LE/EECS 3221 – Operating System Fundamentals

Fall 2019

Programming Assignment 1

Submission Deadline: September 22, 2019 before 23:59

Objectives

- Learn how to use Linux system calls
- Learn how to create Linux Kernel Modules
- Learn how to access different OS related information is Linux Kernel Modules

Submission Requirements

Please submit your results/output in the solution template provided.

The questions that involve performing task on the computer, your answer will be the screen shot of your steps; writing the answer in the text form is not acceptable.

The questions that involve coding/script, you have to write your code/script in text form in the document.

The submission deadline is September 22, 2019 before 23:59.

Assignment Requirements and Setup

You are required to perform the tasks in a Linux environment using C/C++ language. Before starting the tasks, first change your command prompt as following. As a result, make sure that you see your Student ID in command prompt every time you write a new Linux command.

In the Terminal, open the **.bashrc** file using some text editor such as gedit.

Case 1: Locate the line starting with **PS1=**. Here, modify the text **\u**; write your Student ID instead.

Case 2: If you do not see a line **PS1=**, then write one as **PS1="your student ID"**

Save and close. Restart the Terminal (or reload the bash) to see the effect of this change.

In the labs at YorkU you might face the TurboC shell. In that shell, the settings are stored in **.cshrc** or **.tcshrc** file. There you need to add/modify as following:

set prompt= 'your student ID'

And there are some other shells too. Your installation might be running a different shell. Just find the way to change the prompt in that shell.

Question 1:

Write a program that copies the contents of one file to a destination file. This program works by first prompting the user for the name of the source and destination files. Be sure to include all necessary error checking, including ensuring that the source file exists. Also, if available, you have to use Linux system calls and not standard C library functions in your program.

- **#1-A [20 Marks]:** Write/type your source code in your submission file/document.

  [**Restrictions**] F**or file handling operations, only use the open, close, read and write system calls**. For other operations you can use the library functions. **For every use of library function where system call was required, 2 marks will be deducted**.

  Following error situations must be taken care of in the code. 1 mark per missing case will be deducted: File not present, read error, write error, close error, create error, return value of system calls.

- **#1-B [5 Marks]:** In your submission file/document, add a screenshot which displays:
  - the content of destination directory before the execution of your program
  - execution of your program
  - the content of destination directory after the execution of your program

  [**Restrictions**] The screen shot(s) attached must contain all the three steps above. Two marks per missing case will be deducted.

Question 2 [5 Marks]:

Repeat the execution of your program in Question 1 using "strace" utility. Redirect the output to a file and attach that file with your submission.

[**Restrictions**] All 5 marks will be deducted if this action is not performed and documented properly.

Question 3:

Read the "Programming Projects-Kernel Modules Overview" section at the end of Chapter 2. Download the source file "simple.c" from the book website and understand the code in this file. Perform the following steps in section "Loading and Removing Kernel Modules". Add a screenshot of this step.

- **[5x4=20 Marks]:** Use the "insmod", "lsmod", "dmesg" and "rmmod" with your program (the executable version of simple.c). Add screen shot.

  [**Restrictions**] The screen shot(s) attached must contain all the four steps/commands above. Five marks per missing case will be deducted. If the command is tried/used, but the outcome is not as expected, three marks will be deducted.

  The outcome of "lsmod" is usually quite long and it is hard to trace your own module. Use a suitable command line technique to display the row for your own module only. 2 Marks will be deducted if this action is not performed.

Question 4 [20 Marks]:

At the end of Chapter 3, read the section, "Project III-Linux Kernel Module for Listing Tasks". Design the kernel module as required in Part I i.e. a kernel module that can iterate over all existing tasks the way it is described in that section.

- Additionally, this module should display the PIDs of all the processes. Submit your code in the document.

[**Restrictions**] The code should follow the kernel module structure as mentioned in Question3. Provide the code in a separate C/CPP file.

- Load and execute your module and add the screen shot of this action in the document.

[**Restrictions**] Redirect the output of execute step to a separate file and attach that file to your submission. 5 marks will be deducted for each missing step. 5 marks will be deducted if the PIDs are not shown.