Juan Segura Rico ID: 921725126
Github: velojr CSC415 Operating Systems

Assignment 6 – Coding a Device Driver

Description:

This assignment is to create a device driver of our choice and allow proper functionality using user-space interaction through standard file operations and ioctl commands. I chose a translator that takes in a string, and translates it from english to spanish or vice versa.

Approach:

While starting up the project, I thought about beginning with setting up and thinking about which device driver, and ended up choosing the translator, which enables both word translations between English and Spanish. I focused on thinking about how I can implement everything that was needed (IOCTL commands, etc.). The driver will manage these translations and toggle between English-to-Spanish and Spanish-to-English modes through IOCTL. Key file operations (read, write, and ioctl) will facilitate user-space interaction with the device, ensuring proper data handling, memory allocation, and communication. Kernel functions like vmalloc and vfree will manage dynamic memory for our structures. Then, to create our testing file, I focused on sending words via the write system call, and retrieving translations through read (as we've worked with these functions before). Afterwards, it will switch translation modes using IOCTL commands and test the translation capabilities both ways. Once these are implemented, I'll focus on debugging and testing under various scenarios to aim for proper results.

Issues and Resolutions:

My first issue was memory allocation failures in the DeviceOpen function, especially when allocating memory for the device structure (DeviceStruct) and word buffer.

- I implemented a fix by adding proper error handling to check the return value of vmalloc and clean up our previously allocated resources before returning errors.

The next issue was null terminating strings was causing translation errors

- I fixed this by adding strncpy to ensure that anything copied into the kernel buffer is properly managed, especially when it's with user-provided strings.

The next issue was when I would try to pass incorrect arguments to IOCTL and it would cause incorrect info when switching translation modes.

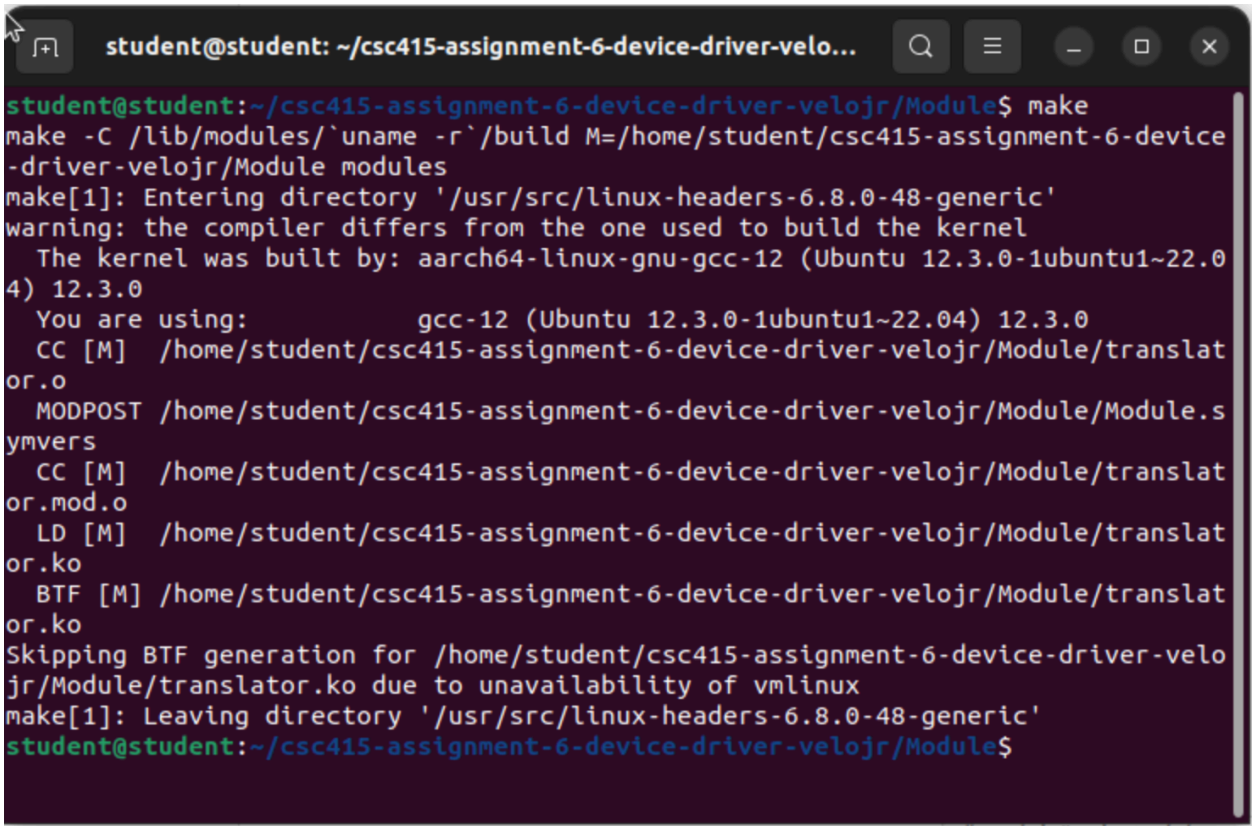
- I tried to validate cmd values and add extra error handling to ensure that only recognized commands are processed.

The final issue was trying to get it to run.

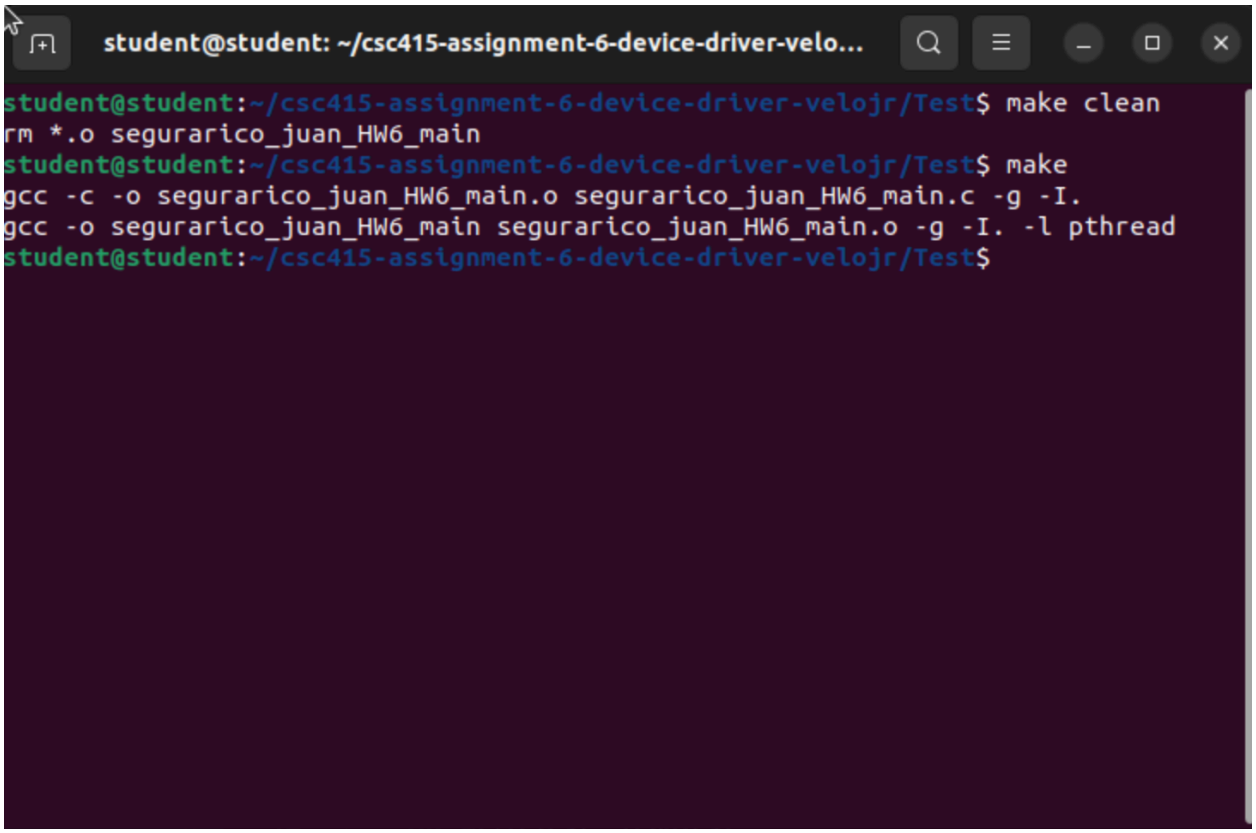
- I wasn't able to fix a problem the code had or find a way to properly run it and test it, as I ran out of time

Analysis: **N/A**

Screenshot of compilation (MODULE):



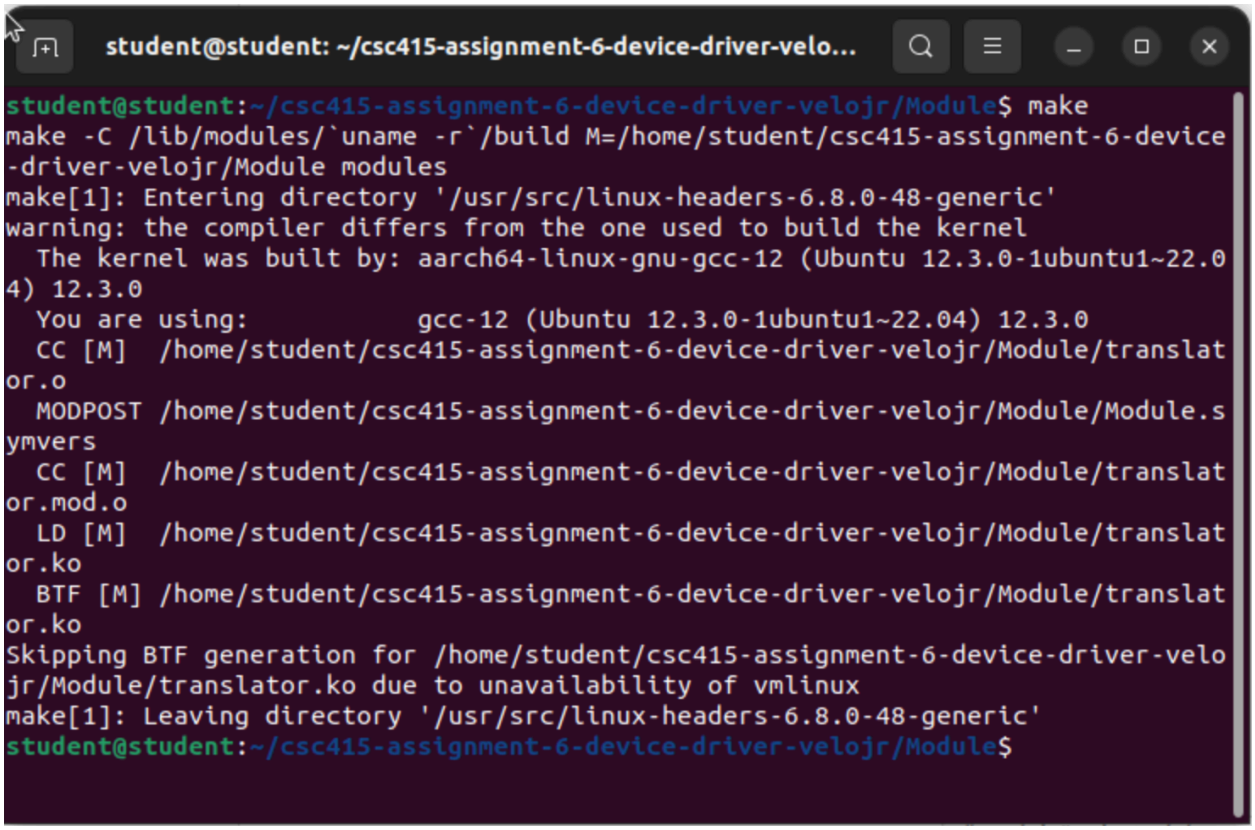
Screenshot of compilation (TEST):



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

Used:

- **sudo insmod**
- **sudo mknod**
- **sudo chmod**



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