/***

* Class: CSC-415-0 2022

* Name: Arslan Alimov

* Student ID: 916612104

* GitHub Name: Arslan Alimov

* Project: Assignment 6 – Device Driver

* File: Arslan_Alimov_assignment6_writeup.pdf

* Description: Device Driver in Kernel Space.

*   Wanted to create device driver that encrypts however didn't figure out how to. The code for encryption is in my Test Code

*   the usage for it is stringEncrypt(10,message)

**/

Assignment 6

For device driver assignment, I chose to make an encryption decryption device driver where it would initialize the device module, create files in the device where it takes 1 parameter which is a message. There is no struct, but a simple buffer that holds an information of message.

This code has 2 parts, it has user test code and module code. The purpose of this assignment is for us to learn what is it device driver and how to write our own's for future. There are different devices that our computers are using, such as keyboard, monitor, mouse etc. . All those devices use some sort of a driver to interact with out OS. We can code so much thing with the device drivers, some people might find it interesting and might create "hackings" devices with that, but for me I wanted a simple purpose of this device to encrypt and decrypt the messages that are being passed from one user to another.

The file extension .ko, is a file of our module which is being installed and expands the functionality of our Linux System Kernel. We can manually insert this modules into the kernel or with insmod command. There are lots of examples on the internet on how to create a device driver , however it was hard for me to understand it.

There are multiple issues I faced when coding this device driver. First since there is hardly any starter code provided by a professor I had to look into the videos as well as on the internet for different things, one of those things was to find a way on how to print messages there is a command such as printk() , but also we may just use pr_error e.t.c . I had trouble trying to figure out how to get buffer from user and what message user input into buffer, and therefore I was not fully able to design my device driver the way I wanted. In my C file for test program I created 2 methods Encrypt and Decrypt , however those 2 functions were not able to be used in the device module itself, there are bunch of errors that I was getting and I did not really understand why. Therefore I decided to leave it as is ,inside my .c file for test. Another hard part of this assignment was to understand what is exactly device driver looking on the internet I did not find any information on it and since my English is not the best for such things I could not proceed with my finds. I decided to submit this project since I did not really understand on how to implement everything that I had in my mind, but rather submit something that nothing. That was the biggest issue that I had with this code.

It would be really nice if professor helped me and explained how I can implement encryption and decryption functions inside my program instead of it just being there for a beautiful view of the code.

I found a way to solve professors problem with command of 2 ,we have to use our own macros to input the 2 command ,because we are hitting macros that is already premade in the driver ,therefore we are unable to output what we are looking for.

This was found on the internet which made it easy for me to distinguish between errors and such. Another finding that I found interesting is that there is an interesting macros when coding kernel Linux module , such as “IS_ERR()” it acts as a Boolean for our if statement, prints out 0 if successful else it prints some other number. And the proper way to return is return PTR_ERR

The Driver and Functions

Our driver has multiple functions as all device drivers out there: open,release, write,read,unlocked_ioctl

Open – for my function I created a method that would allocate memory to our device buffer. The buffer size is being store in my header file name Encryptor.h

The size is 1000 for my buffer_size. I decided to store my buffer in private data of my file which I Called fs. On open so we won’t have to do a global buffer , it is much easier and less lines of codes. And I calculate how many times the device was opened.

Release – this function closes my device(releases it) from being used. First I am creating a buffer and assigning to it a buffer from fs->private_data (as we know I assign it on my open). Then we check if there is a buffer present and vfree same as in non-kernel coding. For every malloc we need to free and null the buffer. This way we avoid memory leaks.

Write – This function assigns a buffer into my private buffer from fs. Creating amount of bytes that is being passed and written into a buffer. Our offset is basically our position in buffer[offset] . To calculate the bytes we need to take our size – from whatever copy_from_user passes to our buffer. And we return the amount of bytes that is being written and the offset (position) in the buffer that it was moved into.

Read – it is same as write except we do copy to user and same logic as write,instead of writing into buffer we are outputting whatever is written in our buffer to user. We return the amount of bytes that were read from our buffer.

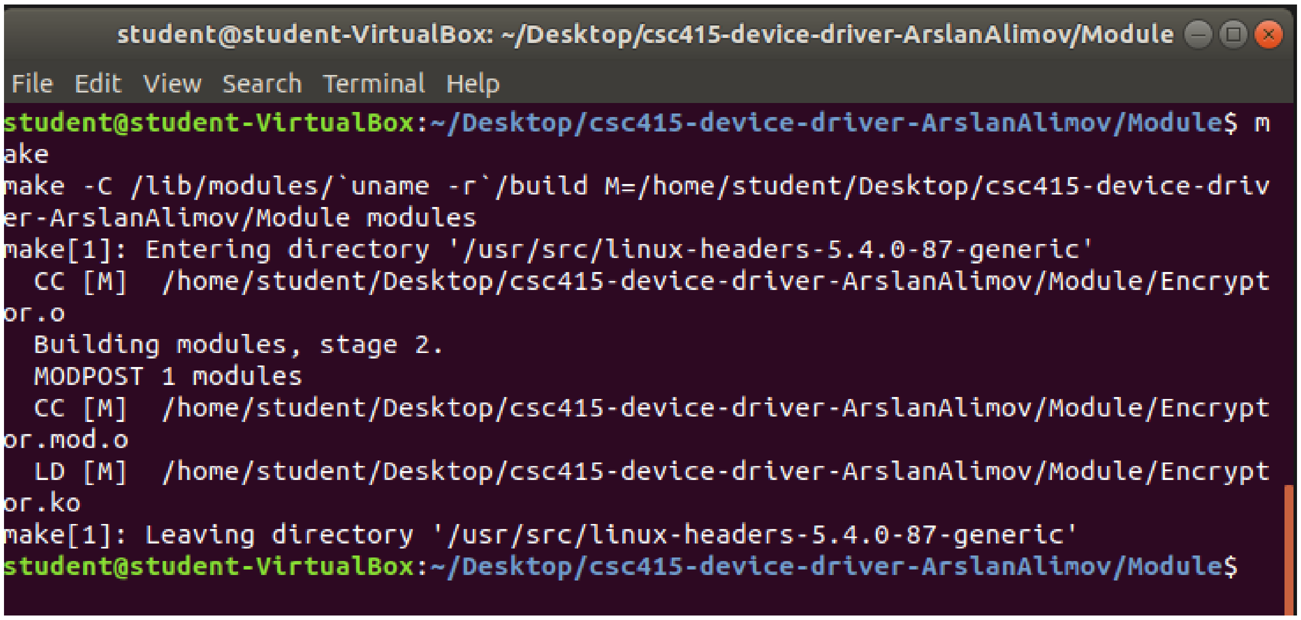
IOCTL – It is when we pass different commands to our device so it should work. However, I did not make it and did not understand how to do it.

My device driver had to encrypt strings adding a certain value to it , but I did not know how to do it outside of my test case.

-How to build our module

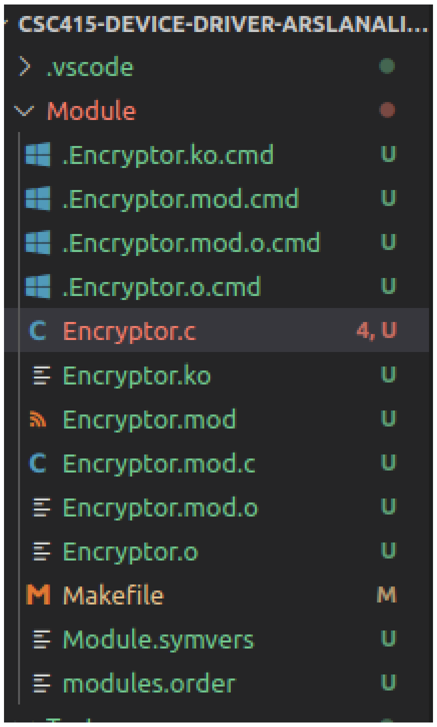
“cd Module” <=we need to make sure to go into a file with out project stored where we can see folder named “Module”

We type make in our shell as shown in the picture.


student@student-VirtualBox: ~/Desktop/csc415-device-driver-ArslanAlimov/Module
File Edit View Search Terminal Help

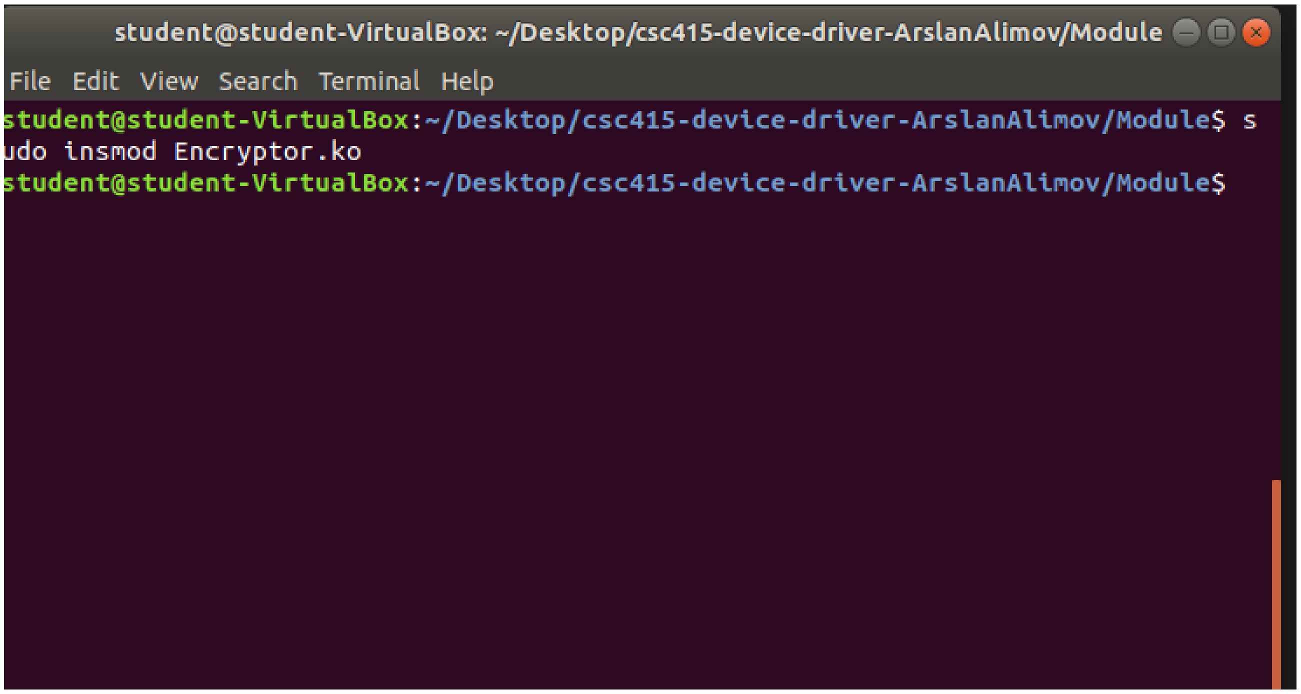
```
student@student-VirtualBox:~/Desktop/csc415-device-driver-ArslanAlimov/Module$ m
ake
make -C /lib/modules/`uname -r`/build M=/home/student/Desktop/csc415-device-driv
er-ArslanAlimov/Module modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-87-generic'
  CC [M]  /home/student/Desktop/csc415-device-driver-ArslanAlimov/Module/Encrypt
or.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC [M]  /home/student/Desktop/csc415-device-driver-ArslanAlimov/Module/Encrypt
or.mod.o
  LD [M]  /home/student/Desktop/csc415-device-driver-ArslanAlimov/Module/Encrypt
or.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-87-generic'
student@student-VirtualBox:~/Desktop/csc415-device-driver-ArslanAlimov/Module$
```

We need to make sure all files are inside our Module folder



```
CSC415-DEVICE-DRIVER-ARSLANALI...
> .vscode
v Module
  .Encryptor.ko.cmd          U
  .Encryptor.mod.cmd         U
  .Encryptor.mod.o.cmd       U
  .Encryptor.o.cmd           U
  Encryptor.c             4, U
  Encryptor.ko               U
  Encryptor.mod              U
  Encryptor.mod.c            U
  Encryptor.mod.o            U
  Encryptor.o                U
  Makefile                   M
  Module.symvers             U
  modules.order              U
```

After we did this step we need to do sudo insmod Encryptor.ko



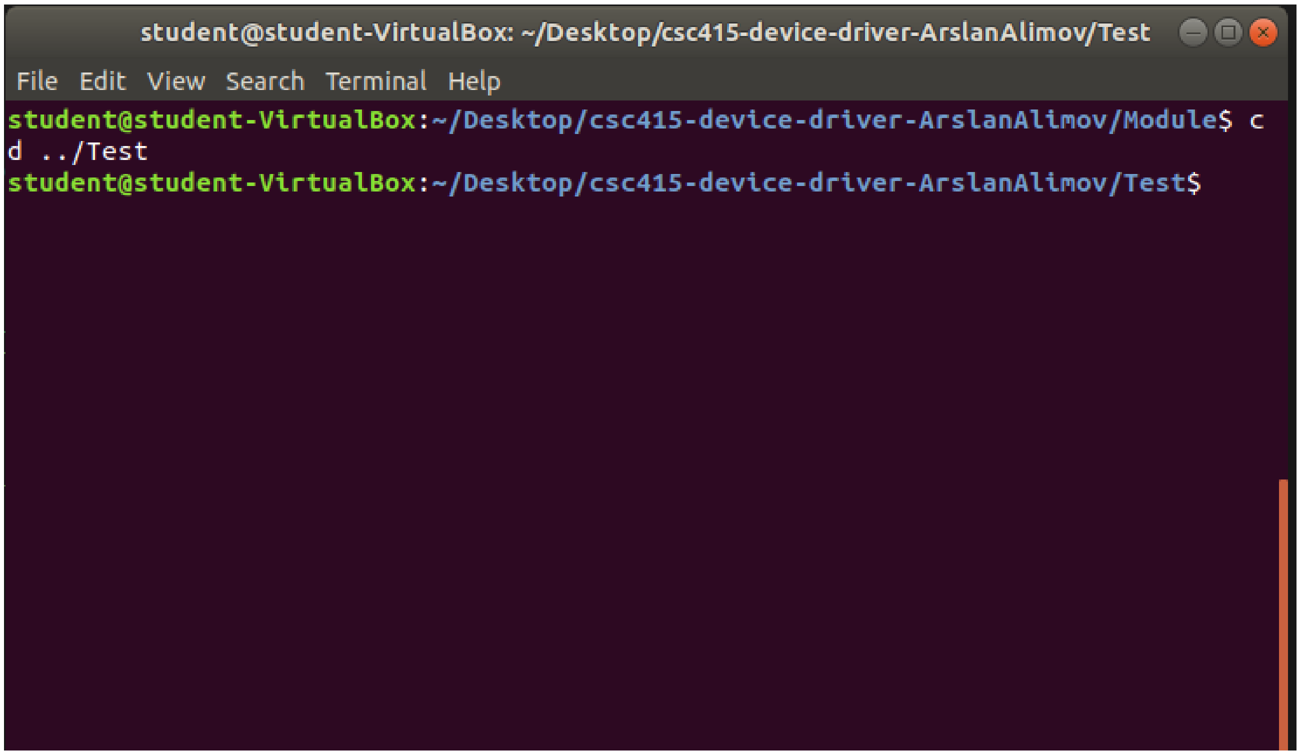
```
student@student-VirtualBox:~/Desktop/csc415-device-driver-ArslanAlimov/Module$ s
udo insmod Encryptor.ko
student@student-VirtualBox:~/Desktop/csc415-device-driver-ArslanAlimov/Module$
```

As we can see no errors showed up

Our module is successfully implemented into linux driver without any errors.

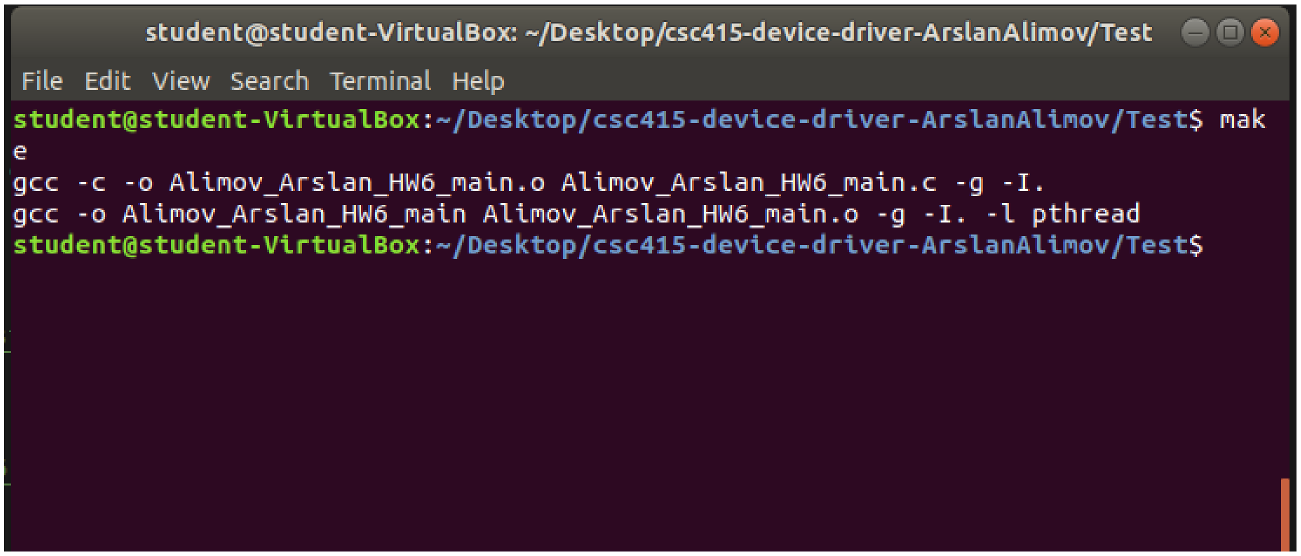
Next thing to do is get our test code running

Cd ../Test



```
student@student-VirtualBox:~/Desktop/csc415-device-driver-ArslanAlimov/Module$ c
d ../Test
student@student-VirtualBox:~/Desktop/csc415-device-driver-ArslanAlimov/Test$
```

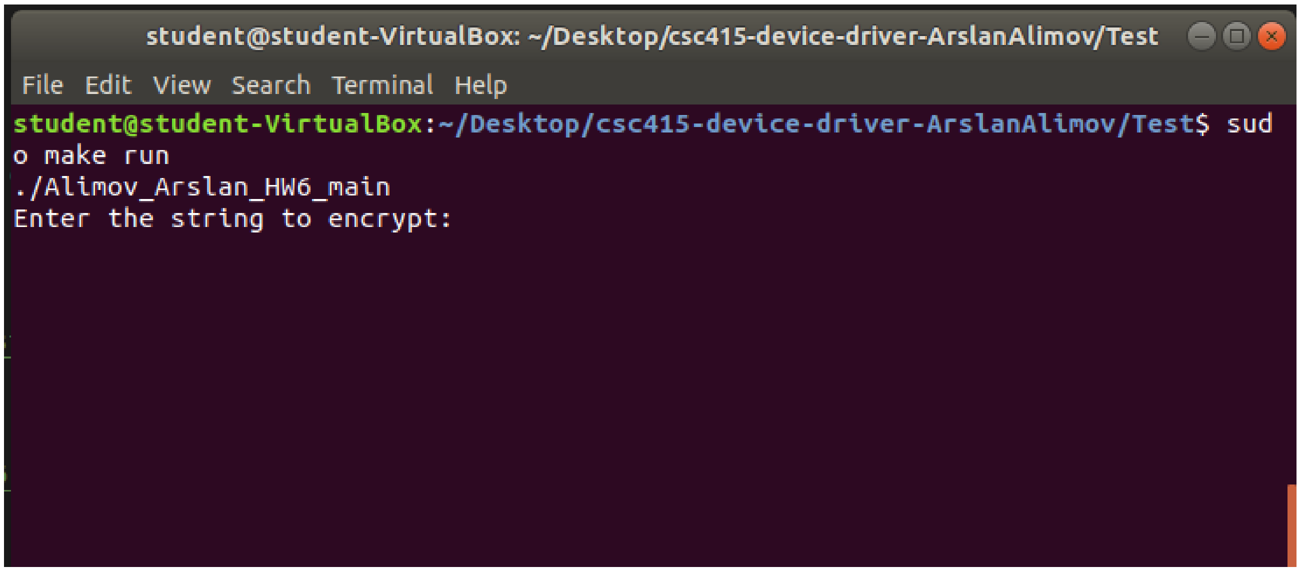
make



student@student-VirtualBox: ~/Desktop/csc415-device-driver-ArslanAlimov/Test
File Edit View Search Terminal Help

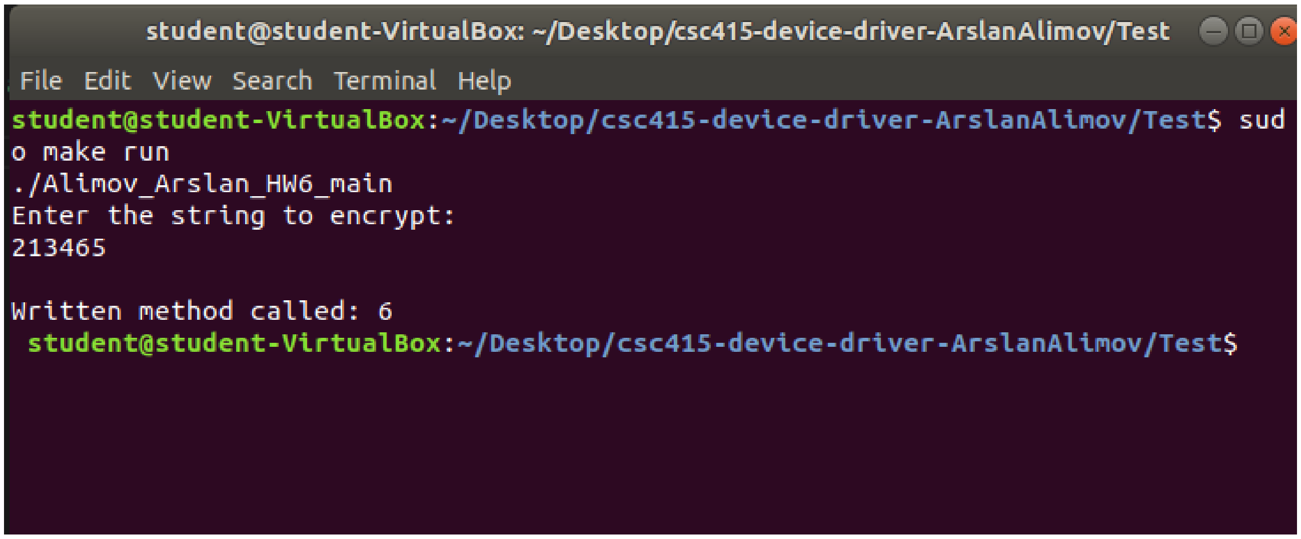
```
student@student-VirtualBox:~/Desktop/csc415-device-driver-ArslanAlimov/Test$ mak
e
gcc -c -o Alimov_Arslan_HW6_main.o Alimov_Arslan_HW6_main.c -g -I.
gcc -o Alimov_Arslan_HW6_main Alimov_Arslan_HW6_main.o -g -I. -l pthread
student@student-VirtualBox:~/Desktop/csc415-device-driver-ArslanAlimov/Test$
```

And sudo make run



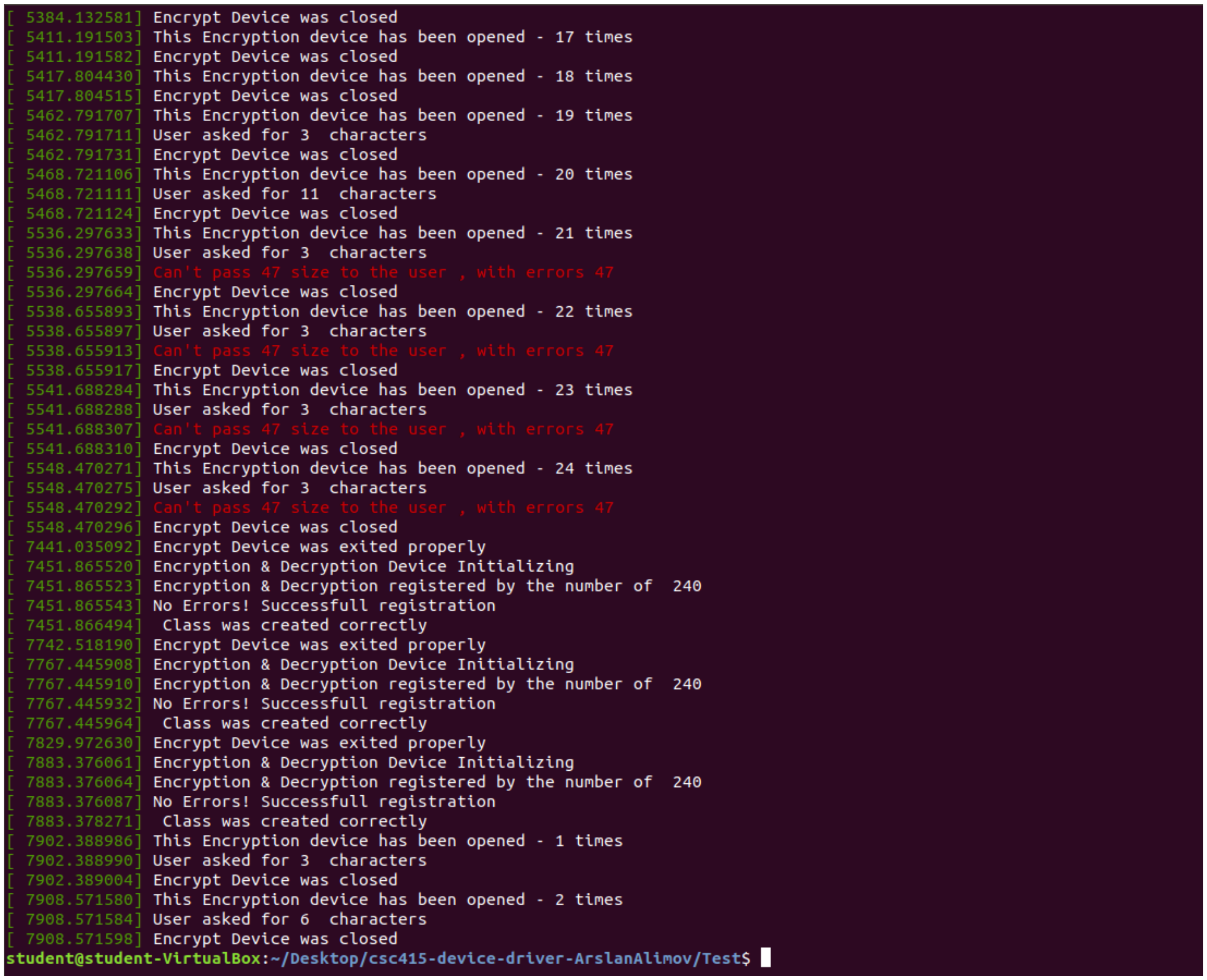
student@student-VirtualBox: ~/Desktop/csc415-device-driver-ArslanAlimov/Test
File Edit View Search Terminal Help

```
student@student-VirtualBox:~/Desktop/csc415-device-driver-ArslanAlimov/Test$ sud
o make run
./Alimov_Arslan_HW6_main
Enter the string to encrypt:
```


student@student-VirtualBox: ~/Desktop/csc415-device-driver-ArslanAlimov/Test
File Edit View Search Terminal Help

```
student@student-VirtualBox:~/Desktop/csc415-device-driver-ArslanAlimov/Test$ sud
o make run
./Alimov_Arslan_HW6_main
Enter the string to encrypt:
213465

Written method called: 6
 student@student-VirtualBox:~/Desktop/csc415-device-driver-ArslanAlimov/Test$
```

As we can see it works and outputs the amount we input into the device driver



```
[ 5384.132581] Encrypt Device was closed
[ 5411.191503] This Encryption device has been opened - 17 times
[ 5411.191582] Encrypt Device was closed
[ 5417.804430] This Encryption device has been opened - 18 times
[ 5417.804515] Encrypt Device was closed
[ 5462.791707] This Encryption device has been opened - 19 times
[ 5462.791711] User asked for 3  characters
[ 5462.791731] Encrypt Device was closed
[ 5468.721106] This Encryption device has been opened - 20 times
[ 5468.721111] User asked for 11  characters
[ 5468.721124] Encrypt Device was closed
[ 5536.297633] This Encryption device has been opened - 21 times
[ 5536.297638] User asked for 3  characters
[ 5536.297659] Can't pass 47 size to the user , with errors 47
[ 5536.297664] Encrypt Device was closed
[ 5538.655893] This Encryption device has been opened - 22 times
[ 5538.655897] User asked for 3  characters
[ 5538.655913] Can't pass 47 size to the user , with errors 47
[ 5538.655917] Encrypt Device was closed
[ 5541.688284] This Encryption device has been opened - 23 times
[ 5541.688288] User asked for 3  characters
[ 5541.688307] Can't pass 47 size to the user , with errors 47
[ 5541.688310] Encrypt Device was closed
[ 5548.470271] This Encryption device has been opened - 24 times
[ 5548.470275] User asked for 3  characters
[ 5548.470292] Can't pass 47 size to the user , with errors 47
[ 5548.470296] Encrypt Device was closed
[ 7441.035092] Encrypt Device was exited properly
[ 7451.865520] Encryption & Decryption Device Initializing
[ 7451.865523] Encryption & Decryption registered by the number of  240
[ 7451.865543] No Errors! Successfull registration
[ 7451.866494]  Class was created correctly
[ 7742.518190] Encrypt Device was exited properly
[ 7767.445908] Encryption & Decryption Device Initializing
[ 7767.445910] Encryption & Decryption registered by the number of  240
[ 7767.445932] No Errors! Successfull registration
[ 7767.445964]  Class was created correctly
[ 7829.972630] Encrypt Device was exited properly
[ 7883.376061] Encryption & Decryption Device Initializing
[ 7883.376064] Encryption & Decryption registered by the number of  240
[ 7883.376087] No Errors! Successfull registration
[ 7883.378271]  Class was created correctly
[ 7902.388986] This Encryption device has been opened - 1 times
[ 7902.388990] User asked for 3  characters
[ 7902.389004] Encrypt Device was closed
[ 7908.571580] This Encryption device has been opened - 2 times
[ 7908.571584] User asked for 6  characters
[ 7908.571598] Encrypt Device was closed
student@student-VirtualBox:~/Desktop/csc415-device-driver-ArslanAlimov/Test$
```

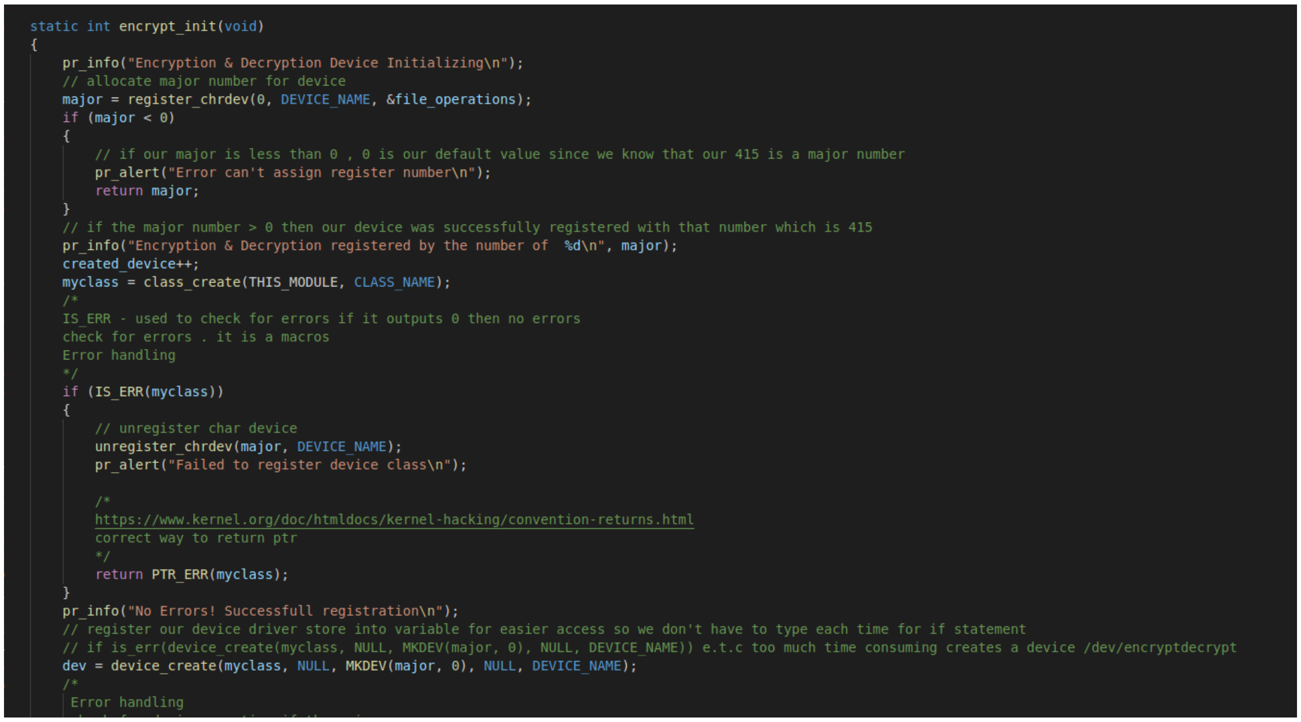
As we can see the device is being initialized

Registered the number of 240

Creating class

And exits.

It also counts how many times we opened the device driver.



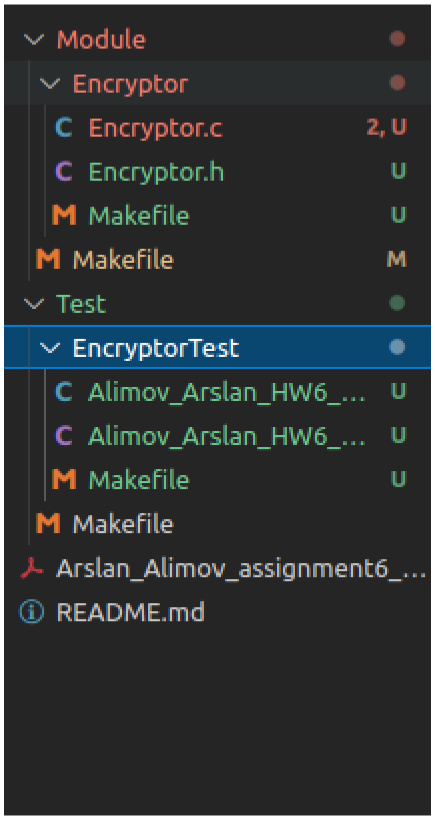
```c
static int encrypt_init(void)
{
    pr_info("Encryption & Decryption Device Initializing\n");
    // allocate major number for device
    major = register_chrdev(0, DEVICE_NAME, &file_operations);
    if (major < 0)
    {
        // if our major is less than 0 , 0 is our default value since we know that our 415 is a major number
        pr_alert("Error can't assign register number\n");
        return major;
    }
    // if the major number > 0 then our device was successfully registered with that number which is 415
    pr_info("Encryption & Decryption registered by the number of  %d\n", major);
    created_device++;
    myclass = class_create(THIS_MODULE, CLASS_NAME);
    /*
    IS_ERR - used to check for errors if it outputs 0 then no errors
    check for errors . it is a macros
    Error handling
    */
    if (IS_ERR(myclass))
    {
        // unregister char device
        unregister_chrdev(major, DEVICE_NAME);
        pr_alert("Failed to register device class\n");

        /*
        https://www.kernel.org/doc/htmldocs/kernel-hacking/convention-returns.html
        correct way to return ptr
        */
        return PTR_ERR(myclass);
    }
    pr_info("No Errors! Successfull registration\n");
    // register our device driver store into variable for easier access so we don't have to type each time for if statement
    // if is_err(device_create(myclass, NULL, MKDEV(major, 0), NULL, DEVICE_NAME)) e.t.c too much time consuming creates a device /dev/encryptdecrypt
    dev = device_create(myclass, NULL, MKDEV(major, 0), NULL, DEVICE_NAME);
    /*
     Error handling
```

I tried to include as much error handling as possible in order for the device to be able to be initialized as well I also included lots of comments for everyone to understand.

Cleaned Code a bit

Submission 2

The whole code stayed the same I just moved the pieces of code into header, for a cleaner look.
Everything compiles and works the same.



```
Module
  Encryptor
    Encryptor.c        2, U
    Encryptor.h           U
    Makefile              U
  Makefile                M
Test
  EncryptorTest
    Alimov_Arslan_HW6_...  U
    Alimov_Arslan_HW6_...  U
    Makefile               U
  Makefile
Arslan_Alimov_assignment6_...
README.md
```

New instruction on how to build the code:

First we need to compile code:

We need to type cd ../csc415-device-driver-ArslanAlimov

Inside the folder we will see Module(folder) , Makefile, Test(folder).

In order to compile our module we need to go into our shell type those commands:

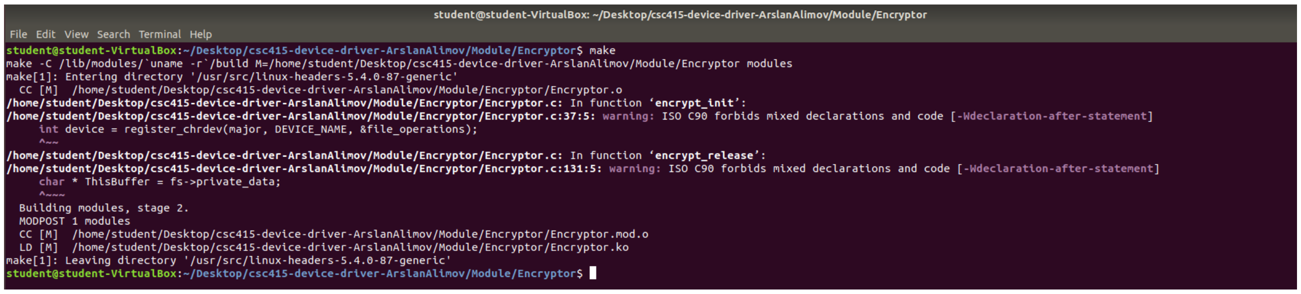
1) Make will compile all our files

   cd ../csc415-device-driver-ArslanAlimov

   cd Module

       cd Encryptor

           make

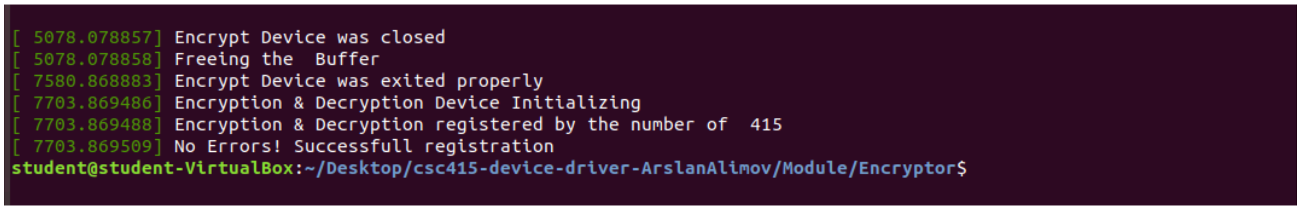


```
student@student-VirtualBox: ~/Desktop/csc415-device-driver-ArslanAlimov/Module/Encryptor
File  Edit  View  Search  Terminal  Help
student@student-VirtualBox:~/Desktop/csc415-device-driver-ArslanAlimov/Module/Encryptor$ make
make -C /lib/modules/`uname -r`/build M=/home/student/Desktop/csc415-device-driver-ArslanAlimov/Module/Encryptor modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-87-generic'
  CC [M]  /home/student/Desktop/csc415-device-driver-ArslanAlimov/Module/Encryptor/Encryptor.o
/home/student/Desktop/csc415-device-driver-ArslanAlimov/Module/Encryptor/Encryptor.c: In function ‘encrypt_init’:
/home/student/Desktop/csc415-device-driver-ArslanAlimov/Module/Encryptor/Encryptor.c:37:5: warning: ISO C90 forbids mixed declarations and code [-Wdeclaration-after-statement]
     int device = register_chrdev(major, DEVICE_NAME, &file_operations);
     ^~~
/home/student/Desktop/csc415-device-driver-ArslanAlimov/Module/Encryptor/Encryptor.c: In function ‘encrypt_release’:
/home/student/Desktop/csc415-device-driver-ArslanAlimov/Module/Encryptor/Encryptor.c:131:5: warning: ISO C90 forbids mixed declarations and code [-Wdeclaration-after-statement]
     char * ThisBuffer = fs->private_data;
     ^~~~
  Building modules, stage 2.
  MODPOST 1 modules
  CC [M]  /home/student/Desktop/csc415-device-driver-ArslanAlimov/Module/Encryptor/Encryptor.mod.o
  LD [M]  /home/student/Desktop/csc415-device-driver-ArslanAlimov/Module/Encryptor/Encryptor.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-87-generic'
student@student-VirtualBox:~/Desktop/csc415-device-driver-ArslanAlimov/Module/Encryptor$
```

as we can see it compiles

Next we need to install module into linux
we need to type sudo insmod Encryptor.ko into our shell.

To check that everything is good and it works we check command dmesg



```
[ 5078.078857] Encrypt Device was closed
[ 5078.078858] Freeing the  Buffer
[ 7580.868883] Encrypt Device was exited properly
[ 7703.869486] Encryption & Decryption Device Initializing
[ 7703.869488] Encryption & Decryption registered by the number of  415
[ 7703.869509] No Errors! Successfull registration
student@student-VirtualBox:~/Desktop/csc415-device-driver-ArslanAlimov/Module/Encryptor$
```

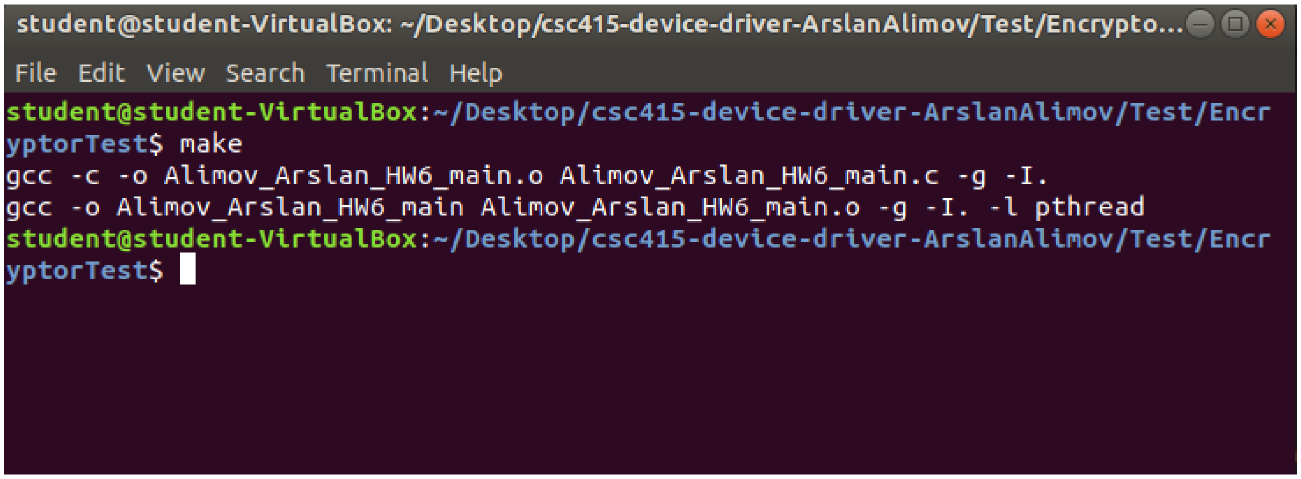
As we can see it was installed and it is working

For my test I am making simple code that does input / output from device driver

To compile it we are doing

1) Make will compile all our files
   cd ../csc415-device-driver-ArslanAlimov

   cd Test
           cd EncryptorTest
                   make

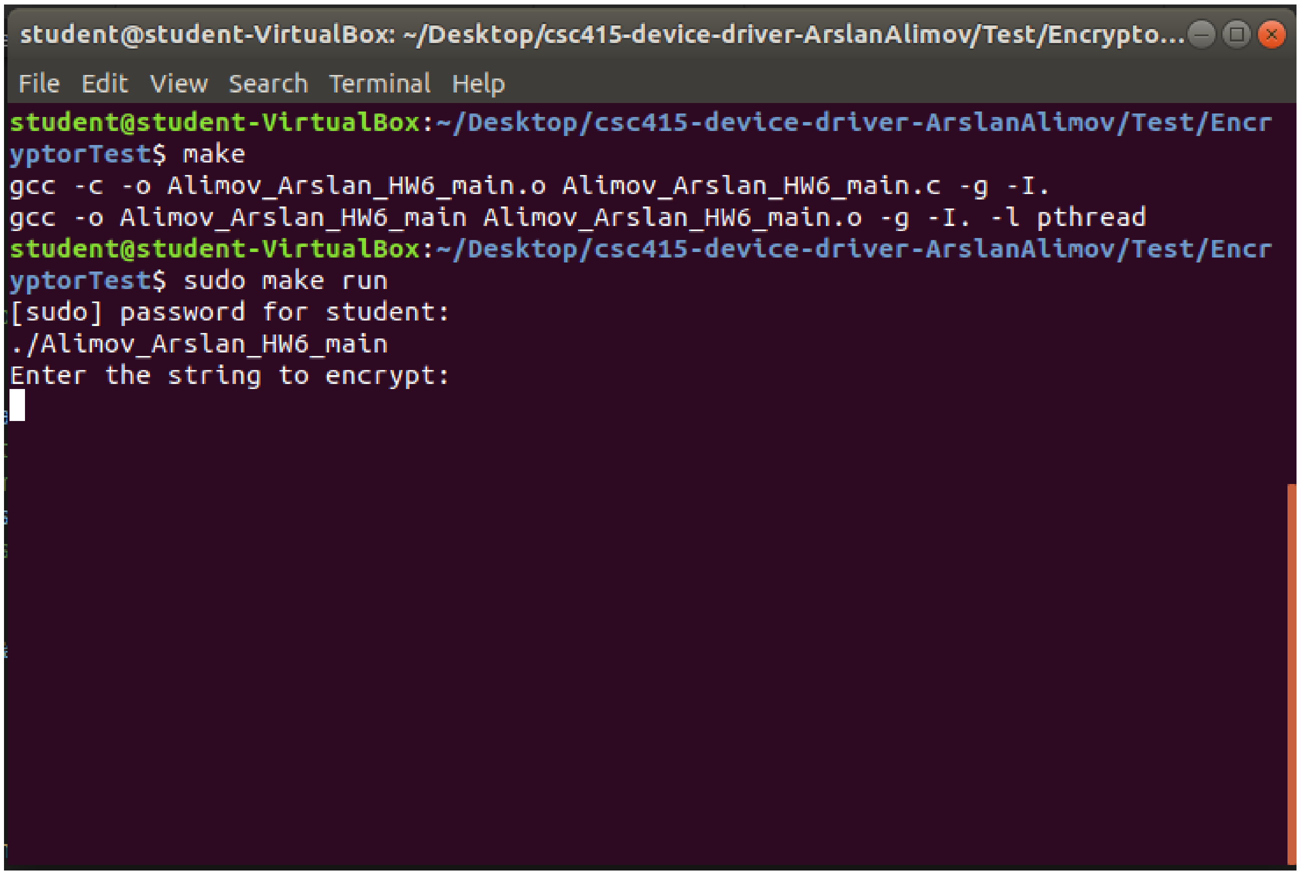


student@student-VirtualBox: ~/Desktop/csc415-device-driver-ArslanAlimov/Test/Encrypto...
File Edit View Search Terminal Help

```
student@student-VirtualBox:~/Desktop/csc415-device-driver-ArslanAlimov/Test/Encr
yptorTest$ make
gcc -c -o Alimov_Arslan_HW6_main.o Alimov_Arslan_HW6_main.c -g -I.
gcc -o Alimov_Arslan_HW6_main Alimov_Arslan_HW6_main.o -g -I. -l pthread
student@student-VirtualBox:~/Desktop/csc415-device-driver-ArslanAlimov/Test/Encr
yptorTest$ 
```

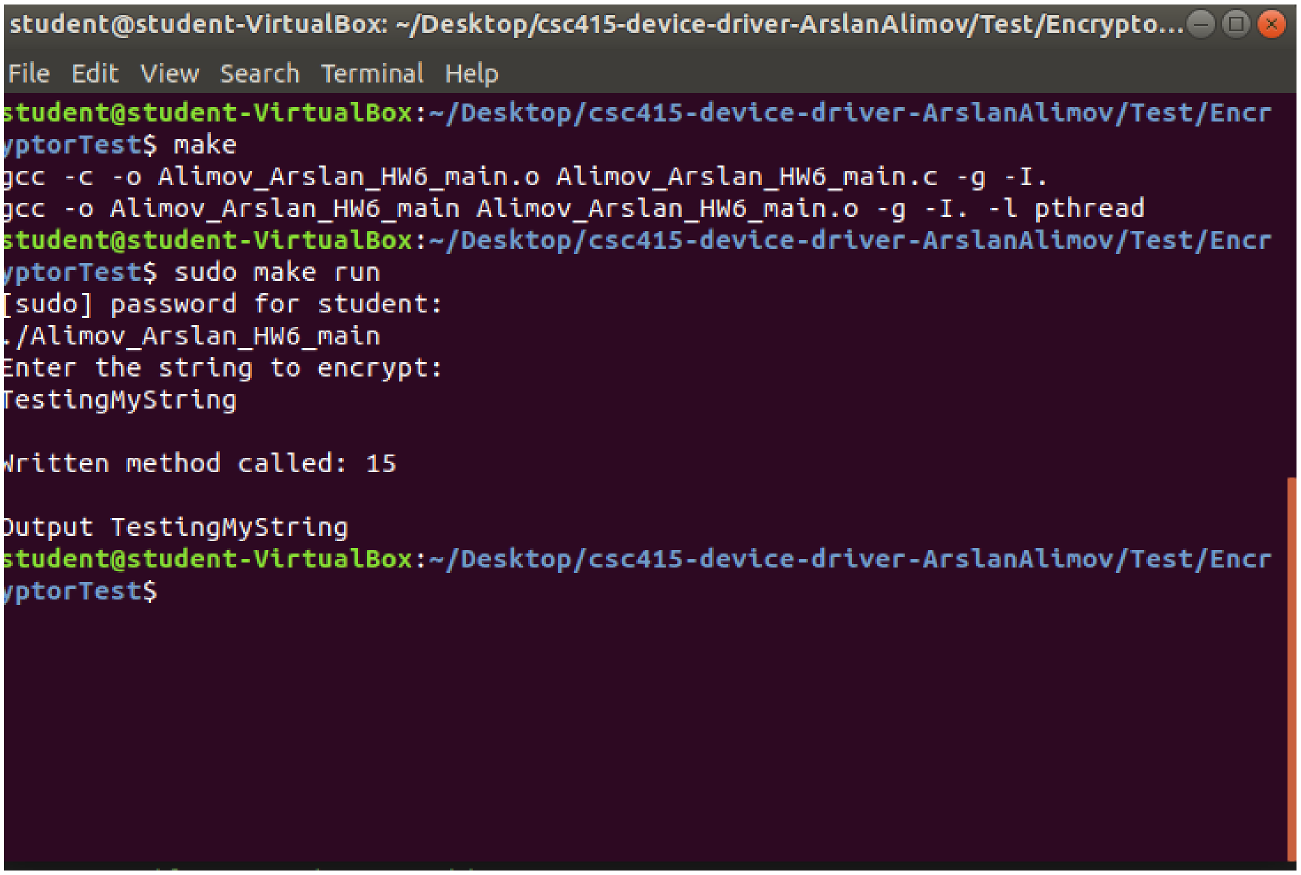
As we can see it compiles

Now to run this test we need to
sudo make run
the code will prompt us for an input of string.



```
student@student-VirtualBox:~/Desktop/csc415-device-driver-ArslanAlimov/Test/Encr
yptorTest$ make
gcc -c -o Alimov_Arslan_HW6_main.o Alimov_Arslan_HW6_main.c -g -I.
gcc -o Alimov_Arslan_HW6_main Alimov_Arslan_HW6_main.o -g -I. -l pthread
student@student-VirtualBox:~/Desktop/csc415-device-driver-ArslanAlimov/Test/Encr
yptorTest$ sudo make run
[sudo] password for student:
./Alimov_Arslan_HW6_main
Enter the string to encrypt:
```

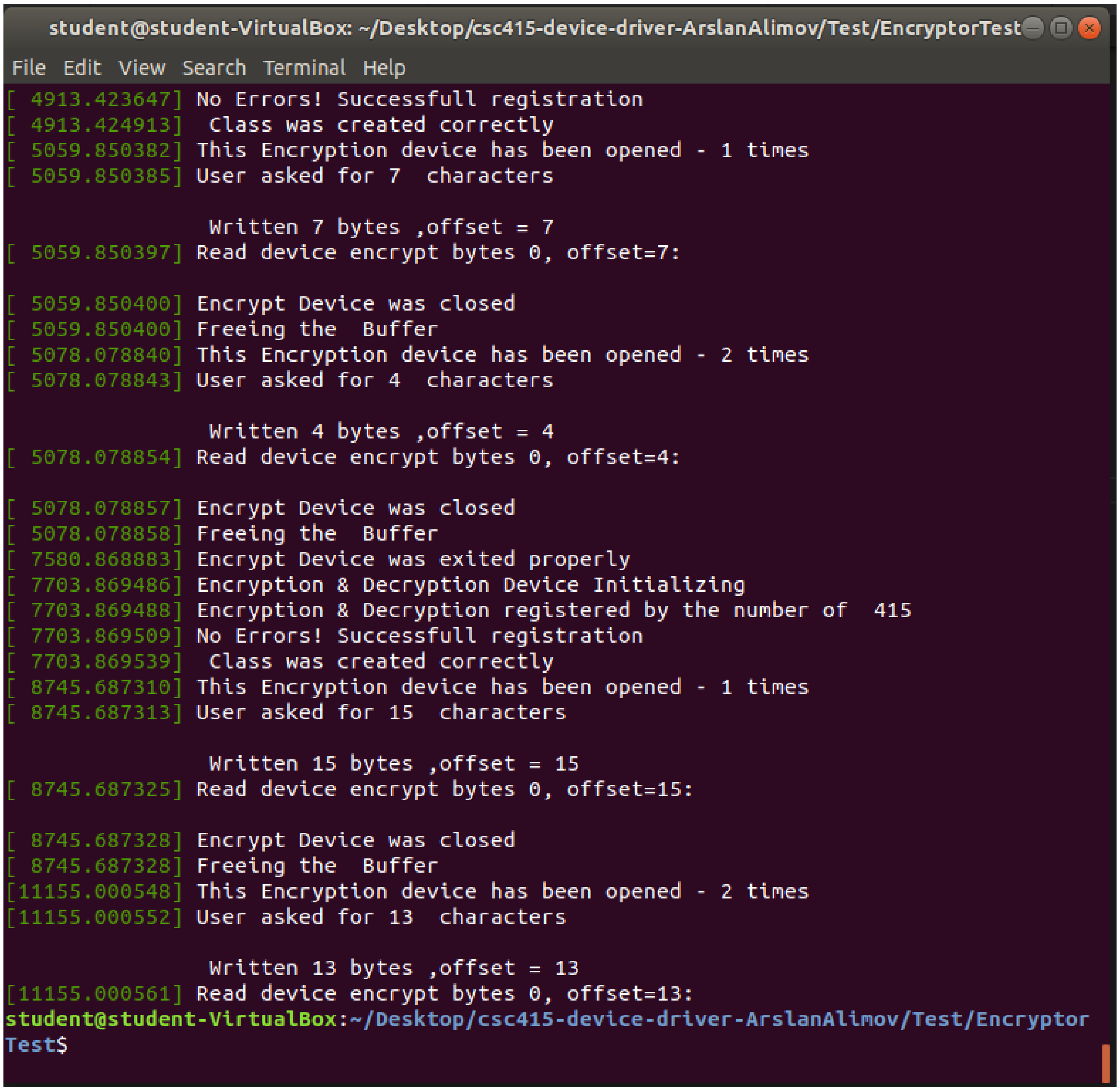
Execution:


student@student-VirtualBox: ~/Desktop/csc415-device-driver-ArslanAlimov/Test/Encrypto...
File Edit View Search Terminal Help

```
student@student-VirtualBox:~/Desktop/csc415-device-driver-ArslanAlimov/Test/Encr
yptorTest$ make
gcc -c -o Alimov_Arslan_HW6_main.o Alimov_Arslan_HW6_main.c -g -I.
gcc -o Alimov_Arslan_HW6_main Alimov_Arslan_HW6_main.o -g -I. -l pthread
student@student-VirtualBox:~/Desktop/csc415-device-driver-ArslanAlimov/Test/Encr
yptorTest$ sudo make run
[sudo] password for student:
./Alimov_Arslan_HW6_main
Enter the string to encrypt:
TestingMyString

Written method called: 15

Output TestingMyString
student@student-VirtualBox:~/Desktop/csc415-device-driver-ArslanAlimov/Test/Encr
yptorTest$
```

Written method called:15 is number of strings

Output : is the string that we got from device


student@student-VirtualBox: ~/Desktop/csc415-device-driver-ArslanAlimov/Test/EncryptorTest

File Edit View Search Terminal Help

```
[ 4913.423647] No Errors! Successfull registration
[ 4913.424913]  Class was created correctly
[ 5059.850382] This Encryption device has been opened - 1 times
[ 5059.850385] User asked for 7  characters

                Written 7 bytes ,offset = 7
[ 5059.850397] Read device encrypt bytes 0, offset=7:

[ 5059.850400] Encrypt Device was closed
[ 5059.850400] Freeing the  Buffer
[ 5078.078840] This Encryption device has been opened - 2 times
[ 5078.078843] User asked for 4  characters

                Written 4 bytes ,offset = 4
[ 5078.078854] Read device encrypt bytes 0, offset=4:

[ 5078.078857] Encrypt Device was closed
[ 5078.078858] Freeing the  Buffer
[ 7580.868883] Encrypt Device was exited properly
[ 7703.869486] Encryption & Decryption Device Initializing
[ 7703.869488] Encryption & Decryption registered by the number of  415
[ 7703.869509] No Errors! Successfull registration
[ 7703.869539]  Class was created correctly
[ 8745.687310] This Encryption device has been opened - 1 times
[ 8745.687313] User asked for 15  characters

                Written 15 bytes ,offset = 15
[ 8745.687325] Read device encrypt bytes 0, offset=15:

[ 8745.687328] Encrypt Device was closed
[ 8745.687328] Freeing the  Buffer
[11155.000548] This Encryption device has been opened - 2 times
[11155.000552] User asked for 13  characters

                Written 13 bytes ,offset = 13
[11155.000561] Read device encrypt bytes 0, offset=13:
student@student-VirtualBox:~/Desktop/csc415-device-driver-ArslanAlimov/Test/Encryptor
Test$
```