalado manualmente.
Los paquetes indicados a continuación se instalaron de forma automática y ya no
son necesarios.
 libjuh-java libjurt-java liblibreoffice-java libmwaw-0.3-3 libneon27-gnutls
 liborcus-0.15-0 libreoffice-style-tango libridl-java libunolodner-java
 libwps-0.4-4 ure-java
Utilice «sudo apt autoremove» para eliminarlos.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
cristinam@cristinam-VirtualBox:~$
```

Figura 1: Preparar el sistema

2. Código del módulo creado

He implementado un código sencillo que muestra por pantalla “EBB: Hola mundo desde el BBB LKM!” y “EBB: Adios mundo desde el BBB LKM!” cuando se ejecuta la función *printk()* en el registro del log del kernel.

```
#include <linux/init.h>
#include <linux/module.h>
#include <linux/kernel.h>

MODULE_LICENSE("GPL"); ///< tipo de licencia
MODULE_AUTHOR("Cristina Maria"); ///< Autora
MODULE_DESCRIPTION("Un simple controlador de Linux."); ///< descripción
MODULE_VERSION("0.1"); ///< Versión del módulo

static char *name = "mundo";

module_param(name, charp, S_IRUGO);
MODULE_PARM_DESC(name, "El nombre para mostrar en el registro");
```

```

/** @brief Funcion inicialización del LKM
 * @return returns 0 if successful
 */
static int __init helloBBB_init(void){
    printk(KERN_INFO "EBB: Hola %s desde el BBB LKM!\n", name);
    return 0;
}

/** @brief Función de limpieza LKM
 */
static void __exit helloBBB_exit(void){
    printk(KERN_INFO "EBB: Adios %s desde el BBB LKM!\n", name);
}

/** @brief Identificar la función de inicialización en el momento de la inserción y
 */
module_init(helloBBB_init);
module_exit(helloBBB_exit);

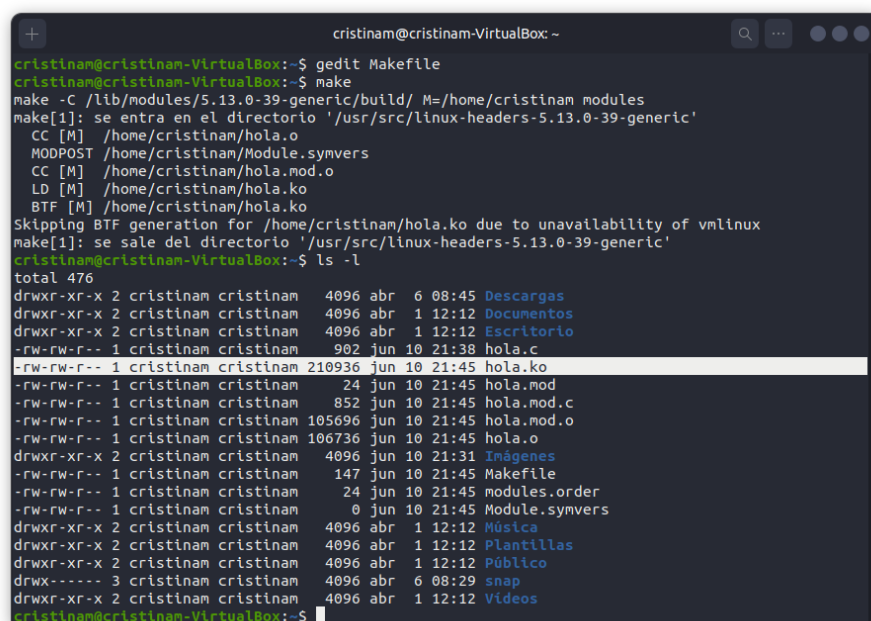
```

Además, he generado un Makefile para poder compilar el archivo *hola.c* y obtener un fichero LKM llamado *hola.ko*. El código del Makefile es el siguiente:

```

obj-m+=hola.o
all:
make -C /lib/modules/$(shell uname -r)/build/ M=$(PWD) modules
clean:
make -C /lib/modules/$(shell uname -r)/build/ M=$(PWD)

```



```

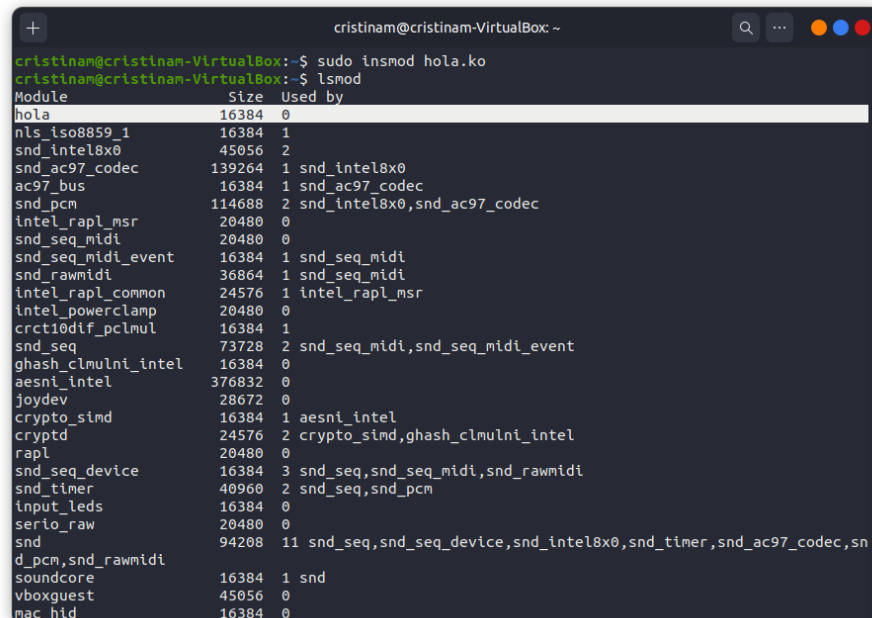
cristinam@cristinam-VirtualBox: ~
cristinam@cristinam-VirtualBox:~$ gedit Makefile
cristinam@cristinam-VirtualBox:~$ make
make -C /lib/modules/5.13.0-39-generic/build/ M=/home/cristinam modules
make[1]: se entra en el directorio '/usr/src/linux-headers-5.13.0-39-generic'
  CC [M] /home/cristinam/hola.o
  MODPOST /home/cristinam/Module.symvers
  CC [M] /home/cristinam/hola.mod.o
  LD [M] /home/cristinam/hola.ko
  BTF [M] /home/cristinam/hola.ko
Skipping BTF generation for /home/cristinam/hola.ko due to unavailability of vmlinux
make[1]: se sale del directorio '/usr/src/linux-headers-5.13.0-39-generic'
cristinam@cristinam-VirtualBox:~$ ls -l
total 476
drwxr-xr-x 2 cristinam cristinam 4096 abr  6 08:45 Descargas
drwxr-xr-x 2 cristinam cristinam 4096 abr  1 12:12 Documentos
drwxr-xr-x 2 cristinam cristinam 4096 abr  1 12:12 Escritorio
-rw-rw-r-- 1 cristinam cristinam 902 jun 10 21:38 hola.c
-rw-rw-r-- 1 cristinam cristinam 210936 jun 10 21:45 hola.ko
-rw-rw-r-- 1 cristinam cristinam 24 jun 10 21:45 hola.mod
-rw-rw-r-- 1 cristinam cristinam 852 jun 10 21:45 hola.mod.c
-rw-rw-r-- 1 cristinam cristinam 105696 jun 10 21:45 hola.mod.o
-rw-rw-r-- 1 cristinam cristinam 106736 jun 10 21:45 hola.o
drwxr-xr-x 2 cristinam cristinam 4096 jun 10 21:31 Imágenes
-rw-rw-r-- 1 cristinam cristinam 147 jun 10 21:45 Makefile
-rw-rw-r-- 1 cristinam cristinam 24 jun 10 21:45 modules.order
-rw-rw-r-- 1 cristinam cristinam 0 jun 10 21:45 Module.symvers
drwxr-xr-x 2 cristinam cristinam 4096 abr  1 12:12 Música
drwxr-xr-x 2 cristinam cristinam 4096 abr  1 12:12 Plantillas
drwxr-xr-x 2 cristinam cristinam 4096 abr  1 12:12 Público
drwx----- 3 cristinam cristinam 4096 abr  6 08:29 snap
drwxr-xr-x 2 cristinam cristinam 4096 abr  1 12:12 Videos
cristinam@cristinam-VirtualBox:~$

```

Figura 2: Ejecutar Makefile

3. Probar el módulo LKM

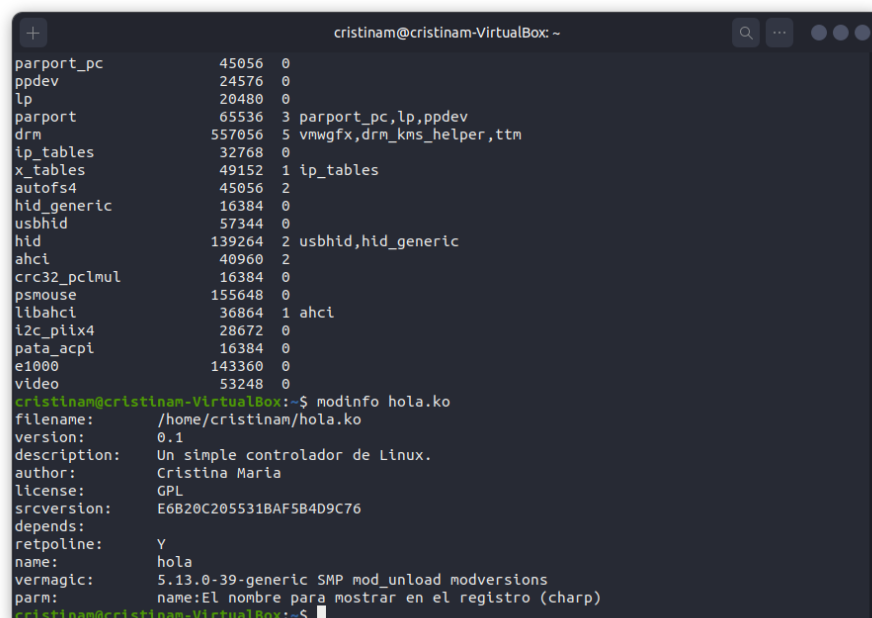
Para probar el módulo LKM que acabo de crear lo primero que he hecho ha sido cargar el módulo, mediante el uso del comando `$ sudo insmod hola.ko`. Y, a continuación, comprobamos que se ha creado correctamente con el comando `$ lsmod`



```
cristinam@cristinam-VirtualBox: ~  
cristinam@cristinam-VirtualBox:~$ sudo insmod hola.ko  
cristinam@cristinam-VirtualBox:~$ lsmod  
Module              Size  Used by  
hola                 16384  0  
nls_iso8859_1       16384  1  
snd_intel8x0        45056  2  
snd_ac97_codec     139264  1 snd_intel8x0  
ac97_bus            16384  1 snd_ac97_codec  
snd_pcm            114688  2 snd_intel8x0,snd_ac97_codec  
intel_rapl_msr      20480  0  
snd_seq_midi        20480  0  
snd_seq_midi_event  16384  1 snd_seq_midi  
snd_rawmidi         36864  1 snd_seq_midi  
intel_rapl_common   24576  1 intel_rapl_msr  
intel_powerclamp    20480  0  
crc10dif_pclmul     16384  1  
snd_seq             73728  2 snd_seq_midi,snd_seq_midi_event  
ghash_clmulni_intel 16384  0  
aesni_intel        376832  0  
joydev              28672  0  
crypto_simd         16384  1 aesni_intel  
cryptd              24576  2 crypto_simd,ghash_clmulni_intel  
rapl                20480  0  
snd_seq_device      16384  3 snd_seq,snd_seq_midi,snd_rawmidi  
snd_timer           40960  2 snd_seq,snd_pcm  
input_leds          16384  0  
serio_raw           20480  0  
snd                 94208  11 snd_seq,snd_seq_device,snd_intel8x0,snd_timer,snd_ac97_codec,snd  
d_pcm,snd_rawmidi  
soundcore           16384  1 snd  
vboxguest           45056  0  
mac_hid             16384  0
```

Figura 3: Cargar el módulo LKM

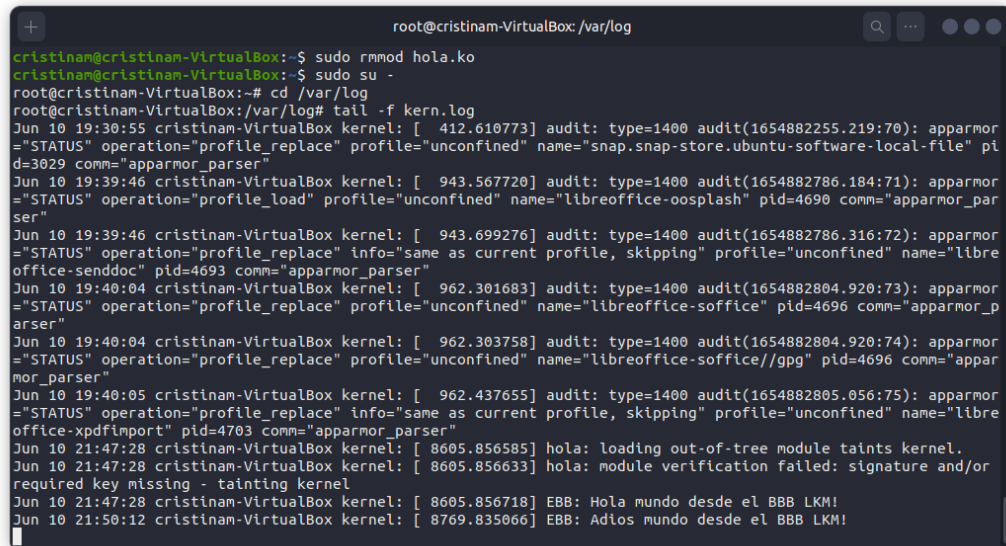
Para obtener la información del módulo podemos ejecutar el comando `$ modinfo hola.ko`



```
cristinam@cristinam-VirtualBox: ~  
cristinam@cristinam-VirtualBox:~$ modinfo hola.ko  
filename:           /home/cristinam/hola.ko  
version:            0.1  
description:        Un simple controlador de Linux.  
author:             Cristina Maria  
license:            GPL  
srcversion:         E6B20C205531BAF5B4D9C76  
depends:               
retpoline:          Y  
name:               hola  
vermagic:           5.13.0-39-generic SMP mod_unload modversions  
parm:               name:El nombre para mostrar en el registro (charp)  
cristinam@cristinam-VirtualBox:~$
```

Figura 4: Información sobre el módulo LKM

Por último, hay que eliminar el módulo del kernel con el comando `$ sudo rmmod hola.ko` y comprobamos la salida de la función `printk()` en el registro del kernel. Para comprobar la salida hay que iniciarse con el usuario root (`$ sudo su -`) y moverse al directorio `/var/log` y ejecutar el comando `$ tail -f kern.log`



```
root@cristinam-VirtualBox: /var/log
cristinam@cristinam-VirtualBox:~$ sudo rmmod hola.ko
cristinam@cristinam-VirtualBox:~$ sudo su -
root@cristinam-VirtualBox:~# cd /var/log
root@cristinam-VirtualBox:/var/log# tail -f kern.log
Jun 10 19:30:55 cristinam-VirtualBox kernel: [ 412.610773] audit: type=1400 audit(1654882255.219:70): apparmor
="STATUS" operation="profile_replace" profile="unconfined" name="snap.snap-store.ubuntu-software-local-file" pi
d=3029 comm="apparmor_parser"
Jun 10 19:39:46 cristinam-VirtualBox kernel: [ 943.567720] audit: type=1400 audit(1654882786.184:71): apparmor
="STATUS" operation="profile_load" profile="unconfined" name="libreoffice-oosplash" pid=4690 comm="apparmor_p
arser"
Jun 10 19:39:46 cristinam-VirtualBox kernel: [ 943.699276] audit: type=1400 audit(1654882786.316:72): apparmor
="STATUS" operation="profile_replace" info="same as current profile, skipping" profile="unconfined" name="libre
office-senddoc" pid=4693 comm="apparmor_parser"
Jun 10 19:40:04 cristinam-VirtualBox kernel: [ 962.301693] audit: type=1400 audit(1654882804.920:73): apparmor
="STATUS" operation="profile_replace" profile="unconfined" name="libreoffice-soffice" pid=4696 comm="apparmor_p
arser"
Jun 10 19:40:04 cristinam-VirtualBox kernel: [ 962.303758] audit: type=1400 audit(1654882804.920:74): apparmor
="STATUS" operation="profile_replace" profile="unconfined" name="libreoffice-soffice//gpg" pid=4696 comm="appar
mor_parser"
Jun 10 19:40:05 cristinam-VirtualBox kernel: [ 962.437655] audit: type=1400 audit(1654882805.056:75): apparmor
="STATUS" operation="profile_replace" info="same as current profile, skipping" profile="unconfined" name="libre
office-xpdfimport" pid=4703 comm="apparmor_parser"
Jun 10 21:47:28 cristinam-VirtualBox kernel: [ 8605.856585] hola: loading out-of-tree module taints kernel.
Jun 10 21:47:28 cristinam-VirtualBox kernel: [ 8605.856633] hola: module verification failed: signature and/or
required key missing - tainting kernel
Jun 10 21:47:28 cristinam-VirtualBox kernel: [ 8605.856718] EBB: Hola mundo desde el BBB LKM!
Jun 10 21:50:12 cristinam-VirtualBox kernel: [ 8769.835066] EBB: Adios mundo desde el BBB LKM!
```

Figura 5: Eliminar el módulo del kernel y mostrar salida función `printk()`