

## Assignment 6 – Coding A Device Driver

### Description:

A software device driver that takes a user's input(string), and encrypts it by using Caesar Cipher method. In my case, I make every character in the string move to its next character in the ascii table.

### Approach:

This program actually got me into a lot of trouble. The logic for writing this isn't hard, what's hard is that I'm not familiar with writing at the kernel level.

Firstly, I wrote the myOpen function, which is for users to open a file. However, the file that is being opened is my device driver file. I just tested the open function to see if it works in the test file, but open doesn't have a real use in my device driver.

For the close function, I freed every variables used vmalloc, mainly the myDS structures.

Then, I read and write. Write will first take the user's buffer, and use the cipher algorithm to encrypt the user input, and store it in the structure. Read will take what is currently stored in the structure, decipher it and then copy to the user's buffer.

For the cipher/decipher function. My program will only take numbers and letters(both caps and lower cases). It will do nothing to the numbers, and it will add 1 to the letters' ascii number. For example, the ascii code for 'a' is 97, after cipher, it will become 98, which is 'b'. For letters 'z' or 'Z', they will be rounded back to 'a' or 'A'. The decipher process is minusing 1 to the characters.

Lastly, for the ioctl function. I simply set if the command number equals 5, then everything in the myDS structure will be erased. This provides a faster way for users to empty the content in the driver.

### Issues and Resolutions:

- My first issue was I didn't know how to actually run the device driver from the test file. I did make and run in both test and Module folders. But it doesn't seem like the make in Module will create a devicedriver file in the dev folder.

- I resolved it by searching online, I found out this process of doing:

```
Sudo mknod /dev/MyCaesarCipherDriver c 415 0
```

```
Sudo chmod 666 /dev/MycaesarCipherDriver
```

This way it will create the device driver file inside of the dev folder.

- My second issue was that I didn't know how to use printk. Because printk wasn't giving me results on my terminal, which makes my program hard to debug.

- I resolved it by rewatching the lecture, I found out that printk doesn't print to the standard out. Instead, it will write its log to file. So I had to use the cat command to find out, and to debug.

- My third issue was to figure out all the new keywords, because there doesn't seem to be a lot of resources online about kernel level keywords.
- I rewatched the lecture a couple times, and also kept researching online to figure out the meaning of keywords like MKDEV, cdev\_init, register\_chrdev\_region, copy\_from\_user, copy\_to\_user. It seemed like once getting the confusion about the keywords out of the way, the program seemed to not be that hard.
- My fourth Issue was what ioctl function is used for.
- I searched online, and found out that ioctl is provided for us to write custom functions instead of only having the read and write function. So I wrote a custom function to delete the content in myDS structure.

### Screen shot of compilation:

(make for the test file)

```
student@student: ~/CSC415/csc415-assignment-6-device-driver-XuefengGuan1/Test$ make
gcc -c -o Guan_Xuefeng_HW6_main.o Guan_Xuefeng_HW6_main.c -g -I.
gcc -o Guan_Xuefeng_HW6_main Guan_Xuefeng_HW6_main.o -g -I. -l pthread
student@student:~/CSC415/csc415-assignment-6-device-driver-XuefengGuan1/Test$
```

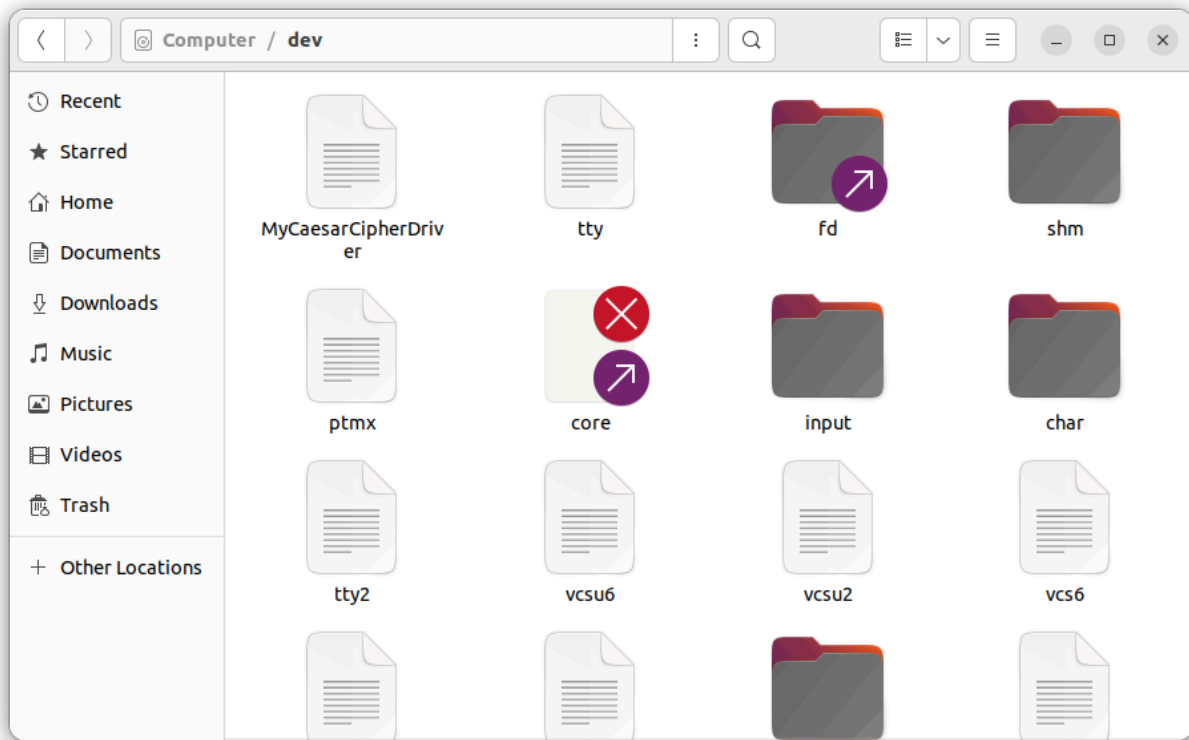
(Make for the driver)

```
student@student: ~/CSC415/csc415-assignment-6-device-driver-XuefengGuan1/Module$ make
make -C /lib/modules/`uname -r`/build M=/home/student/CSC415/csc415-assignment-6-device-driver-XuefengGuan1/Module modules
make[1]: Entering directory '/usr/src/linux-headers-6.5.0-45-generic'
warning: the compiler differs from the one used to build the kernel
The kernel was built by: x86_64-linux-gnu-gcc-12 (Ubuntu 12.3.0-1ubuntu1~22.04) 12.3.0
You are using: gcc-12 (Ubuntu 12.3.0-1ubuntu1~22.04) 12.3.0
CC [M] /home/student/CSC415/csc415-assignment-6-device-driver-XuefengGuan1/Module/CaesarCipherDriver.o
MODPOST /home/student/CSC415/csc415-assignment-6-device-driver-XuefengGuan1/Module/Module.symvers
CC [M] /home/student/CSC415/csc415-assignment-6-device-driver-XuefengGuan1/Module/CaesarCipherDriver.mod.o
LD [M] /home/student/CSC415/csc415-assignment-6-device-driver-XuefengGuan1/Module/CaesarCipherDriver.ko
BTF [M] /home/student/CSC415/csc415-assignment-6-device-driver-XuefengGuan1/Module/CaesarCipherDriver.ko
make[1]: Leaving directory '/usr/src/linux-headers-6.5.0-45-generic'
student@student:~/CSC415/csc415-assignment-6-device-driver-XuefengGuan1/Module$
```

(Create the device driver file in the dev folder)

```
student@student: ~/CSC415/csc415-assignment-6-device-driver-XuefengGuan1/Module$ make
make -C /lib/modules/`uname -r`/build M=/home/student/CSC415/csc415-assignment-6-device-driver-XuefengGuan1/Module modules
make[1]: Entering directory '/usr/src/linux-headers-6.5.0-45-generic'
warning: the compiler differs from the one used to build the kernel
The kernel was built by: x86_64-linux-gnu-gcc-12 (Ubuntu 12.3.0-1ubuntu1~22.04) 12.3.0
You are using: gcc-12 (Ubuntu 12.3.0-1ubuntu1~22.04) 12.3.0
CC [M] /home/student/CSC415/csc415-assignment-6-device-driver-XuefengGuan1/Module/CaesarCipherDriver.o
MODPOST /home/student/CSC415/csc415-assignment-6-device-driver-XuefengGuan1/Module/Module.symvers
CC [M] /home/student/CSC415/csc415-assignment-6-device-driver-XuefengGuan1/Module/CaesarCipherDriver.mod.o
LD [M] /home/student/CSC415/csc415-assignment-6-device-driver-XuefengGuan1/Module/CaesarCipherDriver.ko
BTF [M] /home/student/CSC415/csc415-assignment-6-device-driver-XuefengGuan1/Module/CaesarCipherDriver.ko
make[1]: Leaving directory '/usr/src/linux-headers-6.5.0-45-generic'
student@student:~/CSC415/csc415-assignment-6-device-driver-XuefengGuan1/Module$ sudo mknod /dev/MyCaesarCipherDriver c 415 0
student@student:~/CSC415/csc415-assignment-6-device-driver-XuefengGuan1/Module$ sudo chmod 666 /dev/MyCaesarCipherDriver
student@student:~/CSC415/csc415-assignment-6-device-driver-XuefengGuan1/Module$
```

(MyCaesarCipherDriver was created)



Screen shot(s) of the execution of the program:



```
student@student: ~/CSC415/csc415-assignment-6-device-driver-XuefengGuan1/Test$ make
gcc -c -o Guan_Xuefeng_HW6_main.o Guan_Xuefeng_HW6_main.c -g -I.
gcc -o Guan_Xuefeng_HW6_main Guan_Xuefeng_HW6_main.o -g -I. -l pthread
student@student:~/CSC415/csc415-assignment-6-device-driver-XuefengGuan1/Test$ make run
./Guan_Xuefeng_HW6_main
Wrote to device: HelloWorld123
Read from device: IfmmpXpsme123
Buffer cleared
Read from device after clearing buffer:
student@student:~/CSC415/csc415-assignment-6-device-driver-XuefengGuan1/Test$
```

### How to use my program:

My program will take a user's input and encrypt the input. The program only takes upper case, lower case letters or numbers. Use write to make the program encrypt the input, use read to retrieve your original input.

To run the program, first run make in the Module folder, and then run

Sudo mknod /dev/MyCaesarCipherDriver c 415 0

Sudo chmod 666 /dev/MycaesarCipherDriver

these two commands to install the driver in your dev folder. Then the driver should be ready to use.

If the user wants to empty the content stored in the driver structure. The User can call the ioctl function with command = 5 to achieve that.