Chase Alexander ID: 921040156
Github: CalDevC CSC415 Operating Systems

Assignment 6 – Device Driver

Description:
This assignment was to develop a device driver with some minor functionality that can be loaded and run in Linux. Along with this device driver, we were to develop a sample application that interacted with it.

Approach / What I Did:
For my device driver, I chose to add the functionality of modifying some text provided by the user. My device driver will perform one of the following operations depending on what command number is provided:

Command	Operation
3	Convert the input to all upper case letters
4	Convert the input to all lower case letters
5	Invert the case of each letter (upper case becomes lower case and vice versa)
6	Reverse the input

I started this assignment by writing out all of the skeleton device driver code. During this process, I initially encountered some errors but was eventually able to resolve them. The open, release, write, init, and clean-up functions were all pretty straightforward. After figuring out the role of the ioctl function I moved on to the read function where most of the code specific to my device driver is held. In this function, I used a switch statement to determine what operation to perform based on the command number that was provided by the user in their call to ioctl function. Each case in the switch statement would then perform the appropriate operation on the user's provided input. These operations consisted of mostly ASCII value manipulation with the exception of the reverse operation which used a simple algorithm to reverse the user's input. This algorithm started with an iterator on each end of the string. The iterators would swap characters and then move towards the center. Once they reached the center the loop would end and the string would be reversed. Once the operation was complete I used the copy_to_user function to copy the new data into the user's buffer.

After the driver itself was complete, I moved on to making my install and uninstall scripts which would be used to load and unload the driver respectively. These were relatively simple and included echo calls to inform that the driver had been installed/uninstalled.

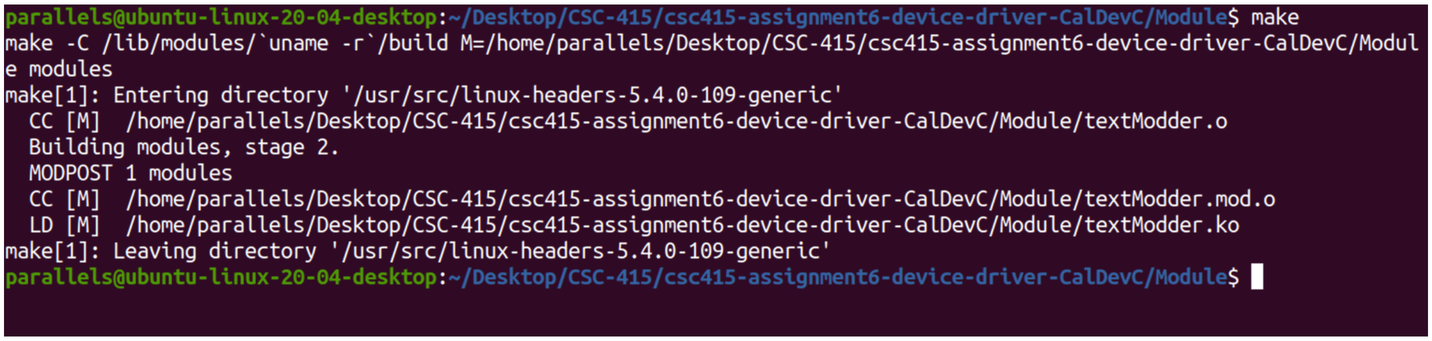
The final piece of this assignment was the user application. This part of the assignment was pretty simple, and I started by opening my device file. Once open all I had to do was call my device driver functions and pass whatever data I wished for my tests. After completing the test code for each of my four ioctl commands, I close the device file and the test application ends.

Issues and Resolutions:
My first issue was getting the basic driver code that would compile. This is something I struggled with for a while until I finally remembered that device drivers are not just normal programs. Once I realized this I stopped trying to use C libraries and started using my Makefile to figure out when I had an error instead of my IDE's IntelliSense features. The Intellisense is really what was throwing me off for a while because I thought U was having errors when I really wasn't. Using my Makefile to find errors was ultimately my resolution for this group of issues.

The other issue I faced was trying to reverse the input string. For some reason, the resulting string was always coming out as an empty string. I finally resolved this issue by subtracting 2 from the buffer position (a part of my data structure). The problem was that I was only subtracting one from this value which wasn't accounting for the index starting at 0 and the NULL character.

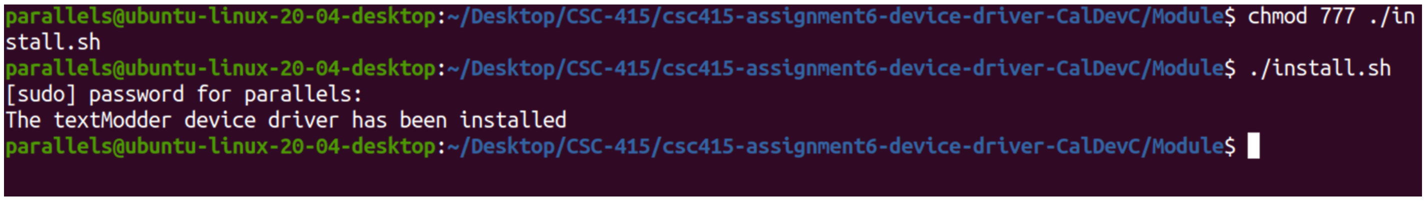
Analysis:
None required.

How to Build the Driver:
The device driver can be built by simply running "make" in the "Module" directory.



```
parallels@ubuntu-linux-20-04-desktop:~/Desktop/CSC-415/csc415-assignment6-device-driver-CalDevC/Module$ make
make -C /lib/modules/`uname -r`/build M=/home/parallels/Desktop/CSC-415/csc415-assignment6-device-driver-CalDevC/Modul
e modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-109-generic'
  CC [M]  /home/parallels/Desktop/CSC-415/csc415-assignment6-device-driver-CalDevC/Module/textModder.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC [M]  /home/parallels/Desktop/CSC-415/csc415-assignment6-device-driver-CalDevC/Module/textModder.mod.o
  LD [M]  /home/parallels/Desktop/CSC-415/csc415-assignment6-device-driver-CalDevC/Module/textModder.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-109-generic'
parallels@ubuntu-linux-20-04-desktop:~/Desktop/CSC-415/csc415-assignment6-device-driver-CalDevC/Module$
```

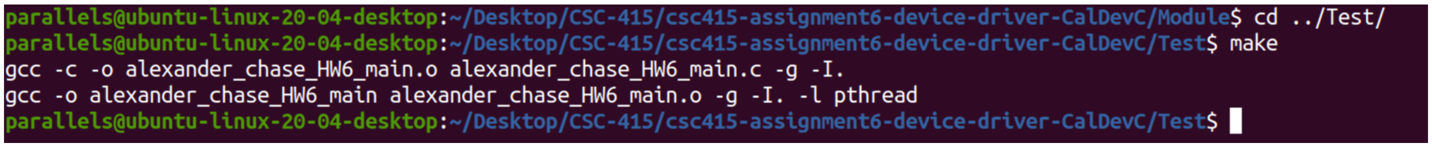
How to Load the Driver:
To load the driver, run the "install.sh" script which is also located in the "Module" directory. Before running this script you may have to run "chmod 777 ./install.sh" to make sure that it has permissions to execute.



```
parallels@ubuntu-linux-20-04-desktop:~/Desktop/CSC-415/csc415-assignment6-device-driver-CalDevC/Module$ chmod 777 ./in
stall.sh
parallels@ubuntu-linux-20-04-desktop:~/Desktop/CSC-415/csc415-assignment6-device-driver-CalDevC/Module$ ./install.sh
[sudo] password for parallels:
The textModder device driver has been installed
parallels@ubuntu-linux-20-04-desktop:~/Desktop/CSC-415/csc415-assignment6-device-driver-CalDevC/Module$
```

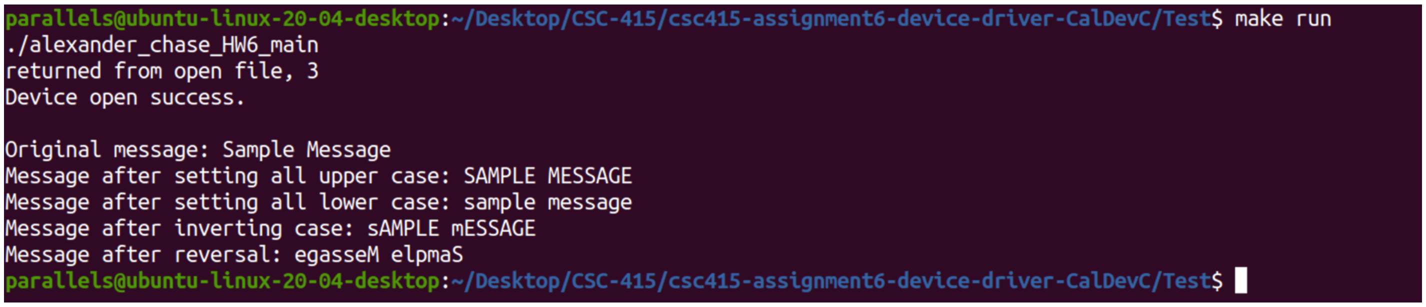
How to Interact with the Driver

To compile the test program that interacts with the driver you will need to first navigate to the "Test" directory. From here the testing application can be compiled by running the "make" command.



```
parallels@ubuntu-linux-20-04-desktop:~/Desktop/CSC-415/csc415-assignment6-device-driver-CalDevC/Module$ cd ../Test/
parallels@ubuntu-linux-20-04-desktop:~/Desktop/CSC-415/csc415-assignment6-device-driver-CalDevC/Test$ make
gcc -c -o alexander_chase_HW6_main.o alexander_chase_HW6_main.c -g -I.
gcc -o alexander_chase_HW6_main alexander_chase_HW6_main.o -g -I. -l pthread
parallels@ubuntu-linux-20-04-desktop:~/Desktop/CSC-415/csc415-assignment6-device-driver-CalDevC/Test$ 
```

Run the test program using the "make run" command within the same directory.

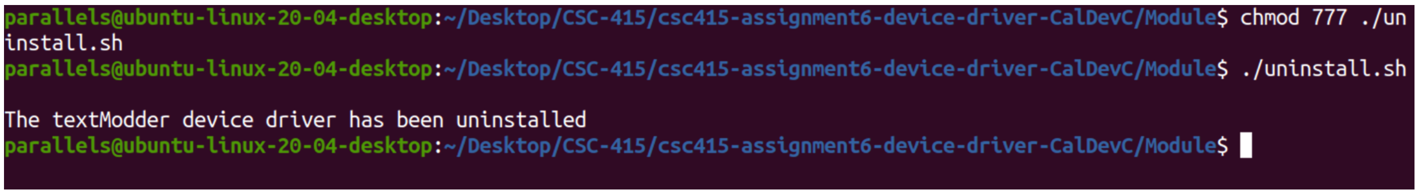


```
parallels@ubuntu-linux-20-04-desktop:~/Desktop/CSC-415/csc415-assignment6-device-driver-CalDevC/Test$ make run
./alexander_chase_HW6_main
returned from open file, 3
Device open success.

Original message: Sample Message
Message after setting all upper case: SAMPLE MESSAGE
Message after setting all lower case: sample message
Message after inverting case: sAMPLE mESSAGE
Message after reversal: egasseM elpmaS
parallels@ubuntu-linux-20-04-desktop:~/Desktop/CSC-415/csc415-assignment6-device-driver-CalDevC/Test$ 
```

How to Unload the Driver

To unload the driver, run the "uninstall.sh" script which is located in the "Module" directory. Before running this script you may have to run "chmod 777 ./uninstall.sh" to make sure that it has permissions to execute.



```
parallels@ubuntu-linux-20-04-desktop:~/Desktop/CSC-415/csc415-assignment6-device-driver-CalDevC/Module$ chmod 777 ./un
install.sh
parallels@ubuntu-linux-20-04-desktop:~/Desktop/CSC-415/csc415-assignment6-device-driver-CalDevC/Module$ ./uninstall.sh

The textModder device driver has been uninstalled
parallels@ubuntu-linux-20-04-desktop:~/Desktop/CSC-415/csc415-assignment6-device-driver-CalDevC/Module$ 
```