Modulo Hola Mundo

Hacer andar el modulo hola mundo https://bitbucket.org/sor2/tp0.

1. Usando el código provisto y los comandos de buildeo y carga/descarga de drivers cargar nuestro primer modulo

Codigo 1: miModulo.c

```c
#include <linux/module.h>
#include <linux/kernel.h>

int init_module(void)
{ /* Constructor */
    printk(KERN_INFO "UNGS: Driver registrado\n");
    return 0;
}

void cleanup_module(void)
{/* Destructor */
    printk(KERN_INFO "UNGS: Driver desregistrado\n");
}

MODULE_LICENSE("GPL");
MODULE_AUTHOR("UNGS");
MODULE_DESCRIPTION("Un primer driver");
```

Codigo 2: Makefile

```
obj-m := chardev_buffer.o

all:
        make -C /lib/modules/$(shell uname -r)/build SUBDIRS=$(shell
           pwd) modules
clean:
        make -C /lib/modules/$(shell uname -r)/build SUBDIRS=$(shell
           pwd) clean
```

2. Verificar que funcione.
    - Como hacemos eso?

Modulo Char Device

Elaborar un kernel module para un char device

Material de lectura: https://www.tldp.org/LDP/lkmpg/2.6/lkmpg.pdf.

1. Definir funciones *init_module* y *cleanup_module*
   - ¿Que 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
4. Hacer que nuestro char device cuando lo lea me devuelva lo ultimo que fue escrito
   - Tip 4: cat /dev/device me tiene que devolver lo que puse con echo
5. Hacer que ahora devuelva el mensaje al reves caracter por caracter, por ej si el mensaje es hola: aloh