Devices

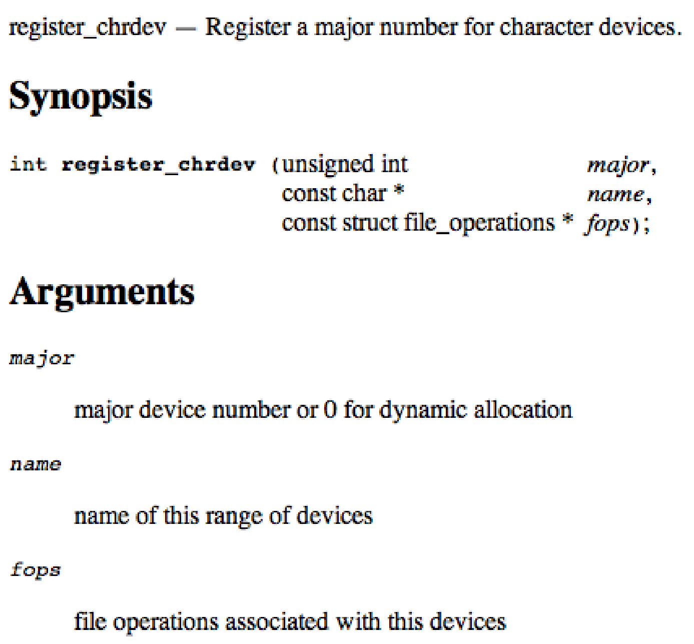
File(s) for this lab:

Objective: In this lab you will follow a walkthrough to build a character device "chardev" and communicate with it via a user mode application called "chartest".

/dev is the location of special or device files. It is a very interesting directory that highlights one important aspect of the Linux filesystem - everything is a file or a directory. Look through this directory and you should hopefully see ex. sda1, sda2 etc.... which represent the various partitions on the first master drive of the system. /dev/cdrom and /dev/fd0 represent your CD-ROM drive and your floppy drive. This may seem strange but it will make sense if you compare the characteristics of files to that of your hardware. Both can be read from and written to.

The majority of devices are either block or character devices; however other types of devices exist and can be created. In general, 'block devices' are devices that store or hold data, 'character devices' can be thought of as devices that transmit or transfer data. For example, diskette drives, hard drives and CD-ROM drives are all block devices while serial ports, mice and parallel printer ports are all character devices.
In this lab you will create a character device and communicate with it via its /dev entry.

1. To create a /dev entry we first need to use the function register_chrdev():



register_chrdev — Register a major number for character devices.

Synopsis

```
int register_chrdev (unsigned int                     major,
                     const char *                     name,
                     const struct file_operations *   fops);
```

Arguments

major

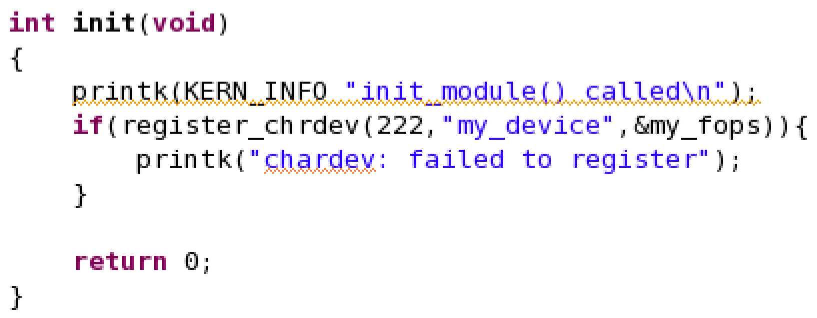
major device number or 0 for dynamic allocation

name

name of this range of devices

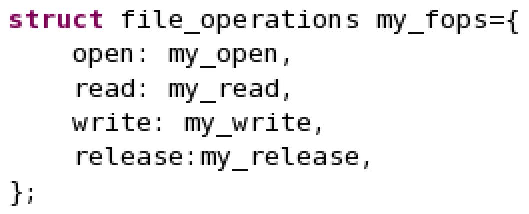
fops

file operations associated with this devices



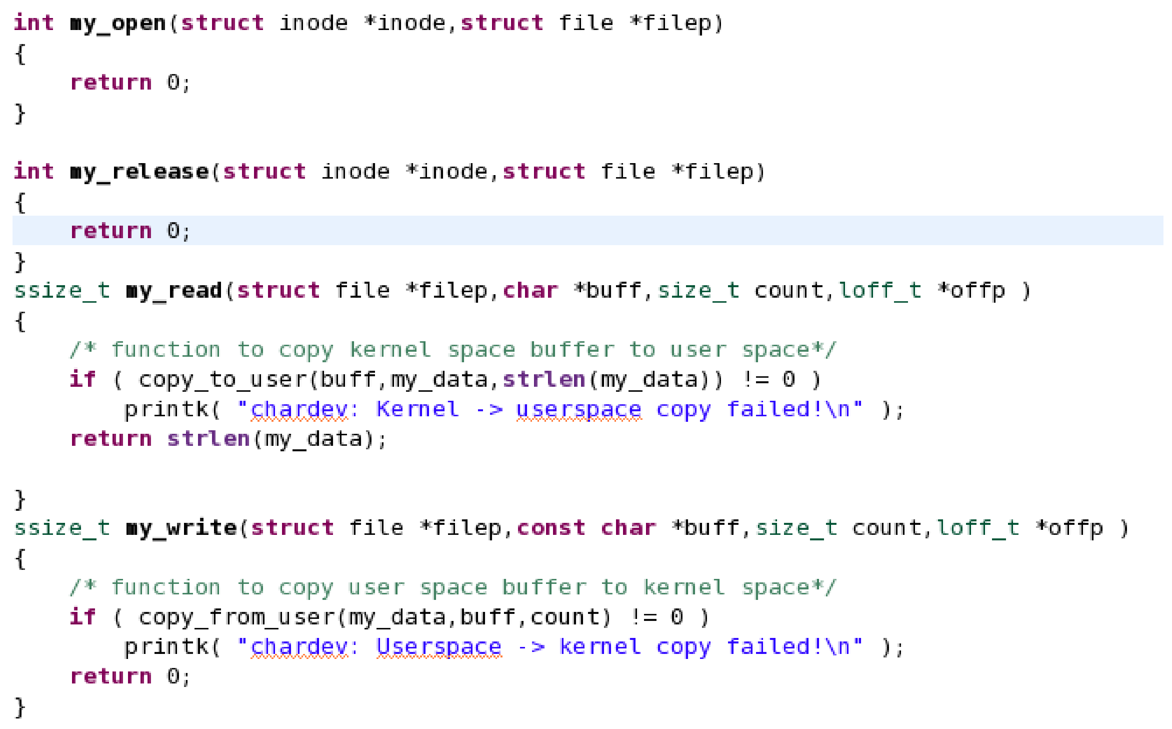
```
int init(void)
{
    printk(KERN_INFO "init_module() called\n");
    if(register_chrdev(222,"my_device",&my_fops)){
        printk("chardev: failed to register");
    }

    return 0;
}
```

2. We have already been exposed to the file_operations structure. We need to build our own functions for read and write to communicate with our user mode application.



```
struct file_operations my_fops={
    open: my_open,
    read: my_read,
    write: my_write,
    release:my_release,
};
```

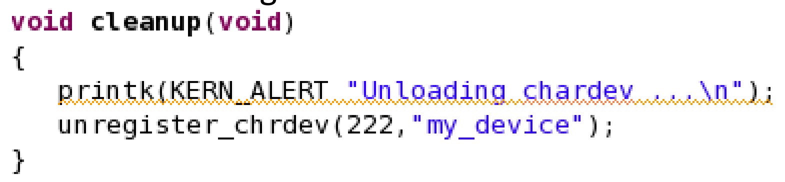
3. For our example open and release will be dummy functions.



```
int my_open(struct inode *inode,struct file *filep)
{
    return 0;
}

int my_release(struct inode *inode,struct file *filep)
{
    return 0;
}
ssize_t my_read(struct file *filep,char *buff,size_t count,loff_t *offp )
{
    /* function to copy kernel space buffer to user space*/
    if ( copy_to_user(buff,my_data,strlen(my_data)) != 0 )
        printk( "chardev: Kernel -> userspace copy failed!\n" );
    return strlen(my_data);

}
ssize_t my_write(struct file *filep,const char *buff,size_t count,loff_t *offp )
{
    /* function to copy user space buffer to kernel space*/
    if ( copy_from_user(my_data,buff,count) != 0 )
        printk( "chardev: Userspace -> kernel copy failed!\n" );
    return 0;
}
```

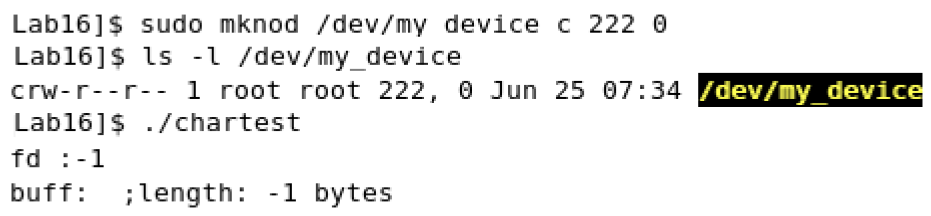
4. We'll use copy_to_user() and copy_from_user() to read/write with usermode. (we probably should do error checking here for security …)

5. Finally, we need to unregister the device when we unload the module.



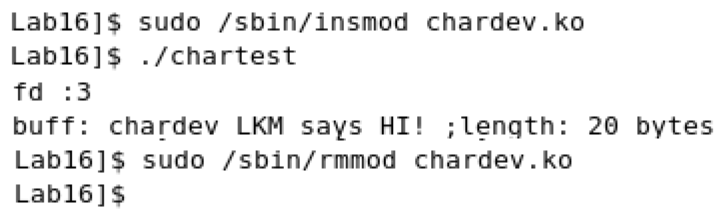
```
void cleanup(void)
{
   printk(KERN_ALERT "Unloading chardev ...\n");
   unregister_chrdev(222,"my_device");
}
```

6. To test the module we need to create the usermode device using the "mknod" command.



```
Lab16]$ sudo mknod /dev/my device c 222 0
Lab16]$ ls -l /dev/my_device
crw-r--r-- 1 root root 222, 0 Jun 25 07:34 /dev/my_device
Lab16]$ ./chartest
fd :-1
buff:  ;length: -1 bytes
```

7. Initially we get an error when running our user mode application because the module is not loaded yet and it has not registered itself with the device subsystem.



```
Lab16]$ sudo /sbin/insmod chardev.ko
Lab16]$ ./chartest
fd :3
buff: chardev LKM says HI! ;length: 20 bytes
Lab16]$ sudo /sbin/rmmod chardev.ko
Lab16]$
```