Abdullah Barghouti
ECE 373 HW2
4/20/2019

SOURCE CODE TO KERNEL MODULE

```
/*
Abdullah Barghouti
ECE 373 HW 2
Portland State University 2019
*/

#include <linux/module.h>
#include <linux/types.h>
#include <linux/kdev_t.h>
#include <linux/fs.h>
#include <linux/cdev.h>
#include <linux/usb.h>
#include <linux/slab.h>
#include <linux/uaccess.h>

#define DEVCNT 5
#define DEVNAME "char_driver"

static struct mydev_dev{
	struct cdev cdev;
	dev_t mydev_node;
	int sys_int;
	int syscall_val;
} mydev;


static int sys_intial_value = 40;
module_param(sys_intial_value, int, S_IRUSR | S_IWUSR);

// function for opening module
static int char_driver_open(struct inode *inode, struct file *file)
{
	printk(KERN_INFO "open module succeful \n");
	mydev.syscall_val = sys_intial_value;
	//mydev.sys_int = 40;
	return 0;
}

// function for reading
static ssize_t char_driver_read(struct file *file, char __user *buf, size_t len, loff_t *offset){
	int ret;
	printk(KERN_INFO "Currently Reading...");

	if(*offset >= sizeof(int))
		return 0;
	if(!buf){
		ret = -EINVAL;
		goto out;
```

```
	}
	if(copy_to_user(buf, &mydev.syscall_val, sizeof(int))){
		ret = -EFAULT;
		goto out;
	}
	ret = sizeof(int);
	*offset += len;

	/* Good to go, so printk the thingy */
	printk(KERN_INFO "User got from us %d\n", mydev.sys_int);

out:
	return ret;
}

//function to write/ update based on userspace
static ssize_t char_driver_write(struct file *file, const char __user *buf, size_t len, loff_t *offset){

	char *kern_buf;
	int ret;
	printk(KERN_INFO "Currently Writing...");

	if(!buf){
		ret = -EINVAL;
		goto out;
	}

	kern_buf = kmalloc(len, GFP_KERNEL);

	if(!kern_buf){
		ret = -ENOMEM;
		goto out;
	}

	if(copy_from_user(kern_buf, buf, len)){
		ret = -EFAULT;
		goto mem_out;
	}

	ret = len;
	printk(KERN_INFO "Userspace wrote \"%s\" to us \n", kern_buf);

mem_out:
	kfree(kern_buf);
out:
	return ret;
}
// open, read, write struct
static struct file_operations mydev_fops = {
	.owner = THIS_MODULE,
	.open = char_driver_open,
	.read = char_driver_read,
	.write = char_driver_write,
};
```

```
// init function
static int __init char_driver_init(void){
	printk(KERN_INFO "Module loading... \n");


	if(alloc_chrdev_region(&mydev.mydev_node, 0, DEVCNT, DEVNAME)){
		printk(KERN_ERR "Allocating chrdev failed \n");
		return -1;
	}

	printk(KERN_INFO "Allocated %d devices at major: %d\n", DEVCNT,
MAJOR(mydev.mydev_node));

	/*initialize the character device */
	cdev_init(&mydev.cdev, &mydev_fops);
	mydev.cdev.owner = THIS_MODULE;

	if(cdev_add(&mydev.cdev, mydev.mydev_node, DEVCNT)){
		printk(KERN_ERR "cdev_add failed \n");
		/* clean up allocation of chrdev */
		unregister_chrdev_region(mydev.mydev_node, DEVCNT);
		return -1;
	}
	 return 0;
}

// exit function
static void __exit char_driver_exit(void){
	cdev_del(&mydev.cdev);
	unregister_chrdev_region(mydev.mydev_node, DEVCNT);
	printk(KERN_INFO "Unloaded module \n");
}

MODULE_AUTHOR("Abdullah Barghouti");
MODULE_LICENSE("GPL");
MODULE_VERSION("0.2");
module_init(char_driver_init);
module_exit(char_driver_exit);
```

MAKEFILE

```
KERNEL_DIR = /LIB/MODULES/$(SHELL UNAME -R)/BUILD
PWD := $(SHELL PWD)
OBJ-M += CHAR_DRIVER.O
DEFAULT:
		$(MAKE) -C $(KERNEL_DIR) SUBDIRS=$(PWD) MODULES
CLEAN:
		$(MAKE) -C $(KERNEL_DIR) SUBDIRS=$(PWD) CLEAN
```

USERSPACE PROGRAM

```
/*
Abdullah Barghouti
ECE 373 HW 2
Portland State University 2019
*/

#include <sys/stat.h>
#include <string.h>
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <errno.h>
#include <fcntl.h>

int main()
{
	char newCharBuffer[40];
	char currentCharBuffer[40];
	int fd, value;

	// open file
	fd = open("/dev/char_driver", O_RDWR);
	if(fd < 0)
		printf("ERROR! couldnt open file...\n");

	// read file
	int ret = read(fd, currentCharBuffer, sizeof(int));
	memcpy(&value, currentCharBuffer, sizeof(int) * sizeof(char));
	printf("\nRead from device: %d\n", value);

	//get user's value
	printf("Enter new value to send to the device ");
	fgets(newCharBuffer, 40, stdin);
	printf("the new value that was sent to the driver is: %s\n",newCharBuffer);

	//close file
	close(fd);
	return 0;
}
```

TYPESCRIPT OF LOADING THE DRIVER AND RUNNING USERSPACE PROGRAM

```
Script started on Sun 21 Apr 2019 12:45:28 AM PDT
barghouti@barghouti-VirtualBox: ~/Desktop/hw/hw2/Kernel$ make
make -C /lib/modules/4.15.0-47-generic/build SUBDIRS=/home/barghouti/Desktop/hw/hw2/
Kernel modules
make[1]: Entering directory '/usr/src/linux-headers-4.15.0-47-generic'
Makefile:976: "Cannot use CONFIG_STACK_VALIDATION=y, please install libelf-dev, libelf-
devel or elfutils-libelf-devel"
```

```
  CC [M]  /home/barghouti/Desktop/hw/hw2/Kernel/char_driver.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/barghouti/Desktop/hw/hw2/Kernel/char_driver.mod.o
  LD [M]  /home/barghouti/Desktop/hw/hw2/Kernel/char_driver.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.15.0-47-generic'
barghouti@barghouti-VirtualBox: ~/Desktop/hw/hw2/Kernel$ sudo insmod char_driver.ko
[sudo] password for barghouti:
barghouti@barghouti-VirtualBox: ~/Desktop/hw/hw2/Kernel$ cd ../userspace/
barghouti@barghouti-VirtualBox: ~/Desktop/hw/hw2/userspace$ ./a.out

Read from device: 40
Enter new value to send to the device 40
the new value that was sent to the driver is: 40

barghouti@barghouti-VirtualBox: ~/Desktop/hw/hw2/userspace$ ./a.out

Read from device: 40
Enter new value to send to the device 120
the new value that was sent to the driver is: 120

barghouti@barghouti-VirtualBox: ~/Desktop/hw/hw2/userspace$ sudo rmmod char_driver

]0;barghouti@barghouti-VirtualBox: ~/Desktop/hw/hw2/Kernel$ sudo insmod char_driver.ko
sys_intial_value=11
]0;barghouti@barghouti-VirtualBox: ~/Desktop/hw/hw2/Kernel$ cd ../userspace/
]0;barghouti@barghouti-VirtualBox: ~/Desktop/hw/hw2/userspace$ ./a.out

Read from device: 11
Enter new value to send to the device 11
the new value that was sent to the driver is: 11

]0;barghouti@barghouti-VirtualBox: ~/Desktop/hw/hw2/userspace$ exit
exit

Script done on Sun 21 Apr 2019 12:49:04 AM PDT
```

TYPESCRIPT OF /PROC/DEVICES

```
Script started on Sun 21 Apr 2019 12:51:34 AM PDT
barghouti@barghouti-VirtualBox: ~/Desktop/hw/hw2/Kernel$ cat /proc/devices | grep char
243 char_driver
barghouti@barghouti-VirtualBox: ~/Desktop/hw/hw2/Kernel$ exit
exit

Script done on Sun 21 Apr 2019 12:52:12 AM PDT
```