Project 3 - Loadable Kernel

CSC381 Operating System Principles

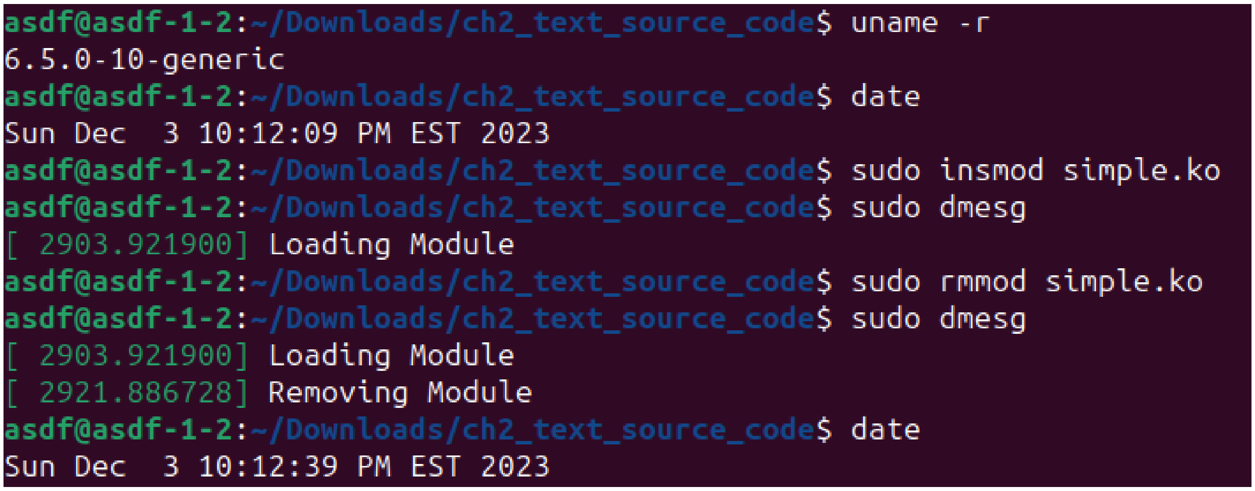
Joshua Richardson
12/4/2023

Linux supports loadable kernel modules. That is, a program written in c that is compiled into a module which can be loaded directly into the kernel without needing to recompile or restart the system.

This project required us to write several kernel modules, namely, printing the amount of jiffies, and interacting with the proc filesystem.

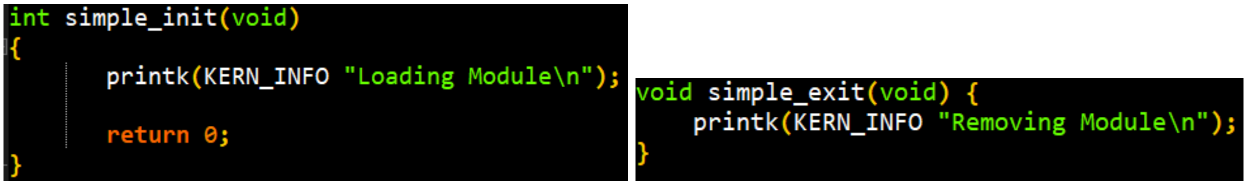
Parts I and II

Part I consisted of loading a module and observing what it printed.



```
asdf@asdf-1-2:~/Downloads/ch2_text_source_code$ uname -r
6.5.0-10-generic
asdf@asdf-1-2:~/Downloads/ch2_text_source_code$ date
Sun Dec  3 10:12:09 PM EST 2023
asdf@asdf-1-2:~/Downloads/ch2_text_source_code$ sudo insmod simple.ko
asdf@asdf-1-2:~/Downloads/ch2_text_source_code$ sudo dmesg
[ 2903.921900] Loading Module
asdf@asdf-1-2:~/Downloads/ch2_text_source_code$ sudo rmmod simple.ko
asdf@asdf-1-2:~/Downloads/ch2_text_source_code$ sudo dmesg
[ 2903.921900] Loading Module
[ 2921.886728] Removing Module
asdf@asdf-1-2:~/Downloads/ch2_text_source_code$ date
Sun Dec  3 10:12:39 PM EST 2023
```

The module prints “Loading Module” when it is loaded, and “Removing Module” when removed.

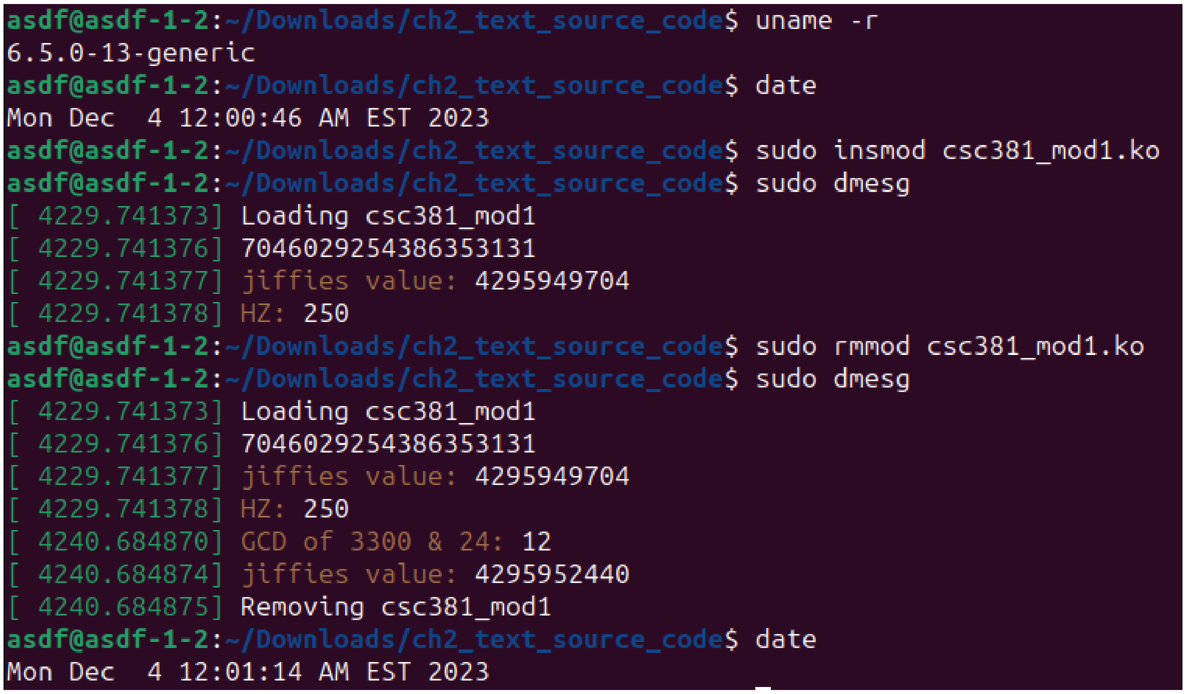


```
int simple_init(void)
{
        printk(KERN_INFO "Loading Module\n");

        return 0;
}
```

```
void simple_exit(void) {
    printk(KERN_INFO "Removing Module\n");
}
```

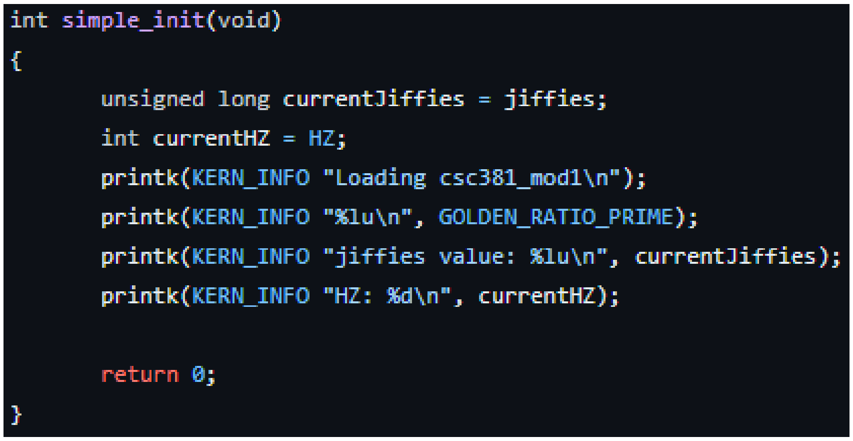
These two blocks of code are what is responsible for printing the text.

Part II asks to print the value of “GOLDEN_RATIO_PRIME” and the values of “jiffies” and “HZ” when the module is loaded. It also asks to print the greatest common divisor of 3300 and 24, as well as the value of “jiffies” when it is removed.

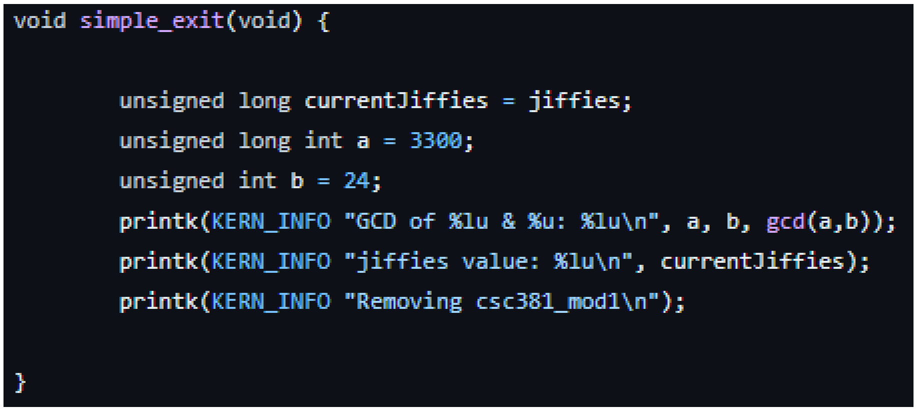


```
asdf@asdf-1-2:~/Downloads/ch2_text_source_code$ uname -r
6.5.0-13-generic
asdf@asdf-1-2:~/Downloads/ch2_text_source_code$ date
Mon Dec  4 12:00:46 AM EST 2023
asdf@asdf-1-2:~/Downloads/ch2_text_source_code$ sudo insmod csc381_mod1.ko
asdf@asdf-1-2:~/Downloads/ch2_text_source_code$ sudo dmesg
[ 4229.741373] Loading csc381_mod1
[ 4229.741376] 7046029254386353131
[ 4229.741377] jiffies value: 4295949704
[ 4229.741378] HZ: 250
asdf@asdf-1-2:~/Downloads/ch2_text_source_code$ sudo rmmod csc381_mod1.ko
asdf@asdf-1-2:~/Downloads/ch2_text_source_code$ sudo dmesg
[ 4229.741373] Loading csc381_mod1
[ 4229.741376] 7046029254386353131
[ 4229.741377] jiffies value: 4295949704
[ 4229.741378] HZ: 250
[ 4240.684870] GCD of 3300 & 24: 12
[ 4240.684874] jiffies value: 4295952440
[ 4240.684875] Removing csc381_mod1
asdf@asdf-1-2:~/Downloads/ch2_text_source_code$ date
Mon Dec  4 12:01:14 AM EST 2023
```

The code responsible for making this happen is located in the file “csc381_mod1” within the “simple_init” and “simple_exit” functions. It should be noted that certain libraries must be included in the c file.



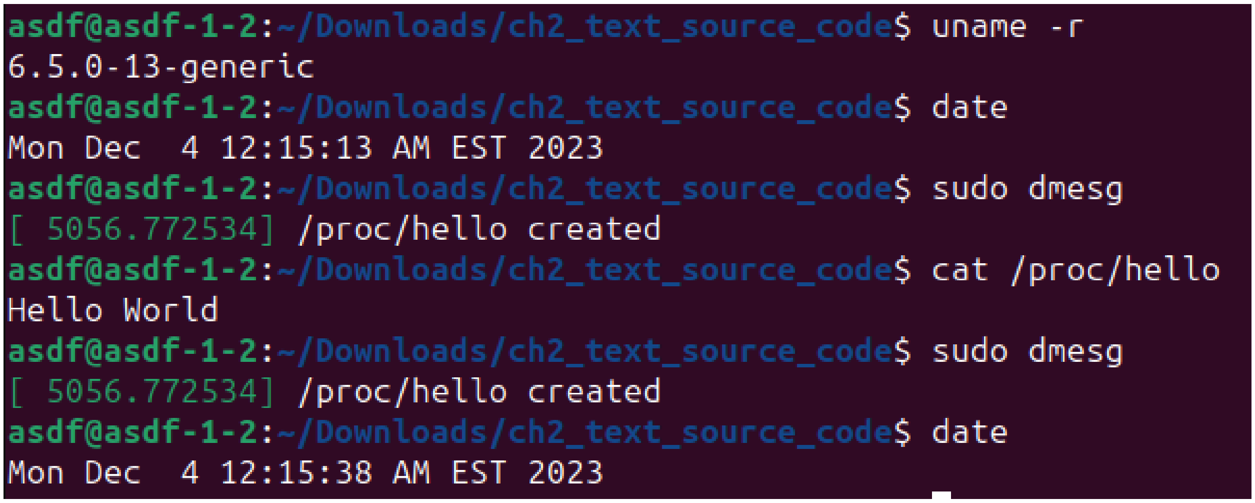
```
int simple_init(void)
{
        unsigned long currentJiffies = jiffies;
        int currentHZ = HZ;
        printk(KERN_INFO "Loading csc381_mod1\n");
        printk(KERN_INFO "%lu\n", GOLDEN_RATIO_PRIME);
        printk(KERN_INFO "jiffies value: %lu\n", currentJiffies);
        printk(KERN_INFO "HZ: %d\n", currentHZ);

        return 0;
}
```



```
void simple_exit(void) {

        unsigned long currentJiffies = jiffies;
        unsigned long int a = 3300;
        unsigned int b = 24;
        printk(KERN_INFO "GCD of %lu & %u: %lu\n", a, b, gcd(a,b));
        printk(KERN_INFO "jiffies value: %lu\n", currentJiffies);
        printk(KERN_INFO "Removing csc381_mod1\n");

}
```

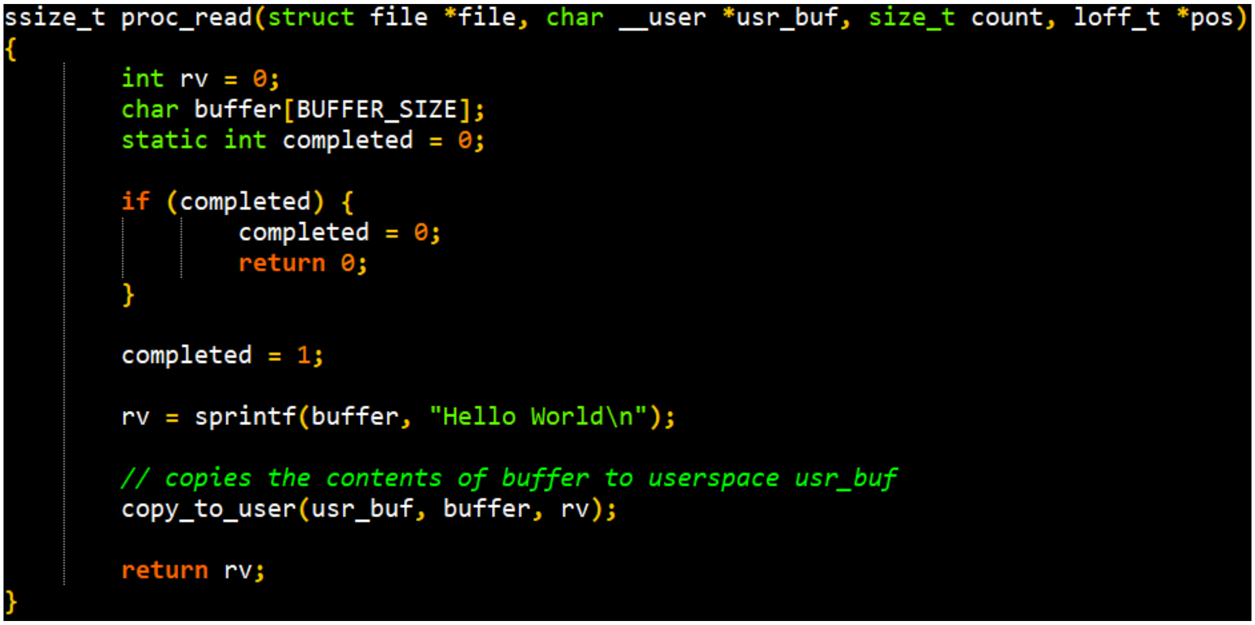
Parts III and IV

Part III asks to load a module named “hello2” This module interacts with the proc filesystem. After loading the module, running the command “cat /proc/hello” will print the text “Hello World.”



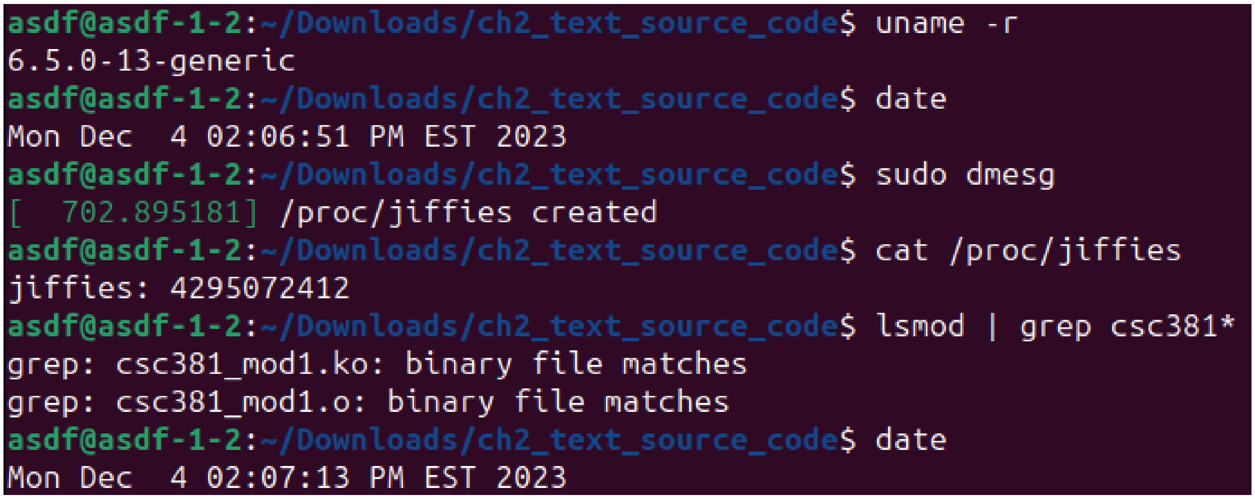
```
asdf@asdf-1-2:~/Downloads/ch2_text_source_code$ uname -r
6.5.0-13-generic
asdf@asdf-1-2:~/Downloads/ch2_text_source_code$ date
Mon Dec  4 12:15:13 AM EST 2023
asdf@asdf-1-2:~/Downloads/ch2_text_source_code$ sudo dmesg
[ 5056.772534] /proc/hello created
asdf@asdf-1-2:~/Downloads/ch2_text_source_code$ cat /proc/hello
Hello World
asdf@asdf-1-2:~/Downloads/ch2_text_source_code$ sudo dmesg
[ 5056.772534] /proc/hello created
asdf@asdf-1-2:~/Downloads/ch2_text_source_code$ date
Mon Dec  4 12:15:38 AM EST 2023
```

The code to print this is within the proc_read function in hello2.c



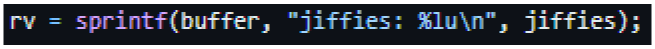
```
ssize_t proc_read(struct file *file, char __user *usr_buf, size_t count, loff_t *pos)
{
        int rv = 0;
        char buffer[BUFFER_SIZE];
        static int completed = 0;

        if (completed) {
                completed = 0;
                return 0;
        }

        completed = 1;

        rv = sprintf(buffer, "Hello World\n");

        // copies the contents of buffer to userspace usr_buf
        copy_to_user(usr_buf, buffer, rv);

        return rv;
}
```

Part IV combines previous parts, by asking to create a kernel module that returns the value of “jiffies” when “cat /proc/jiffies” is called.



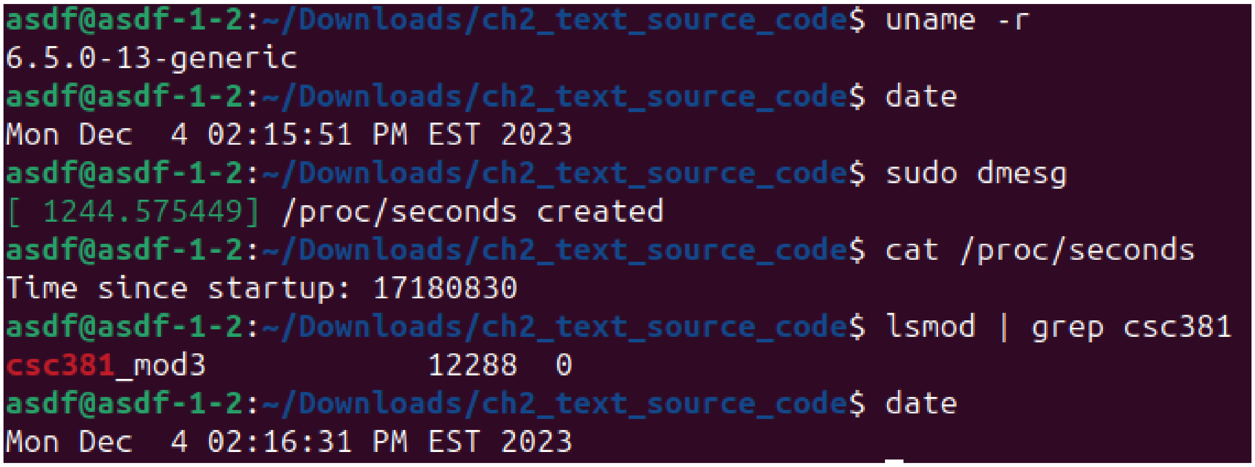
```
asdf@asdf-1-2:~/Downloads/ch2_text_source_code$ uname -r
6.5.0-13-generic
asdf@asdf-1-2:~/Downloads/ch2_text_source_code$ date
Mon Dec  4 02:06:51 PM EST 2023
asdf@asdf-1-2:~/Downloads/ch2_text_source_code$ sudo dmesg
[  702.895181] /proc/jiffies created
asdf@asdf-1-2:~/Downloads/ch2_text_source_code$ cat /proc/jiffies
jiffies: 4295072412
asdf@asdf-1-2:~/Downloads/ch2_text_source_code$ lsmod | grep csc381*
grep: csc381_mod1.ko: binary file matches
grep: csc381_mod1.o: binary file matches
asdf@asdf-1-2:~/Downloads/ch2_text_source_code$ date
Mon Dec  4 02:07:13 PM EST 2023
```

In the c file, some libraries must be included, and the proc_name must be properly defined to jiffies. This line from the proc_read function is responsible for getting the information to print the amount of jiffies.



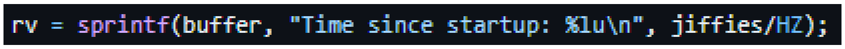
```
rv = sprintf(buffer, "jiffies: %lu\n", jiffies);
```

Part IV also asks to create a second module that prints the amount of time in seconds since startup, using jiffies and HZ.



```
asdf@asdf-1-2:~/Downloads/ch2_text_source_code$ uname -r
6.5.0-13-generic
asdf@asdf-1-2:~/Downloads/ch2_text_source_code$ date
Mon Dec  4 02:15:51 PM EST 2023
asdf@asdf-1-2:~/Downloads/ch2_text_source_code$ sudo dmesg
[ 1244.575449] /proc/seconds created
asdf@asdf-1-2:~/Downloads/ch2_text_source_code$ cat /proc/seconds
Time since startup: 17180830
asdf@asdf-1-2:~/Downloads/ch2_text_source_code$ lsmod | grep csc381
csc381_mod3            12288  0
asdf@asdf-1-2:~/Downloads/ch2_text_source_code$ date
Mon Dec  4 02:16:31 PM EST 2023
```

The code is similar to the first part, but jiffies must be divided by HZ to get the amount of seconds that have elapsed.



```
rv = sprintf(buffer, "Time since startup: %lu\n", jiffies/HZ);
```

Conclusion

This project taught me a lot about kernel modules. I do not have much experience programming in C, so it was interesting to solve the problems presented. I like the hands-on approach to the project, by interacting with a real kernel, albeit in a virtual machine. There were a few issues I encountered, namely having to use the hello2.c file instead of hello, as well as figuring out how modules work in general, as I did not have much experience prior to this.