Exercise - 6

22.03.2021

CED17I017

FIROZ MOHAMMAD

Write a simple Char Device Driver. Test it by using passing on data.

char_DD.c

```
#include <linux/init.h>
#include <linux/module.h>
#include <linux/cdev.h>
#include <linux/device.h>
#include <linux/kernel.h>
#include <linux/uaccess.h>
#include <linux/fs.h>

#define MAX_DEV 2

static int mychardev_open(struct inode *inode, struct file *file);
static int mychardev_release(struct inode *inode, struct file *file);
static long mychardev_ioctl(struct file *file, unsigned int cmd, unsigned
long arg);
static ssize_t mychardev_read(struct file *file, char __user *buf, size_t
count, loff_t *offset);
static ssize_t mychardev_write(struct file *file, const char __user *buf,
size_t count, loff_t *offset);

static const struct file_operations mychardev_fops = {
    .owner      = THIS_MODULE,
    .open       = mychardev_open,
    .release    = mychardev_release,
    .unlocked_ioctl = mychardev_ioctl,
    .read       = mychardev_read,
    .write       = mychardev_write
};

struct mychar_device_data {
    struct cdev cdev;
};

static int dev_major = 0;
static struct class *mychardev_class = NULL;
```

```
static struct mychar_device_data mychardev_data[MAX_DEV];

static int mychardev_uevent(struct device *dev, struct kobj_uevent_env
*env)
{
    add_uevent_var(env, "DEVMODE=%#o", 0666);
    return 0;
}

static int __init mychardev_init(void)
{
    int err, i;
    dev_t dev;

    err = alloc_chrdev_region(&dev, 0, MAX_DEV, "mychardev");

    dev_major = MAJOR(dev);

    mychardev_class = class_create(THIS_MODULE, "mychardev");
    mychardev_class->dev_uevent = mychardev_uevent;

    for (i = 0; i < MAX_DEV; i++) {
        cdev_init(&mychardev_data[i].cdev, &mychardev_fops);
        mychardev_data[i].cdev.owner = THIS_MODULE;

        cdev_add(&mychardev_data[i].cdev, MKDEV(dev_major, i), 1);

        device_create(mychardev_class, NULL, MKDEV(dev_major, i), NULL,
"mychardev-%d", i);
    }

    return 0;
}

static void __exit mychardev_exit(void)
{
    int i;

    for (i = 0; i < MAX_DEV; i++) {
        device_destroy(mychardev_class, MKDEV(dev_major, i));
    }
```

```
    class_unregister(mychardev_class);
    class_destroy(mychardev_class);

    unregister_chrdev_region(MKDEV(dev_major, 0), MINORMASK);
}

static int mychardev_open(struct inode *inode, struct file *file)
{
    printk("MYCHARDEV: Device open\n");
    return 0;
}

static int mychardev_release(struct inode *inode, struct file *file)
{
    printk("MYCHARDEV: Device close\n");
    return 0;
}

static long mychardev_ioctl(struct file *file, unsigned int cmd, unsigned
long arg)
{
    printk("MYCHARDEV: Device ioctl\n");
    return 0;
}

static ssize_t mychardev_read(struct file *file, char __user *buf, size_t
count, loff_t *offset)
{
    uint8_t *data = "Hello from the kernel world!\n";
    size_t datalen = strlen(data);

    printk("Reading device: %d\n",
MINOR(file->f_path.dentry->d_inode->i_rdev));

    if (count > datalen) {
        count = datalen;
    }

    if (copy_to_user(buf, data, count)) {
        return -EFAULT;
    }
```

```
    return count;
}

static ssize_t mychardev_write(struct file *file, const char __user *buf,
size_t count, loff_t *offset)
{
    size_t maxdatalen = 30, ncopied;
    uint8_t databuf[maxdatalen];

    printk("Writing device: %d\n",
MINOR(file->f_path.dentry->d_inode->i_rdev));

    if (count < maxdatalen) {
        maxdatalen = count;
    }

    ncopied = copy_from_user(databuf, buf, maxdatalen);

    if (ncopied == 0) {
        printk("Copied %zd bytes from the user\n", maxdatalen);
    } else {
        printk("Could't copy %zd bytes from the user\n", ncopied);
    }

    databuf[maxdatalen] = 0;

    printk("Data from the user: %s\n", databuf);

    return count;
}

MODULE_LICENSE("GPL");
MODULE_AUTHOR("Firoz Mohammad>");

module_init(mychardev_init);
module_exit(mychardev_exit);
```

Obviously we can not compile this module , for this we would need a Makefile.

Makefile

```
BINARY     := mychardev
KERNEL      := /lib/modules/$(shell uname -r)/build
ARCH        := x86
C_FLAGS     := -Wall
KMOD_DIR    := $(shell pwd)
TARGET_PATH := /lib/modules/$(shell uname -r)/kernel/drivers/char

OBJECTS := char_dd.o

ccflags-y += $(C_FLAGS)

obj-m += $(BINARY).o

$(BINARY)-y := $(OBJECTS)

$(BINARY).ko:
	make -C $(KERNEL) M=$(KMOD_DIR) modules

install:
	cp $(BINARY).ko $(TARGET_PATH)
	depmod -a

uninstall:
	rm $(TARGET_PATH)/$(BINARY).ko
	depmod -a

clean:
	make -C $(KERNEL) M=$(KMOD_DIR) clean
```

Now, after running “make” , we will get the “HelloWorld.ko” and others related file.

```
(base) firoz@320-15ikb:~/Desktop/faltu/firoz/sem_8/Device_Driver/Practice/Lab2/C
har_DD$ make
make -C /lib/modules/5.3.0-7648-generic/build M=/home/firoz/Desktop/faltu/firoz/
sem_8/Device_Driver/Practice/Lab2/Char_DD modules
make[1]: Entering directory '/usr/src/linux-headers-5.3.0-7648-generic'
  CC [M]  /home/firoz/Desktop/faltu/firoz/sem_8/Device_Driver/Practice/Lab2/Char
_DD/char_dd.o
/home/firoz/Desktop/faltu/firoz/sem_8/Device_Driver/Practice/Lab2/Char_DD/char_d
d.c: In function ‘mychardev_write’:
/home/firoz/Desktop/faltu/firoz/sem_8/Device_Driver/Practice/Lab2/Char_DD/char_d
d.c:117:5: warning: ISO C90 forbids variable length array ‘databuf’ [-Wvla]
     uint8_t databuf[maxdatalen];
     ^~~~~~~
  LD [M]  /home/firoz/Desktop/faltu/firoz/sem_8/Device_Driver/Practice/Lab2/Char
_DD/mychardev.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/firoz/Desktop/faltu/firoz/sem_8/Device_Driver/Practice/Lab2/Char
_DD/mychardev.mod.o
  LD [M]  /home/firoz/Desktop/faltu/firoz/sem_8/Device_Driver/Practice/Lab2/Char
_DD/mychardev.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.3.0-7648-generic'
(base) firoz@320-15ikb:~/Desktop/faltu/firoz/sem_8/Device_Driver/Practice/Lab2/C
har_DD$
```

To check creation of dev file :

```
(base) firoz@320-15ikb:~/Desktop/faltu/firoz/sem_8/Device_Driver/Practice/Lab2/C
har_DD$ tree /sys/devices/virtual/mychardev/
/sys/devices/virtual/mychardev/
├── mychardev-0
│   ├── dev
│   ├── power
│   │   ├── async
│   │   ├── autosuspend_delay_ms
│   │   ├── control
│   │   ├── runtime_active_kids
│   │   ├── runtime_active_time
│   │   ├── runtime_enabled
│   │   ├── runtime_status
│   │   ├── runtime_suspended_time
│   │   └── runtime_usage
│   ├── subsystem -> ../../../../class/mychardev
│   └── uevent
└── mychardev-1
    ├── dev
    ├── power
    │   ├── async
    │   ├── autosuspend_delay_ms
    │   ├── control
    │   ├── runtime_active_kids
    │   ├── runtime_active_time
    │   ├── runtime_enabled
    │   ├── runtime_status
    │   ├── runtime_suspended_time
    │   └── runtime_usage
    ├── subsystem -> ../../../../class/mychardev
    └── uevent

6 directories, 22 files
(base) firoz@320-15ikb:~/Desktop/faltu/firoz/sem_8/Device_Driver/Practice/Lab2/C
har_DD$
```

As we can see in code, two char devices have been created (mychardev0 and mychardev 1) with major value 238 and minor value 0 and 1 respectively.

To check file permission :

```
(base) firoz@320-15ikb:~/Desktop/faltu/firoz/sem_8/Device_Driver/Practice/Lab2/C
har_DD$ ls -l /dev/mychardev-*
crw-rw-rw- 1 root root 238, 0 Mar 22 23:27 /dev/mychardev-0
crw-rw-rw- 1 root root 238, 1 Mar 22 23:27 /dev/mychardev-1
(base) firoz@320-15ikb:~/Desktop/faltu/firoz/sem_8/Device_Driver/Practice/Lab2/C
har_DD$
```

We have rw-rw permission on each mychardev file.

Lets try to read something from mychardev-0 :

```
(base) firoz@320-15ikb:~/Desktop/faltu/firoz/sem_8/Device_Driver/Practice/Lab2/C
har_DD$ head -c29 /dev/mychardev-0
Hello from the kernel world!
(base) firoz@320-15ikb:~/Desktop/faltu/firoz/sem_8/Device_Driver/Practice/Lab2/C
har_DD$
```

Lets try to write something on mychardev-0 :

```
(base) firoz@320-15ikb:~/Desktop/faltu/firoz/sem_8/Device_Driver/Practice/Lab2/C
har_DD$ echo "Hello from the user Firoz" > /dev/mychardev-0
(base) firoz@320-15ikb:~/Desktop/faltu/firoz/sem_8/Device_Driver/Practice/Lab2/C
har_DD$ tail -n6 /var/log/kern.log
Mar 22 23:42:17 320-15ikb kernel: [ 2979.043463] MYCHARDEV: Device open
Mar 22 23:42:17 320-15ikb kernel: [ 2979.043537] Writing device: 0
Mar 22 23:42:17 320-15ikb kernel: [ 2979.043541] Copied 26 bytes from the user
Mar 22 23:42:17 320-15ikb kernel: [ 2979.043545] Data from the user: Hello from
the user Firoz
Mar 22 23:42:17 320-15ikb kernel: [ 2979.043545]
Mar 22 23:42:17 320-15ikb kernel: [ 2979.043554] MYCHARDEV: Device close
(base) firoz@320-15ikb:~/Desktop/faltu/firoz/sem_8/Device_Driver/Practice/Lab2/C
har_DD$
```

Lets try to write something on mychardev-1 :

```
(base) firoz@320-15ikb:~/Desktop/faltu/firoz/sem_8/Device_Driver/Practice/Lab2/C
har_DD$ echo "Hi from the user Haider" > /dev/mychardev-1
(base) firoz@320-15ikb:~/Desktop/faltu/firoz/sem_8/Device_Driver/Practice/Lab2/C
har_DD$ tail -n6 /var/log/kern.log
Mar 22 23:43:48 320-15ikb kernel: [ 3069.965336] MYCHARDEV: Device open
Mar 22 23:43:48 320-15ikb kernel: [ 3069.965415] Writing device: 1
Mar 22 23:43:48 320-15ikb kernel: [ 3069.965419] Copied 24 bytes from the user
Mar 22 23:43:48 320-15ikb kernel: [ 3069.965423] Data from the user: Hi from the
 user Haider
Mar 22 23:43:48 320-15ikb kernel: [ 3069.965423]
Mar 22 23:43:48 320-15ikb kernel: [ 3069.965436] MYCHARDEV: Device close
(base) firoz@320-15ikb:~/Desktop/faltu/firoz/sem_8/Device_Driver/Practice/Lab2/C
har_DD$
```