



UNIVERSIDAD AUTÓNOMA DE
CHIHUAHUA



**UNIVERSIDAD AUTONOMA DE CHIHUAHUA
FACULTAD DE INGENIERIA**

SISTEMAS OPERATIVOS

Parcial 3 Kernel Module

Grupo: 5HW1

Alumno:

Jorge Eduardo Aguilar Silva, 339002

Docente:

Chavero Jurado Ivan Miguel

Fecha: 30/05/22

Contenido

Funcionamiento.....	3
Como se usa el módulo	4
Capturas de pantalla del funcionamiento	6

Funcionamiento

El código que se realizó tiene la función de crear un módulo de Kernel ingresando la licencia, el nombre del autor, la descripción del módulo y la versión del módulo. Se utilizarán comandos del device para crearlo, así como sus librerías para poder escribir en el device, se utilizará unos comandos que son los “file operations”, que nos permitirán escribir, leer, abrir y lanzar el módulo de Kernel.

Se utilizará una variable int que sirve para registrar el número mayor del device, en dado caso de que no exista un número significará que el módulo no se cargó correctamente. En caso contrario el módulo de Kernel mostrará por el Journal los mensajes para corroborar su funcionamiento, así como el número del registro que se utilizará para poder imprimir en el espacio del usuario el mensaje que ingresamos en el módulo, que en este caso es la letra de una canción.

Dependiendo de la acción realizada, en el Journal se mostrará un mensaje indicando si el módulo es solo de escritura, si el módulo fue cerrado y si se abrió con éxito.

Al finalizar se desacoplará el módulo del Kernel y finalizará el programa.

Como se usa el módulo

1. Para usar el módulo lo primero que tenemos que hacer es abrir dos terminales, una para ver como el módulo se carga y usa el journal (bitácora) y otro para observar cómo corre el módulo desde la perspectiva del usuario

Para que se muestre el journal en una de las dos terminales se ingresara el siguiente comando:

journalctl -f esto funciona para que la terminal observe lo que está sucediendo en Journal.

2. En la otra terminal que no usamos para ejecutar el Journal (Batch), deberemos de posicionarnos en la carpeta en la que se encuentre nuestro modulo, en mi caso

dir Universidad.

3. Usaremos el comando **Make** Para ensamblar el módulo.
4. Con el comando **lsmod | grep Batman** Observaremos que el módulo esta correctamente cargado
5. Utilizaremos el comando **sudo insmod Batman.ko** para que nos muestre en la bitácora el texto que ingresamos en el código.

```
printk(KERN_INFO "Inicializando módulo batman!!!\n");
printk(KERN_INFO "I was assigned major number %d. To talk to\n", major);
printk(KERN_INFO "the driver, create a dev file with\n");
printk(KERN_INFO "'mknod /dev/%s c %d 0'.\n", DEVICE_NAME, major);
printk(KERN_INFO "Try various minor numbers. Try to cat and echo to\n");
printk(KERN_INFO "the device file.\n");
printk(KERN_INFO "Remove the device file and module when done.\n");
```

6. Ingresaremos el comando mostrado en la 4ta fila del mensaje y le agregaremos un sudo

sudo mknod /dev/Batman_module c 237 0

7. Para finalizar ingresaremos el comando **cat /dev/Batman_dev** Para mostrar el mensaje de usuario de Kernel, que en este caso es la canción que ingresamos en el código.
8. Ctrl + C para terminar el programa.

Capturas de pantalla del funcionamiento

[illegible]

```
[jorge@fedora Universidad]$ journalctl -f
-- Journal begins at Wed 2022-03-30 13:25:18 MST. --
May 30 17:16:19 fedora packagekitd[1072]: Failed to get cache filename for libjose
May 30 17:16:19 fedora packagekitd[1072]: Failed to get cache filename for liblouis-utils
May 30 17:16:19 fedora packagekitd[1072]: Failed to get cache filename for liblouisutdml
May 30 17:16:19 fedora packagekitd[1072]: Failed to get cache filename for liblouisutdml-utils
May 30 17:16:19 fedora packagekitd[1072]: Failed to get cache filename for lldpad
May 30 17:16:19 fedora packagekitd[1072]: Failed to get cache filename for python3-chardet
May 30 17:16:19 fedora packagekitd[1072]: Failed to get cache filename for tslib
May 30 17:16:20 fedora PackageKit[1072]: get-details transaction /271 dcecdce from uid 1000 finished with success after 1562ms
May 30 17:17:23 fedora systemd[1431]: Starting Mark boot as successful...
May 30 17:17:23 fedora systemd[1431]: Finished Mark boot as successful.
May 30 17:20:23 fedora systemd[1431]: Created slice User Background Tasks Slice.
May 30 17:20:23 fedora systemd[1431]: Starting Cleanup of User's Temporary Files and Directories...
```

```
May 30 17:28:06 fedora sudo[3467]: pam_unix(sudo:session): session opened for user root@
May 30 17:28:06 fedora audit[3467]: USER_START pid=3467 uid=1000 auid=1000 ses=3 subj=unc
t,pam_limits,pam_systemd,pam_unix acct="root" exe="/usr/bin/sudo" hostname=? addr=? termi
May 30 17:28:06 fedora kernel: Inicializando módulo batman!!!
May 30 17:28:06 fedora kernel: I was assigned major number 237. To talk to
May 30 17:28:06 fedora kernel: the driver, create a dev file with
May 30 17:28:06 fedora kernel: 'mknod /dev/Batman_module c 237 0'.
May 30 17:28:06 fedora kernel: Try various minor numbers. Try to cat and echo to
May 30 17:28:06 fedora kernel: the device file.
May 30 17:28:06 fedora kernel: Remove the device file and module when done.
May 30 17:28:06 fedora audit[3467]: USER_END pid=3467 uid=1000 auid=1000 ses=3 subj=uncor
n,pam_limits,pam_systemd,pam_unix acct="root" exe="/usr/bin/sudo" hostname=? addr=? termi
```