Jose Rios ID: 921160471
Github: Colorbomb1 CSC415 Operating Systems

Assignment 6 – Device Driver

Description:
This assignment is to write a C program that would develop a fully functional device driver. The device driver will be run, prompting user input, and that input will be written to the device driver. Once that input has been taken from the user it will be configured whoever, respective to its device drivers functionality. In my device driver I did a percentage grade to letter grade converter. Once the configured input is written to the device driven, the input will later be read and it will be shown to the user.

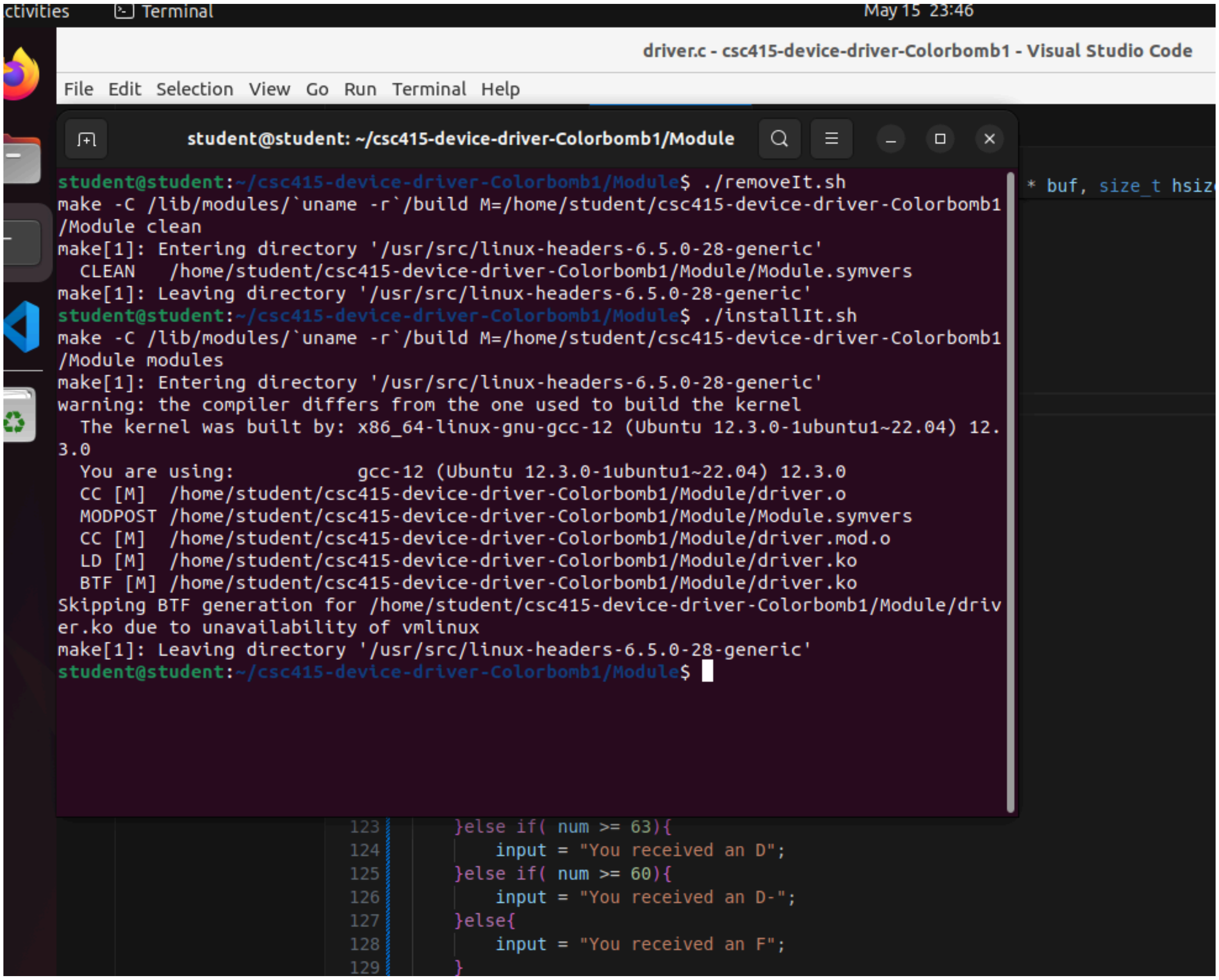
Approach:
My approach was fairly simple, covering all the material given to me about the assignment and more from the internet. From there I would construct my device driver. I began by developing my script files which are installit.sh and deleteit.sh for the module. I constructed the module and was focused on developing the open and close functions (this includes the clean up function). I found help through class material, such as the zoom lectures. I then proceeded to work on the writing and reading. Most of my time was spent on conceptualizing this part of the project and working on a lot of trial and error. I tried utilizing the read function to have whatever was stored in the buffer to be returned and that didn't work. I tried different types of ways of storing the input but I never got it working. My reading and writing were not very hand in hand. I tried to keep a simple concept that input goes to write, write takes the input and configures the input to be configured the way I want (what I want my device driver to do), write continues to take that configured input and writes it to the device driver, read reads from the device driver and returns whats ever stored in the buffer to the user, displaying it. I took a break from trying to fix the read and write solution so I started working on the functionality of what my device driver would do so I ended up choosing a percentage to letter grade converter. Essentially you enter your grade percentage and return the letter grade corresponding to that value. Once I finished writing my code for that I came back to the problem and fixed the errors I had in my driver. I was not freeing and allocating correctly. I never realized that I was not allocating space for my buffer in my device driver so that took a huge portion of my time. Later I realized I was not freeing that same memory later on so I patched that issue. After I fixed that, I continued with my approach but with more detail. I would start by getting user input and then validating that the grade is somewhere from 1 to 100 like most grades should be. I then brought it over to the write function where I store it in the buffer. I would convert the string to an integer utilizing this resource:https://www.geeksforgeeks.org/convert-string-to-integer-without-using-any-in-built-functions/# But before storing the input into my buffer I took the input and classified the grade into a letter grade and stored the letter grade and the message of what it will tell the user into the buffer. That string was then written to the buffer. I then went on to the read function where I would read the string from the buffer and send it back to the user, displaying the outcome of the percentage grade to letter grade converter device driver.

Issues and Resolutions:

Jose Rios ID: 921160471
Github: Colorbomb1 CSC415 Operating Systems

1. The main issue I had was trying to finish my read and write functions properly. I didn't know how I wanted it to be run in the sense of what it takes and what it returns and how they share a shared buffer. Writing to the shared buffer alone was an issue, I was not able to retrieve any data from the buffer itself in the driver. That meant that I was not able to get data back from the driver to the test file. I then went to check the way I was storing information to my buffer and I utilized different methods playing trial and error and finally got it working.
2. Conceptualizing what exactly the module holds and how it connects directly to the test file and driver file. I really didn't know where to start thinking when running my scripts in the module. I was not too familiar with the output it was giving me in the terminal. I wanted to know exactly how the module would contact/work with the driver file and same with the test file. What I took from it is that the module works with the driver file and the driver file works with the test file.
3. Not realizing about minor mistakes such as allocating and freeing space for certain parts of the assignment were a pain. It was very annoying because I was not looking in the places I should have been looking. I added the buffer attribute to my driver but every time I tried to do ds->buffer, I would get a segmentation fault and would continue to look in the wrong places. I then decided to actually reasonably go about debugging and read more into detail the errors that were popping up in my terminal and was able to track down the segmentation faults. By finding the places where I was supposed to be allocating and freeing memory, I was able to work more efficiently.

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

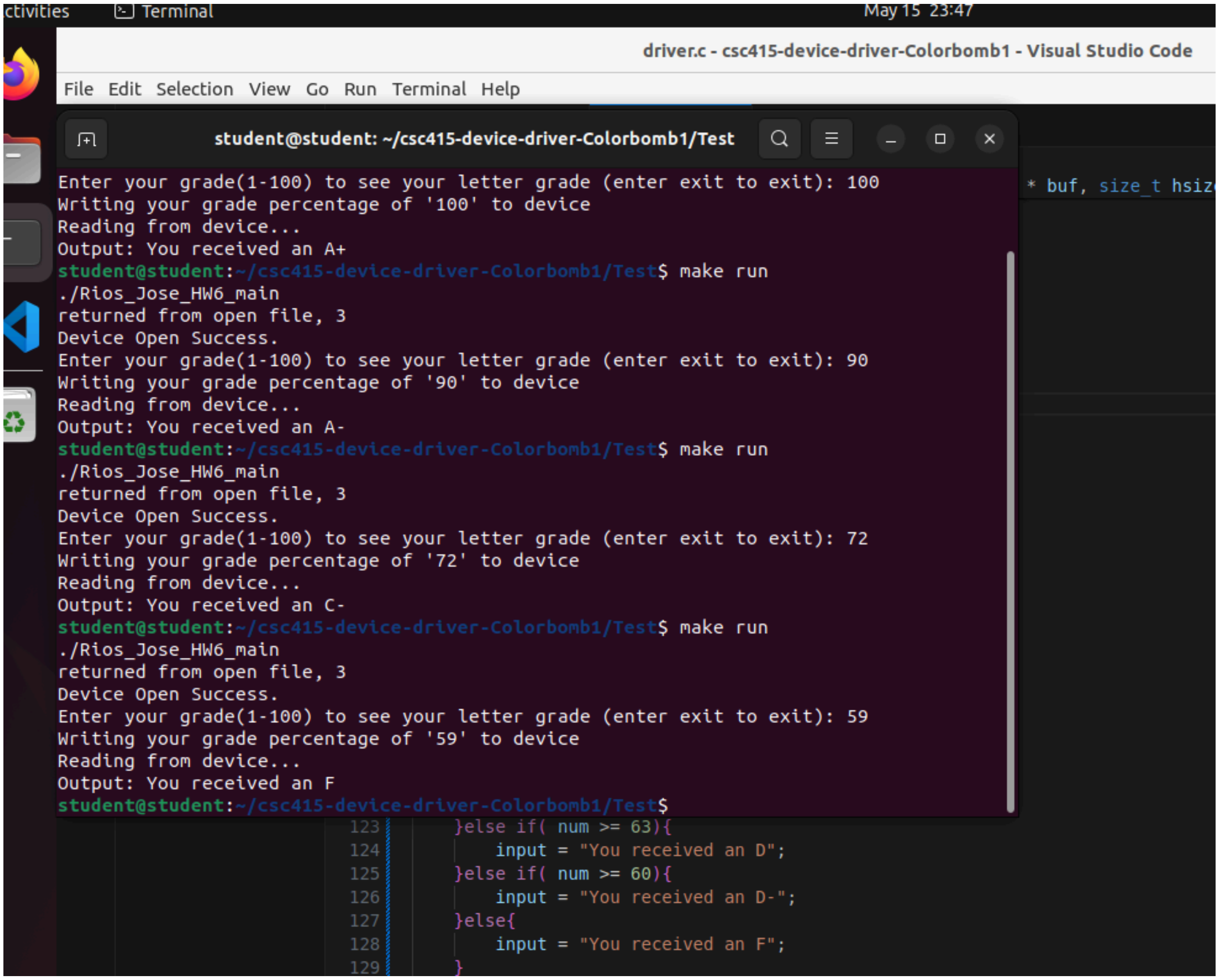

ctivities
Terminal
May 15 23:46
driver.c - csc415-device-driver-Colorbomb1 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
student@student: ~/csc415-device-driver-Colorbomb1/Module

```
student@student:~/csc415-device-driver-Colorbomb1/Module$ ./removeIt.sh
make -C /lib/modules/`uname -r`/build M=/home/student/csc415-device-driver-Colorbomb1
/Module clean
make[1]: Entering directory '/usr/src/linux-headers-6.5.0-28-generic'
  CLEAN   /home/student/csc415-device-driver-Colorbomb1/Module/Module.symvers
make[1]: Leaving directory '/usr/src/linux-headers-6.5.0-28-generic'
student@student:~/csc415-device-driver-Colorbomb1/Module$ ./installIt.sh
make -C /lib/modules/`uname -r`/build M=/home/student/csc415-device-driver-Colorbomb1
/Module modules
make[1]: Entering directory '/usr/src/linux-headers-6.5.0-28-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-device-driver-Colorbomb1/Module/driver.o
  MODPOST /home/student/csc415-device-driver-Colorbomb1/Module/Module.symvers
  CC [M]  /home/student/csc415-device-driver-Colorbomb1/Module/driver.mod.o
  LD [M]  /home/student/csc415-device-driver-Colorbomb1/Module/driver.ko
  BTF [M] /home/student/csc415-device-driver-Colorbomb1/Module/driver.ko
Skipping BTF generation for /home/student/csc415-device-driver-Colorbomb1/Module/driv
er.ko due to unavailability of vmlinux
make[1]: Leaving directory '/usr/src/linux-headers-6.5.0-28-generic'
student@student:~/csc415-device-driver-Colorbomb1/Module$
```

```
* buf, size_t hsiz
```

```
    }else if( num >= 63){
        input = "You received an D";
    }else if( num >= 60){
        input = "You received an D-";
    }else{
        input = "You received an F";
    }
```

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


ctivities
Terminal
May 15 23:47
driver.c - csc415-device-driver-Colorbomb1 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
student@student: ~/csc415-device-driver-Colorbomb1/Test
* buf, size_t hsiz

```
Enter your grade(1-100) to see your letter grade (enter exit to exit): 100
Writing your grade percentage of '100' to device
Reading from device...
Output: You received an A+
student@student:~/csc415-device-driver-Colorbomb1/Test$ make run
./Rios_Jose_HW6_main
returned from open file, 3
Device Open Success.
Enter your grade(1-100) to see your letter grade (enter exit to exit): 90
Writing your grade percentage of '90' to device
Reading from device...
Output: You received an A-
student@student:~/csc415-device-driver-Colorbomb1/Test$ make run
./Rios_Jose_HW6_main
returned from open file, 3
Device Open Success.
Enter your grade(1-100) to see your letter grade (enter exit to exit): 72
Writing your grade percentage of '72' to device
Reading from device...
Output: You received an C-
student@student:~/csc415-device-driver-Colorbomb1/Test$ make run
./Rios_Jose_HW6_main
returned from open file, 3
Device Open Success.
Enter your grade(1-100) to see your letter grade (enter exit to exit): 59
Writing your grade percentage of '59' to device
Reading from device...
Output: You received an F
student@student:~/csc415-device-driver-Colorbomb1/Test$
```

```
    }else if( num >= 63){
        input = "You received an D";
    }else if( num >= 60){
        input = "You received an D-";
    }else{
        input = "You received an F";
    }
```