

SEMNARIO 4.PERIFÉRICOS Y DISPOSITIVOS INTERFAZ HUMANA

Antonio Jesús Ruiz Gómez.

1.Probamos el módulo LKM

Comprobamos que tenemos instaladas las cabeceras de Linux.

Con el primer comando obtenemos la versión exacta del kernel linux que tenemos instalada y con el segundo comando instalamos las cabeceras de nuestra versión kernel :

```
antoniojrg@antoniojrg:~/Escritorio$ uname -a
Linux antoniojrg 4.15.0-106-generic #107-Ubuntu SMP Thu Jun 4 11:27:52 UTC
2020 x86_64 x86_64 x86_64 GNU/Linux
antoniojrg@antoniojrg:~/Escritorio$ uname -r
4.15.0-106-generic
antoniojrg@antoniojrg:~/Escritorio$ sudo apt-get install linux-headers-4.1
5.0-106-generic
```

Cogemos del seminario los códigos hello.c y su correspondiente Makefile para probarlos.

```
antoniojrg@antoniojrg:~$ curl https://raw.githubusercontent.com/derekmolloy/exploringBB/master/extras/kernel/hello/Makefile > Makefile
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Cu
rrent
                                Dload  Upload  Total  Spent  Left  Sp
eed
  0     0    0     0    0     0     0      0  --:--:-- --:--:-- --:--:--
100  154  100  154    0     0   980      0  --:--:-- --:--:-- --:--:--
 980
```

Usamos la orden make para generar el archivo hello.ko, que es el LKM compilado.

```
antoniojrg@antoniojrg:~$ uname -a
Linux antoniojrg 4.15.0-106-generic #107-Ubuntu SMP Thu Jun 4 11:27:52 UTC
2020 x86_64 x86_64 x86_64 GNU/Linux
antoniojrg@antoniojrg:~$ make
make -C /lib/modules/4.15.0-106-generic/build/ M=/home/antoniojrg/modules
make[1]: se entra en el directorio '/usr/src/linux-headers-4.15.0-106-gene
ric'
  CC [M]  /home/antoniojrg/hello.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/antoniojrg/hello.mod.o
  LD [M]  /home/antoniojrg/hello.ko
make[1]: se sale del directorio '/usr/src/linux-headers-4.15.0-106-gene
ric'
```

Aquí podemos ver todo lo que nos ha generado.

```
-rw-rw-r-- 1 antoniojrg antoniojrg      2422 may 13 12:45 hello.c
-rw-r--r-- 1 antoniojrg antoniojrg      5112 may 13 13:12 hello.ko
-rw-r--r-- 1 antoniojrg antoniojrg        603 may 13 13:12 hello.mod.c
-rw-r--r-- 1 antoniojrg antoniojrg      2576 may 13 13:12 hello.mod.o
-rw-r--r-- 1 antoniojrg antoniojrg      4840 may 13 13:12 hello.o
drwxr-xr-x 3 antoniojrg antoniojrg      4096 nov 15  2017 Imágenes
-rw-rw-r-- 1 antoniojrg antoniojrg        153 may 13 13:01 make
-rw-r--r-- 1 antoniojrg antoniojrg        154 may 13 13:11 Makefile
-rw-r--r-- 1 antoniojrg antoniojrg         33 may 13 13:12 modules.order
-rw-r--r-- 1 antoniojrg antoniojrg         0 may 13 13:12 Module.symvers
```

Cargamos el nuevo módulo en el kernel con **sudo insmod hello.ko** , con el comando **lsmod | grep hello** podemos ver si está cargado el nuevo módulo y con **modinfo hello.ko** vemos la información del nuevo módulo.

```
antoniojrg@antoniojrg:~$ sudo insmod hello.ko
antoniojrg@antoniojrg:~$ lsmod | grep hello
hello                16384  0
antoniojrg@antoniojrg:~$ modinfo hello.ko
filename:           /home/antoniojrg/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:          4.15.0-106-generic SMP mod_unload
parm:              name:The name to display in /var/log/kern.log (charp)
```

Por último usamos el comando **sudo rmmod hello.ko** que sirve para descargarlo del kernel, y comprobamos en el log del kernel como el LKM había sido cargado y descargado del kernel, ya que podemos verlo en los mensajes.

```
antoniojrg@antoniojrg:/var/log$ tail -f kern.log
May 13 13:19:21 antoniojrg kernel: [ 3446.146405] hello: loading out-of-tree module taints kernel.
May 13 13:19:21 antoniojrg kernel: [ 3446.146990] EBB: Hello world from the BBB LKM!
May 13 13:22:31 antoniojrg kernel: [ 3636.249661] EBB: Goodbye world from the BBB LKM!
```