

Dynamically major number is assigned

```
desd@actsc4: ~/Documents/nugget/30002_Ashank
desd@actsc4:~/Documents/nugget/30002_Ashank$ sudo insmod char_main.ko
desd@actsc4:~/Documents/nugget/30002_Ashank$ sudo dmesg | tail -10
[ 3780.235959] simplechar driver unloaded
[ 3787.546019] simplechar driver loaded with major 236
desd@actsc4:~/Documents/nugget/30002_Ashank$
```

Using mknod to echo onto one terminal and , cat on another (aka read and write)
Fails when a 3rd minor number is added as code was only written for max of 2 minor devices

```
desd@actsc4: ~/Documents/nugget/30002_Ashank
desd@actsc4:~/Documents/nugget/30002_Ashank$ sudo mknod /dev/char_dev0 c 236 0
desd@actsc4:~/Documents/nugget/30002_Ashank$ sudo mknod /dev/char_dev0 c 236 1
mknod: /dev/char_dev0: File exists
desd@actsc4:~/Documents/nugget/30002_Ashank$ sudo mknod /dev/char_dev1 c 236 1
desd@actsc4:~/Documents/nugget/30002_Ashank$ sudo chmod 666 /dev/char_dev*
desd@actsc4:~/Documents/nugget/30002_Ashank$ echo "hello from device 0" > /dev/char_dev0
desd@actsc4:~/Documents/nugget/30002_Ashank$ echo "hello from device 1" > /dev/char_dev1
desd@actsc4:~/Documents/nugget/30002_Ashank$ sudo mknod /dev/char_dev0 c 236 2
mknod: /dev/char_dev0: File exists
desd@actsc4:~/Documents/nugget/30002_Ashank$ sudo mknod /dev/char_dev2 c 236 2
desd@actsc4:~/Documents/nugget/30002_Ashank$ echo "hello from device 2" > /dev/char_dev2
bash: /dev/char_dev2: Permission denied
desd@actsc4:~/Documents/nugget/30002_Ashank$
```

```
desd@actsc4: ~/Documents/nugget/30002_Ashank
desd@actsc4:~/Documents/nugget/30002_Ashank$ cat /dev/char_dev0
hello from device 0
^C
desd@actsc4:~/Documents/nugget/30002_Ashank$ cat /dev/char_dev1
hello from device 1
```

if the device is being to opened for the second time(in another terminal), device busy error is given to the user-space

```
desd@actsc4: ~/Documents/nugget/30002_Ashank
desd@actsc4:~/Documents/nugget/30002_Ashank$ echo "hello from device 1" > /dev/char_dev1
bash: /dev/char_dev1: Device or resource busy
desd@actsc4:~/Documents/nugget/30002_Ashank$
```

When opened **private_data** ensures that only that device can access this information

```
static int my_open(struct inode *inode, struct file *file)
{
    int minor = iminor(inode);

    if (minor >= NUM_DEVICES)
        return -ENODEV;

    struct my_device *dev = &devices[minor];

    if (dev->device_open)
        return -EBUSY;

    dev->device_open = 1;

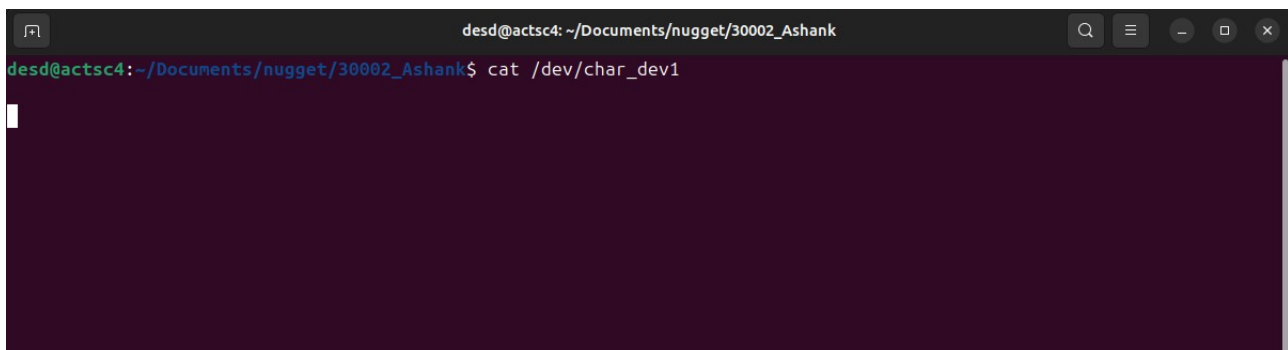
    if (!dev->buffer)
        dev->buffer = kzalloc(BUFFER_SIZE, GFP_KERNEL);

    if (!dev->buffer)
        return -ENOMEM;

    file->private_data = dev;

    pr_info("Device %d opened\n", minor);
    return 0;
}
```

readers get blocked if no data is written

A terminal window with a dark background and light text. The title bar shows the user 'desd' on host 'actsc4' in the directory '~/Documents/nugget/30002_Ashank'. The prompt is 'desd@actsc4:~/Documents/nugget/30002_Ashank\$'. The command entered is 'cat /dev/char_dev1'. The output is a single blank line, indicating that the file is empty or that no data has been written to it yet.

```
desd@actsc4: ~/Documents/nugget/30002_Ashank
desd@actsc4:~/Documents/nugget/30002_Ashank$ cat /dev/char_dev1

```