

CSCE 3402 Spring 2021

Exercise 4 – Version Kernel Module

Kernel Module to Retrieve Current Running Kernel Version

Assigned: Tuesday, March 2nd in Lab

Due: Wednesday, March 9th at Lab time

Delayed submission with penalty until Thursday, March 11th at 11:55pm.

Goals

The goal of this lab exercise is to extend your simple kernel module you developed in the previous exercise to retrieve the current running kernel version and print it on the screen. It is very essential to do this step to be able to do the next lab exercise.

Details

This lab exercise is an individual exercise that you need to carry out on your own. In this lab exercise, your kernel module upon entry, should retrieve the current running kernel version and print it on the screen. The version of the current running kernel can be retrieved from the **proc file** system, precisely the file “/proc/version”. This file stores the current running kernel version number along with other things, so you should be able to extract the version number from the file. Your main challenge is that how are you going to open a file and read from it from within the kernel. Simply put, you do not have access to the standard **libc** library, and hence you do not have access to wrapper functions such as **fopen**, **fread**, and **fclose** :) moreover you cannot really invoke a system call from within the kernel as your kernel module executes within a trap (SW interrupt), so you cannot really induce an interrupt while handling one :):). You should know why ;-). You need to invoke directly the VFS entry point APIs! What is VFS? It stands for the Virtual File System. **Simply, you are required to do the following tasks:**

1. You need to utilize the **filp_open**, **vfs_read**, and **filp_close** kernel functions to access files and read from them; you need to read the man pages or kernel documentation on how to use these functions.
2. You need to define the following struct in your kernel module and use it:

```
struct myfile{  
    struct file * f;  
    mm_segment_t fs;  
    loff_t pos;  
};
```

3. You need to implement the following 3 functions in your kernel module:
 1. **struct myfile * open_file_for_read (char * filename)**

2. **volatile int read_from_file_until (struct myfile * mf, char * buf, unsigned long vlen, char c)**
3. **void close_file (struct myfile * mf)**
4. You will need to utilize the **kmalloc** and **kfree** functions to allocate and deallocate memory; e.g. allocate an instance of the **myfile** struct when needed.
5. Use the 3 functions defined under point #3 to open the “/proc/version” file, extract the version # from it, and close it.

IMPORTANT Note: From now on please make sure that you boot up with the kernel grub entry you created in the lab exercise #3 that has the KASLR disabled.

What to submit

1. All the C code you wrote for the version kernel module.
2. Your Makefile.
3. A small readme file explaining how to use your make files to compile the programs.

How to submit:

Compress all your work: source code, report, readme file, and any extra information into a zip archive. You should name your archive in the specific format <Student_ID>_<Name>_Lab4.zip. Finally, upload your code to blackboard.

Grade

This Lab exercise is worth 10 % of the overall course grade. The exercise will be graded on a 100% grade scale, and then will be scaled down to the 10% its worth. The grading of the assignment will be broken down as follows:

1. 10 % for just submitting a meaningful assignment before or on the due date. This 10% does not account for the correctness of your assignment but submitting an empty assignment without code will definitely result in losing this 10% and consequently the whole grade of this assignment.
2. 80 % for the correctness and the quality of the submitted code and make files.
3. 10 % readme file.

Delays

You have up to 2 working days of delay, after which the assignment will not be accepted and your grade in that case will be ZERO. For every day (of the 2 allowed days), a penalty of 10% will be deducted from the grade. And of course, you will lose the 10% mentioned in point 1 above under the “Grade” section.