

ALGORÍTMICA

SEMINARIO 4:

LKM

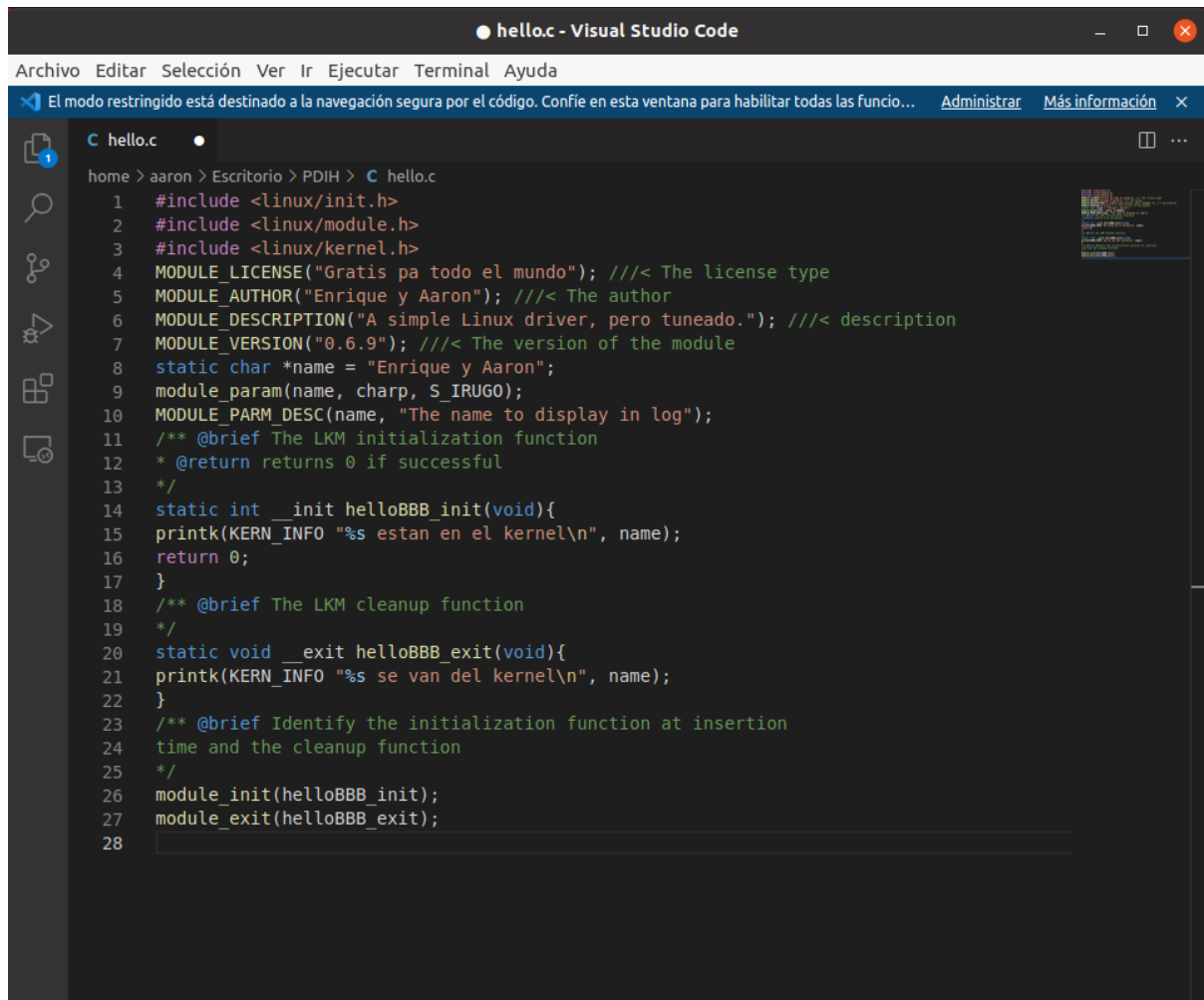


**UNIVERSIDAD
DE GRANADA**

Equipo Original

ENRIQUE GONZÁLEZ LÓPEZ
AARON RIVET RAMÍREZ

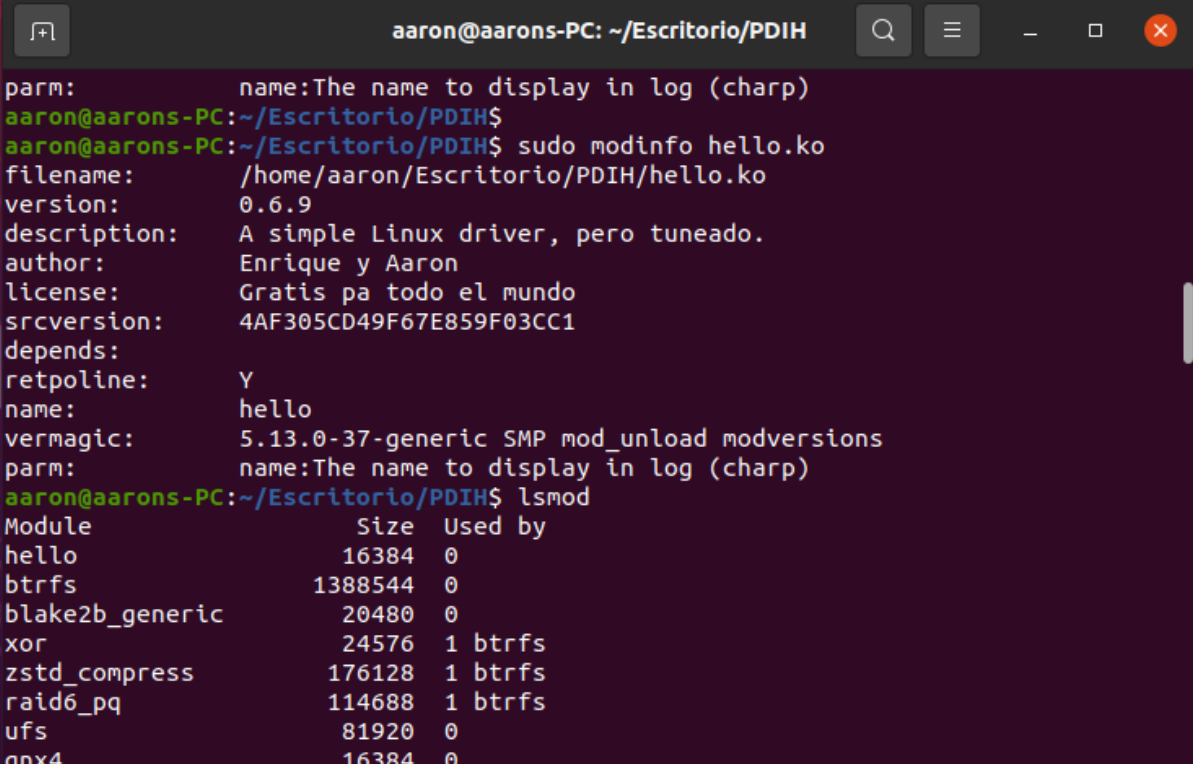
En este seminario, hemos modificado el código que el propio seminario nos daba, y lo hemos dejado tal que así:



```
hello.c - Visual Studio Code
Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
El modo restringido está destinado a la navegación segura por el código. Confíe en esta ventana para habilitar todas las funcio... Administrar Más información X
C hello.c
home > aaron > Escritorio > PDIH > C hello.c
1 #include <linux/init.h>
2 #include <linux/module.h>
3 #include <linux/kernel.h>
4 MODULE_LICENSE("Gratis pa todo el mundo"); ///< The license type
5 MODULE_AUTHOR("Enrique y Aaron"); ///< The author
6 MODULE_DESCRIPTION("A simple Linux driver, pero tuneado."); ///< description
7 MODULE_VERSION("0.6.9"); ///< The version of the module
8 static char *name = "Enrique y Aaron";
9 module_param(name, charp, S_IRUGO);
10 MODULE_PARM_DESC(name, "The name to display in log");
11 /** @brief The LKM initialization function
12  * @return returns 0 if successful
13  */
14 static int __init helloBBB_init(void){
15     printk(KERN_INFO "%s estan en el kernel\n", name);
16     return 0;
17 }
18 /** @brief The LKM cleanup function
19  */
20 static void __exit helloBBB_exit(void){
21     printk(KERN_INFO "%s se van del kernel\n", name);
22 }
23 /** @brief Identify the initialization function at insertion
24  time and the cleanup function
25  */
26 module_init(helloBBB_init);
27 module_exit(helloBBB_exit);
28
```

Como podemos ver, hemos modificado la línea donde se muestran las licencias, autores, versión, descripción y el nombre que mostrará el kernel, además de los 2 mensajes de llegada y de salida al kernel.

Podemos ver después como en la lista de módulos aparece tanto como el módulo hello, como la información de dicho módulo después de insertarlo con el comando insmod.



```
aaron@aarons-PC: ~/Escritorio/PDIH
parm:      name:The name to display in log (charp)
aaron@aarons-PC:~/Escritorio/PDIH$
aaron@aarons-PC:~/Escritorio/PDIH$ sudo modinfo hello.ko
filename:   /home/aaron/Escritorio/PDIH/hello.ko
version:    0.6.9
description: A simple Linux driver, pero tuneado.
author:     Enrique y Aaron
license:    Gratis pa todo el mundo
srcversion: 4AF305CD49F67E859F03CC1
depends:
retpoline:  Y
name:       hello
vermagic:   5.13.0-37-generic SMP mod_unload modversions
parm:      name:The name to display in log (charp)
aaron@aarons-PC:~/Escritorio/PDIH$ lsmod
Module              Size  Used by
hello                16384  0
btrfs               1388544  0
blake2b_generic      20480  0
xor                  24576  1 btrfs
zstd_compress        176128  1 btrfs
raid6_pq             114688  1 btrfs
ufs                   81920  0
qnx4                  16384  0
```

También adjuntamos una imagen luego de haber borrado el módulo, en la cual podemos ver como no era un módulo peligroso para el sistema ni necesario puesto que el sistema funciona con total normalidad.

```
aaron@aarons-PC: ~/Escritorio/PDIH
aaron@aarons-PC:~/Escritorio/PDIH$ sudo rmmod hello.ko
aaron@aarons-PC:~/Escritorio/PDIH$ lsmod
Module                  Size  Used by
btrfs                   1388544  0
blake2b_generic         20480   0
xor                     24576   1 btrfs
zstd_compress           176128   1 btrfs
raid6_pq                114688   1 btrfs
ufs                     81920   0
qnx4                    16384   0
hfsplus                 110592   0
hfs                     61440   0
minix                   36864   0
ntfs                    106496   0
msdos                   20480   0
jfs                     188416   0
xfs                     1515520  0
libcrc32c               16384   2 btrfs,xfs
rfcomm                  81920   4
cmac                    16384   3
algif_hash              16384   1
algif_skcipher          16384   1
af_alg                  28672   6 algif_hash,algif_skcipher
bnep                    24576   2
```

Y para finalizar mostramos el funcionamiento de dicho módulo chequeando los registros del kernel:

```
root@aarons-PC: /var/log
root@aarons-PC: /var/log# tail -f kern.log
May  5 17:15:23 aarons-PC kernel: [ 428.641879] audit: type=1400 audit(1651763723.258:61): apparmor="STATUS" operation="profile_replace" info="same as current profile, skipping" profile="unconfined" name="snap.snap-store.hook.configure" pid=20980 comm="apparmor_parser"
May  5 17:15:23 aarons-PC kernel: [ 428.640659] audit: type=1400 audit(1651763723.262:62): apparmor="STATUS" operation="profile_replace" info="same as current profile, skipping" profile="unconfined" name="snap-update-ns.snap-store" pid=20979 comm="apparmor_parser"
May  5 17:15:23 aarons-PC kernel: [ 428.652484] audit: type=1400 audit(1651763723.266:63): apparmor="STATUS" operation="profile_replace" info="same as current profile, skipping" profile="unconfined" name="snap.snap-store.snap-store" pid=20981 comm="apparmor_parser"
May  5 17:15:23 aarons-PC kernel: [ 428.654993] audit: type=1400 audit(1651763723.270:64): apparmor="STATUS" operation="profile_replace" info="same as current profile, skipping" profile="unconfined" name="snap.snap-store.ubuntu-software" pid=20982 comm="apparmor_parser"
May  5 17:25:12 aarons-PC kernel: [ 1018.221942] hello: loading out-of-tree module taints kernel.
May  5 17:25:12 aarons-PC kernel: [ 1018.221951] hello: module license 'Gratis pa todo el mundo' taints kernel.
May  5 17:25:12 aarons-PC kernel: [ 1018.221954] Disabling lock debugging due to kernel taint
May  5 17:25:12 aarons-PC kernel: [ 1018.222003] hello: module verification failed: signature and/or required key missing - tainting kernel
May  5 17:25:12 aarons-PC kernel: [ 1018.222480] Enrique y Aaron estan en el kernel
May  5 17:28:07 aarons-PC kernel: [ 1193.028982] Enrique y Aaron se van del kernel
```

Aunque de manera pobre, podemos observar como en las últimas líneas aparecen las modificaciones que hemos realizado en el código para mostrar el mensaje modificado.