|  |
| --- |
| Licenciatura en Sistema |
| Informe – TP Drivers |
| Organización del Computador II |

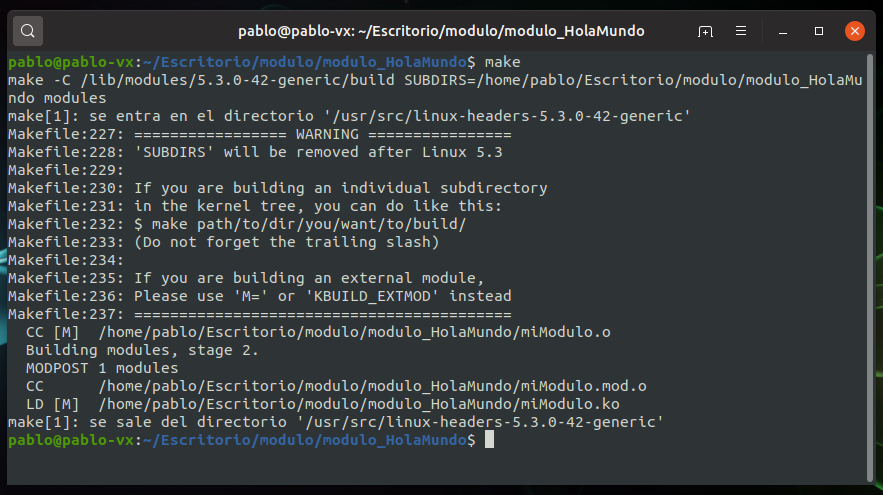
|  |
| --- |
| Pablo Abregu  23-4-2020 |

Modulo Hola Mundo

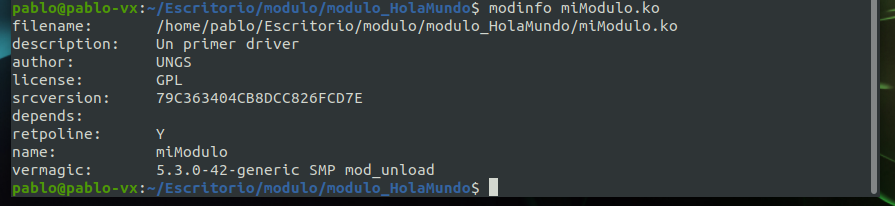
**Archivos**:



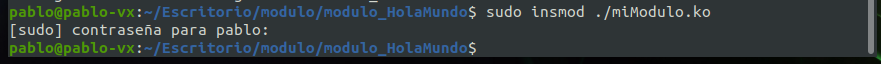
1. Compilando *miModulo.c* con *make*



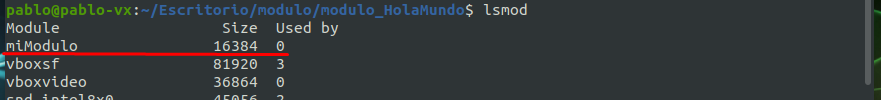
Utilizo *modinfo* para ver la información del módulo, ahora *miModulo.ko*.



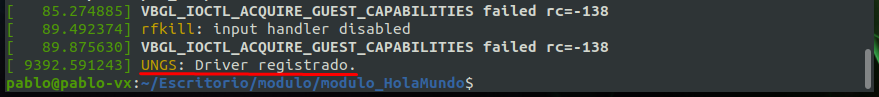
Cargando el módulo al núcleo con *insmod.*



1. Para verificar que funcione podemos listar los módulos cargados, con *lsmod*.



Además, con el comando *dmesg,* se puede revisar el log del kernel.



Modulo Char Device

Elaborando un kernel module para un char device.

1. Definir funciones *init\_module* y *cleanup\_module*
   1. ¿Qué necesito que hagan en un char device?
2. Definir funciones *device\_open* y *device\_release*
3. Hacer que nuestro char device cuando le escribimos imprima en el kernel

* Tip 1: Para escribir a un device en Bash nano /dev/device
* Tip 2: Ojo con los permisos de escritura!
* Tip 3: Al de-registrar y volver a registrar el driver repetir el proceso de crear el archivo, sino esto trae problemas. Mismo no olvidar hacer make clean

1. Hacer que nuestro char device cuando lo lea me devuelva lo último que fue escrito.
   1. Tip 4: cat /dev/device me tiene que devolver lo que puse con echo
   2. Al cargar el módulo, se llama a la función *init\_module*, donde servirá para obtener el Major Number con la función *register\_chrdev(…)* y registrar el device. Lo contrario pasa con *cleanup\_module*, donde sirve para descargar el módulo, y con la función *unregister\_chrdev(…)* permitir anular el registro del device.
2. *device\_open*: llamada cuando se intenta abrir el archivo device.

*device\_release*: llamada cuando se cierra el archivo del device.

static int device\_open**(**struct inode **\***inode**,** struct file **\***file**)**

**{**

**if(**Device\_Open**)**

**return** **-**EBUSY**;**

Device\_Open**++;**

try\_module\_get**(**THIS\_MODULE**);**

**return** SUCCESS**;**

**}**

static int device\_release**(**struct inode **\***inode**,** struct file **\***file**)**

**{**

Device\_Open**--;**

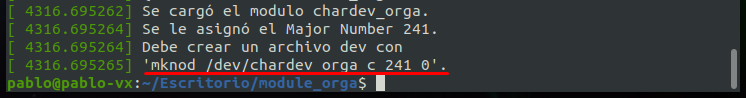
module\_put**(**THIS\_MODULE**);**

**return** SUCCESS**;**

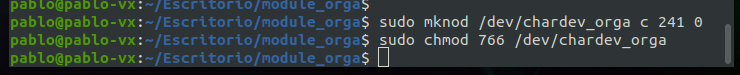
**}**

1. Para escribir al device se debe crear el archivo correspondiente en */dev/.* Teniendo en cuenta los permisos de usuario.

En este caso, *chardev*\_orga.



Creando el archivo en */dev/.*

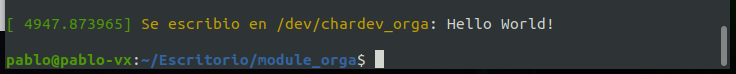


Si le escribimos algo, este lo mostrara en el log del kernel.

Ejemplo: *echo Hello World! > /dev/chardev\_orga*



Miramos el log del Kernel con *dmesg*.



1. Para ver lo último que fue escrito en el device: *cat /dev/.*

Ejemplo: *cat /dev/chardev\_orga*



**Archivos**:

**Repositorio en Git:** <https://github.com/pabloabregu/Module-Char-Device.git>