Andrei Georgescu
CSC 415 - 01
SFSU ID: 920776919
GitHub: https://github.com/doxify
Repository: Link
Fall 2020

Linux Device Driver Program

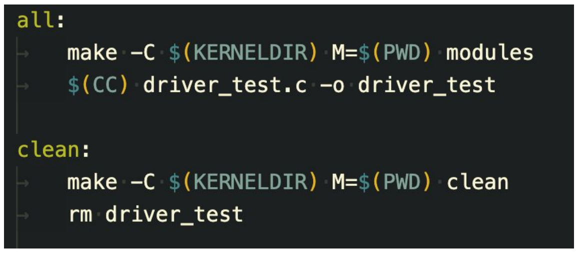
What does it do?

This program is a very basic linux device driver. The goal of the project was to create a skeleton device driver that can be loaded and run in Linux. In addition to this, I added a minor functionality where it takes in a string and returns the number of individual words in the given string. Due to the nature of the program, there are two files that makeup the overall project. The first is the kernel module which is the device driver itself and the second is the user-space program which interacts with the device driver to deliver the minor functionality.

What did I do?

Makefile:

I modified the Makefile to compile the user application when 'make' is called and to remove it when 'make clean' is called.



```
all:
	make -C $(KERNELDIR) M=$(PWD) modules
	$(CC) driver_test.c -o driver_test

clean:
	make -C $(KERNELDIR) M=$(PWD) clean
	rm driver_test
```

driver_module.c:

This is the device driver kernel module. There are five main functions that make up the simple word count driver.

cscdevchar_init() is responsible for initializing the device driver kernel module. It dynamically allocates a major number, registered the device class, and device driver. This function is called when `sudo insmod driver_module.ko` is ran.

cscdevchar_exit() is responsible for initializing everything that was initialized in the above function. It is a cleanup function. This function is called when `sudo rmmod driver_module` is called.

dev_open() is run whenever the device is opened and just keeps track of the amount of times a device has been opened.

dev_read() is responsible for sending data to the user space. This function is used when data is sent from the device and back to the user. It utilizes the copy_to_user() function to send a buffer to the user.

dev_write() is responsible for sending data to the device driver. This function is used when data is sent from the user to the device and this is where I tokenize the buffer in order to count the amount of words in the buffer. I use the strsep method to accomplish this.

dev_release() is responsible for releasing/closing the device. The user program calls this function after it has finished getting input from the user, sending it to the device, and receiving a response from the device via ioctl functions.

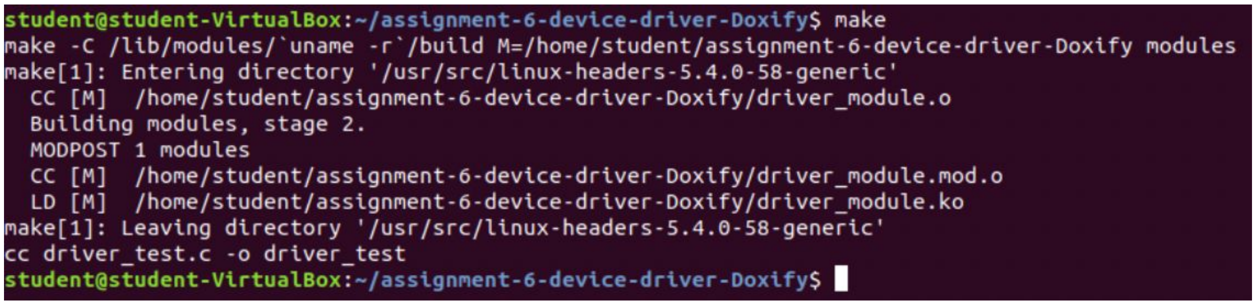
driver_test.c:
This is the user application which interacts with the driver.

I open the device driver module via `open(/dev/cscdevchar)` and then I prompt the user to input a string that is to be sent to the driver. I then write the string to the driver via the `open()` function from the device driver. If that succeeds, then we can read from the driver. I use an ioctl function to read from the kernel space. If that succeeds, then I display the response from the kernel space which is the amount of words in the inputted string.

How to Build & Run the Project:

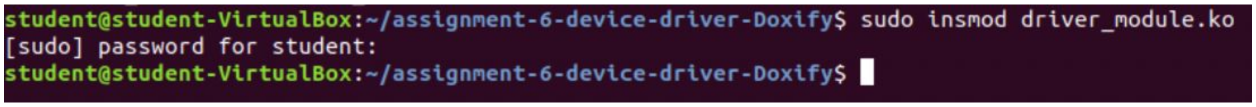
This project must be run on Linux. In addition to this, the project has only been tested on Ubuntu 18.04.1 with kernel release 5.4.0-58-generic.

1. Clone the project from github via the repository link at the top of this document.
2. Once cloned, cd into the root of the project.
    a. The root of the project contains the Makefile.
3. Run `make` to build the device driver and the user-space test program.



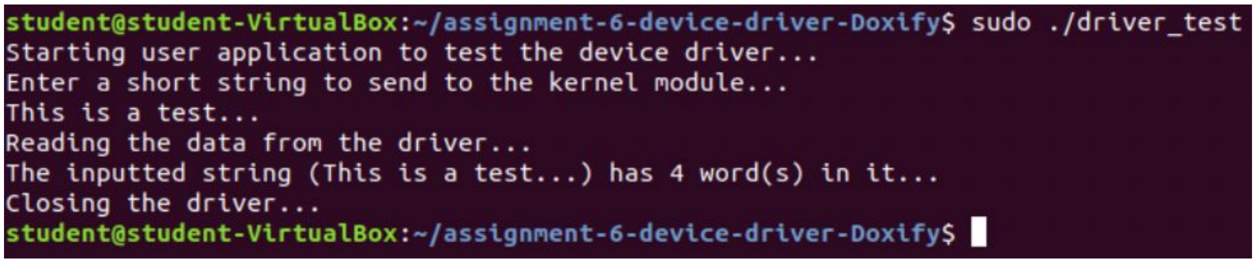
```
student@student-VirtualBox:~/assignment-6-device-driver-Doxify$ make
make -C /lib/modules/`uname -r`/build M=/home/student/assignment-6-device-driver-Doxify modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-58-generic'
  CC [M]  /home/student/assignment-6-device-driver-Doxify/driver_module.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC [M]  /home/student/assignment-6-device-driver-Doxify/driver_module.mod.o
  LD [M]  /home/student/assignment-6-device-driver-Doxify/driver_module.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-58-generic'
cc driver_test.c -o driver_test
student@student-VirtualBox:~/assignment-6-device-driver-Doxify$
```

4. Run `sudo insmod driver_module.ko` to install the device driver module.



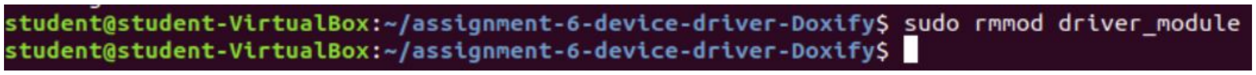
```
student@student-VirtualBox:~/assignment-6-device-driver-Doxify$ sudo insmod driver_module.ko
[sudo] password for student:
student@student-VirtualBox:~/assignment-6-device-driver-Doxify$
```

5. Run 'sudo ./driver_test' to launch the user application and respond to the prompt to test the driver module from the user space.



```
student@student-VirtualBox:~/assignment-6-device-driver-Doxify$ sudo ./driver_test
Starting user application to test the device driver...
Enter a short string to send to the kernel module...
This is a test...
Reading the data from the driver...
The inputted string (This is a test...) has 4 word(s) in it...
Closing the driver...
student@student-VirtualBox:~/assignment-6-device-driver-Doxify$
```

6. When you are ready to remove the device driver module you can run `sudo rmmod driver_module`.

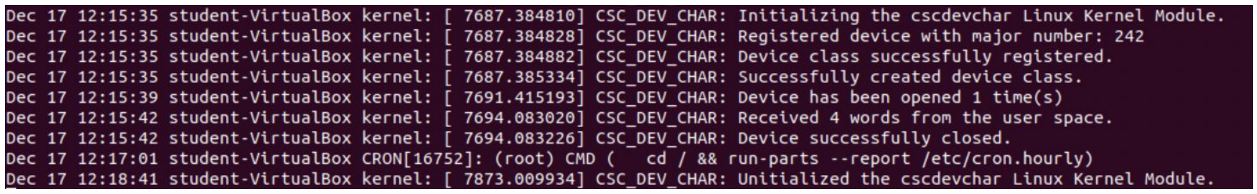


```
student@student-VirtualBox:~/assignment-6-device-driver-Doxify$ sudo rmmod driver_module
student@student-VirtualBox:~/assignment-6-device-driver-Doxify$
```

Kernel Logs:

Each driver function outputs logs to the kernel, you can view them via `tail -f /var/log/syslog`.

Here is the output after running the above commands.



```
Dec 17 12:15:35 student-VirtualBox kernel: [ 7687.384810] CSC_DEV_CHAR: Initializing the cscdevchar Linux Kernel Module.
Dec 17 12:15:35 student-VirtualBox kernel: [ 7687.384828] CSC_DEV_CHAR: Registered device with major number: 242
Dec 17 12:15:35 student-VirtualBox kernel: [ 7687.384882] CSC_DEV_CHAR: Device class successfully registered.
Dec 17 12:15:35 student-VirtualBox kernel: [ 7687.385334] CSC_DEV_CHAR: Successfully created device class.
Dec 17 12:15:39 student-VirtualBox kernel: [ 7691.415193] CSC_DEV_CHAR: Device has been opened 1 time(s)
Dec 17 12:15:42 student-VirtualBox kernel: [ 7694.083020] CSC_DEV_CHAR: Received 4 words from the user space.
Dec 17 12:15:42 student-VirtualBox kernel: [ 7694.083226] CSC_DEV_CHAR: Device successfully closed.
Dec 17 12:17:01 student-VirtualBox CRON[16752]: (root) CMD (   cd / && run-parts --report /etc/cron.hourly)
Dec 17 12:18:41 student-VirtualBox kernel: [ 7873.009934] CSC_DEV_CHAR: Unitialized the cscdevchar Linux Kernel Module.
```