

Russell Azucenas

917422693

GitHub ID: Ninjaruss

### Assignment 6: Device Driver

<https://github.com/CSC415-Fall2020/assignment-6-device-driver-Ninjaruss>

#### Description:

This project's aim is to gain experience working with the kernel and implementing a device driver module. The module should implement a basic skeleton device driver that can be loaded and run. Additionally, the module should include extra basic functionality; the extra functionality is memory management in this case. To test the driver, a test file must be added that demonstrates the use of the device driver.

#### Process:

I simply followed the tutorial that was related to the assignment to build a base framework for the driver. Due to many issues with the include path, it was difficult trying to implement the basic framework. I then adjusted the code to incorporate the use of `kmalloc` and `kfree` to demonstrate the device driver being able to store data in memory. For the test file, I also made some minor adjustments to the test file provided by the tutorial to incorporate my device driver.

#### Linux Kernel Module Tutorial

<http://derekmolloy.ie/writing-a-linux-kernel-module-part-2-a-character-device/>

Issues:

### Include Path

Due to the missing directories within the include path for Visual Studio Code, the driver file presented many undefined type errors for many of the types within the file. The major problem was that some of the header files could not be found on the system. These included `<linux/init.h>`, `<linux/uaccess.h>`, and `<linux/fs.h>`. In order to resolve this issue, I had to go into the C/C++ extension's settings and adjust the include path to check for the linux headers with the appropriate version.

### Storage Allocation

Since I decided to incorporate the use of a character buffer for my assignment, I had to find a way to save data within the device driver. Due to the driver not having access to the C library functions of `malloc` and `free`, I had to find similar functions that worked inside the kernel. Luckily, I found through the "Introduction to Linux Device Drivers" by Muli Ben-Yehuda PDF that there were `kmalloc` and `kfree` functions located in `<linux/slab.h>` file. I implemented those similarly to a regular `malloc` and `free` call from the previous assignments.