

# Seminario 4. Módulos cargables del kernel (LKM)

El primer paso a realizar en este seminario es comprobar si tenemos instaladas las cabeceras de Linux y, si no están instaladas, instalarlas. Para ello, utilizaremos los siguientes comandos en nuestra terminal:

```
$ uname -r
```

```
$ sudo apt-get install linux-headers-5.4.0-65-generic
```

```
(base) vicky@vicky:~$ uname -r
5.4.0-105-generic
(base) vicky@vicky:~$ sudo apt-get install linux-headers-5.4.0-65-generic
[sudo] contraseña para vicky:
Lo sentimos, vuelva a intentarlo.
[sudo] contraseña para vicky:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
ant ant-optional enchant fastjar fonts-lato gnome-software-common jarwrapper
javascript-common junit4 libapache-pom-java libappstream-glib8
libclone-perl libcommon-sense-perl libcommons-beanutils-java
libcommons-cli-java libcommons-codec-java libcommons-collections3-java
libcommons-compress-java libcommons-dbc-java libcommons-digester-java
libcommons-httpclient-java libcommons-logging-java libcommons-parent-java
libcommons-pool-java libdb-java libdb-je-java libdb5.3-java
libdb5.3-java-jni libecj-java libel-api-java libenchmark2a
libequinox-osgi-java libfelix-bundle-repository-java
libfelix-gogo-command-java libfelix-gogo-runtime-java
libfelix-gogo-shell-java libhamcrest-java libhwaloc5 libindicator3-7
libjetty9-java libjline-java libjs-jquery libjsch-java libjson-xs-perl
libjsp-api-java libjtidy-java libjzlib-java libnfs11 libosgi-annotation-java
libosgi-compendium-java libosgi-core-java libplacebo7 libprotobuf-lite10
libqt5script5 libregexp-java libruby2.7 libservlet-api-java
libservlet3.1-java libtypes-serialiser-perl libwebsocket-api-java
linux-generic-hwe-18.04 linux-headers-5.4.0-96
linux-headers-5.4.0-96-generic linux-headers-generic-hwe-18.04
linux-image-5.4.0-96-generic linux-image-generic-hwe-18.04
linux-modules-5.4.0-96-generic linux-modules-extra-5.4.0-96-generic
python-pkg-resources python-setuptools python3-asn1crypto rake ruby
ruby-minitest ruby-net-telnet ruby-power-assert ruby-test-unit ruby-xmllrpc
ruby2.7 rubygems-integration sat4j
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
linux-headers-5.4.0-65
Se instalarán los siguientes paquetes NUEVOS:
linux-headers-5.4.0-65 linux-headers-5.4.0-65-generic
```

El siguiente paso será alterar el fichero *hello.c*, cambiando los datos correspondientes, de tal forma que nos quedaría el siguiente código:

```
11
12 #include <linux/init.h>           // Macros used to mark up functions e.g., __init __exit
13 #include <linux/module.h>         // Core header for loading LKMs into the kernel
14 #include <linux/kernel.h>         // Contains types, macros, functions for the kernel
15
16 MODULE_LICENSE("GPL");            ///< The license type -- this affects runtime behavior
17 MODULE_AUTHOR("Victoria Duenas Salcedo"); ///< The author -- visible when you use modinfo
18 MODULE_DESCRIPTION("Un programita facil de prueba"); ///< The description -- see modinfo
19 MODULE_VERSION("1.0");            ///< The version of the module
20
21 static char *name = "Victoria";    ///< An example LKM argument -- default value is "world"
22 module_param(name, charp, S_IRUGO); ///< Param desc. charp = char ptr, S_IRUGO can be read/not changed
23 MODULE_PARAM_DESC(name, "The name to display in /var/log/kern.log"); ///< parameter description
24
25 /** @brief The LKM initialization function
26  * The static keyword restricts the visibility of the function to within this C file. The __init
27  * macro means that for a built-in driver (not a LKM) the function is only used at initialization
28  * time and that it can be discarded and its memory freed up after that point.
29  * @return returns 0 if successful
30  */
31 static int __init helloBBB_init(void){
32     printk(KERN_INFO "Hola %s desde el kernel\n", name);
33     return 0;
34 }
35
36 /** @brief The LKM cleanup function
37  * Similar to the initialization function, it is static. The __exit macro notifies that if this
38  * code is used for a built-in driver (not a LKM) that this function is not required.
39  */
40 static void __exit helloBBB_exit(void){
41     printk(KERN_INFO "Adios %s desde el kernel\n", name);
42 }
43
44 /** @brief A module must use the module_init() module_exit() macros from linux/init.h, which
45  * identify the initialization function at insertion time and the cleanup function (as
46  * listed above)
47  */
48 module_init(helloBBB_init);
49 module_exit(helloBBB_exit);
50
```

Para la compilación del programa utilizamos el *Makefile* proporcionado por el profesor, dando lugar a un archivo *.ko*.

```
hello.c  Makefile  proceso.txt
home > vicky > Escritorio > PDIH > Seminarios > S4 > modulos-cargables-nucleo > Makefile
1  obj-m+=hello.o
2
3  all:
4      make -C /lib/modules/$(shell uname -r)/build/ M=$(PWD) modules
5  clean:
6      make -C /lib/modules/$(shell uname -r)/build/ M=$(PWD) clean
7
```

```
(base) vicky@vicky:~/Escritorio/PDIH/Seminarios/S4/modulos-cargables-nucleo$ make
make -C /lib/modules/5.4.0-105-generic/build/ M=/home/vicky/Escritorio/PDIH/Seminarios/S4/modulos-cargables-nucleo modules
make[1]: se entra en el directorio '/usr/src/linux-headers-5.4.0-105-generic'
CC [M] /home/vicky/Escritorio/PDIH/Seminarios/S4/modulos-cargables-nucleo/hello.o
Building modules, stage 2.
MODPOST 1 modules
CC [M] /home/vicky/Escritorio/PDIH/Seminarios/S4/modulos-cargables-nucleo/hello.mod.o
LD [M] /home/vicky/Escritorio/PDIH/Seminarios/S4/modulos-cargables-nucleo/hello.ko
make[1]: se sale del directorio '/usr/src/linux-headers-5.4.0-105-generic'
(base) vicky@vicky:~/Escritorio/PDIH/Seminarios/S4/modulos-cargables-nucleo$
```

```
(base) vicky@vicky:~/Escritorio/PDIH/Seminarios/S4/modulos-cargables-nucleo$ ls -l
total 40
-rw-r--r-- 1 vicky vicky 2427 abr  1 19:23 hello.c
-rw-rw-r-- 1 vicky vicky 5600 abr  1 20:28 hello.ko
-rw-rw-r-- 1 vicky vicky  76 abr  1 20:28 hello.mod
-rw-rw-r-- 1 vicky vicky  776 abr  1 20:28 hello.mod.c
-rw-rw-r-- 1 vicky vicky 3192 abr  1 20:28 hello.mod.o
-rw-rw-r-- 1 vicky vicky 3320 abr  1 20:28 hello.o
-rw-r--r-- 1 vicky vicky  154 may 11 2020 Makefile
-rw-rw-r-- 1 vicky vicky  76 abr  1 20:28 modules.order
-rw-rw-r-- 1 vicky vicky   0 abr  1 20:28 Module.symvers
-rw-r--r-- 1 vicky vicky  413 abr  1 11:41 proceso.txt
(base) vicky@vicky:~/Escritorio/PDIH/Seminarios/S4/modulos-cargables-nucleo$
```

Continuamos utilizando los siguientes comandos en la terminal para probar el módulo LKM:

\$ *sudo insmod hello.ko*: insertar el módulo en el kernel.

\$ *lsmod*: comprobar si se ha cargado el módulo en el kernel.

\$ *modinfo hello.ko*: solicitar información sobre el módulo (definida en el código *hello.c*).

```
(base) vicky@vicky:~/Escritorio/PDIH/Seminarios/S4/modulos-cargables-nucleo$ sudo insmod hello.ko
[sudo] contraseña para vicky:
(base) vicky@vicky:~/Escritorio/PDIH/Seminarios/S4/modulos-cargables-nucleo$ lsmod
Module                  Size  Used by
hello                   16384  0
ses                     20480  0
enclosure               16384  1 ses
scsi_transport_sas      36864  1 ses
uas                     28672  0
usb_storage             77824  1 uas
vboxnetadp              28672  0
vboxnetflt              28672  0
vboxdrv                 491520 2 vboxnetadp,vboxnetflt
rfcomm                  81920  4
ccm                     20480  3
inotify_filter          16384  0
```

```
(base) vicky@vicky:~/Escritorio/PDIH/Seminarios/S4/modulos-cargables-nucleo$ modinfo hello.ko
filename:      /home/vicky/Escritorio/PDIH/Seminarios/S4/modulos-cargables-nucleo/hello.ko
version:      1.0
description:   Un programita facil de prueba
author:       Victoria Duenas Salcedo
license:      GPL
srcversion:   5BF49F19D581FE8A3E97BD1
depends:
retpoline:    Y
name:         hello
vermagic:     5.4.0-105-generic SMP mod_unload modversions
parm:         name:The name to display in /var/log/kern.log (charp)
(base) vicky@vicky:~/Escritorio/PDIH/Seminarios/S4/modulos-cargables-nucleo$
```

Para terminar con este seminario, eliminamos el módulo del kernel con el comando *rmmod*. Además, podemos comprobar que el módulo se había cargado correctamente en el log del kernel, ya que mostraría los dos mensajes esperados de nuestro código.

```
(base) vicky@vicky:~/Escritorio/PDIH/Seminarios/S4/modulos-cargables-nucleo$ sudo rmmod hello
(base) vicky@vicky:~/Escritorio/PDIH/Seminarios/S4/modulos-cargables-nucleo$ sudo su
root@vicky:/home/vicky/Escritorio/PDIH/Seminarios/S4/modulos-cargables-nucleo# cd /
root@vicky:/# cd /var/log
root@vicky:/var/log# tail -f kern.log
Apr  1 20:30:18 vicky kernel: [34420.577952] sd 2:0:0:0: [sdb] Write Protect is off
Apr  1 20:30:18 vicky kernel: [34420.577954] sd 2:0:0:0: [sdb] Mode Sense: 47 00 10 08
Apr  1 20:30:18 vicky kernel: [34420.578150] sd 2:0:0:0: [sdb] No Caching mode page found
Apr  1 20:30:18 vicky kernel: [34420.578154] sd 2:0:0:0: [sdb] Assuming drive cache: write through
Apr  1 20:30:18 vicky kernel: [34420.799842]   sdb: sdb1
Apr  1 20:30:18 vicky kernel: [34420.817372] sd 2:0:0:0: [sdb] Attached SCSI disk
Apr  1 20:30:18 vicky kernel: [34420.817530] ses 2:0:0:1: Attached Enclosure device
Apr  1 20:46:48 vicky kernel: [35410.258241] usb 2-1: USB disconnect, device number 2
Apr  1 20:50:21 vicky kernel: [35623.451410] Hola Victoria desde el kernel
Apr  1 20:55:20 vicky kernel: [35922.123440] Adios Victoria desde el kernel
```