Georgia State University

CSc 4320/6320 Operating Systems

Spring 2020

Project 1

Due Time: 11:59 PM, January 31

Part I

The intention of this part is to practice the programming of basic usage of a Linux kernel module, such as creating, loading, and removing a kernel module.

Requirement:

- Read through Part I—Creating Kernel Modules of the Programming Projects for Chapter 2, which is located at the end of the chapter in the textbook.
- Download *simple.c* and *Makefile_1* from iCollege. Complete Part I Assignment in the textbook.
- Take a screenshot of the kernel log buffer right after loading and removing the *simple* module. (to be reported)

Part II

The intention of this part is to practice the programming of management of linked list in a Linux kernel module.

Requirement

- Read through Part II—Kernel Data Structures.
- ***Ignore*** Part II Assignment
- Download *simple-solution.c* and *Makefile_2* from iCollege.
- Edit the incomplete C source code to create a linked list to store the birthday information of 5 random students. For each person, the birthday information should include month, day, year, and ***name***. When the module is loaded, traverse through the linked list and output its content to the kernel log buffer. In addition, write code to identify the oldest student and remove that student from the list. After removing the oldest student, output the updated linked list content to the kernel log buffer. In the

module exit point, delete the elements from the updated linked list and return the free memory back to the kernel. Make sure to output a message to the kernel log buffer every time an element is deleted.

- After compiling *simple-solution.c*, load and unload *simple-solution* kernel module.
- Take a screenshot of the kernel log buffer right after loading and removing the *simple-solution* module. (to be reported)

Important Notes

- It is strongly encouraged to use a Virtual Machine for your project since code errors may crash your system if directly working on the actual machine.
- The code was tested on Ubuntu 12 and 14. If a newer version does not work, you may try an older version of Ubuntu.
- Use two directories for the two parts when testing your program. Make sure there is no space in the directory name, otherwise problems may happen.
- You need to change the file name of “Makefile_1” and “Makefile_2” to “Makefile” in each directory before executing the *make* command.
- *gcc* is required for compiling the code in *make* command. If Ubuntu distribution does not have *gcc* installed, install *gcc* first.
- This is an individual project. One shall not team up with other students.
- It is strongly encouraged to put comments in your code so that the grader can clearly see your steps. It also makes sure that you will not loss any credit that you should have earned.

Submission

Submission should include a report (in PDF) with all “to be reported” items and the completed *simple-solution.c* file. The *simple-solution.c* file will be compiled and run with the Makefile (the one in iCollege) to test your code. Submit the assignment to iCollege assignment submission folder before the deadline.

Failure to follow the submission requirement will cause 10% deduction in the score.