COMP-8567
Assignment 03
Winter 2025
Due Date: Mar/20/2025, 11 PM EDT
50 Marks

Associated Learning Outcomes:

- Apply OS concepts to design algorithms to solve systems programming problems in a variety of different systems, such as Unix/Linux/Android environments.
- Correctly define systems programming problems and identify and apply appropriate solutions approaches.
- Design and implement solutions that use the hardware and/or kernel services to solve systems programming problems involving the latest computing technologies.

Note: Please check the following link for the **complete list** of learning outcomes for COMP 8567
https://ctl2.uwindsor.ca/cuma/public/courses/pdf/ee1b450a-23a6-4635-b0c6-40a47a21331f

Please read the instructions carefully:

1. This assignment **must be implemented** on our CS Linux sever using your official university login.
2. Since this assignment might involve the creating a number of processes to test your program, you might inadvertently create a **chain of processes** that might lead to what is known as a **"fork bomb"** that uses up a lot of system resources.
3. Regardless, it is It is your sole responsibility to execute the statement: **$killall -u *username*** periodically/mandatorily **when you are done working on this assignment on a given day** failing which zero marks will be given to the assignment if there are any complaints from the system administrator

Write a C program **w25shell** (**w25shell$**) that goes into an infinite loop waiting for user's commands. Once a command is entered, the program should **assemble** and execute each command using fork(), exec() and other system calls as required with the following rules and conditions.

NOTE: You **cannot** use the **system()** library function

A modular approach would be helpful.

Rule 1:

- The command **killterm** (**w25shell$**killterm) must kill the current **w25shell** terminal

Rule 2:

- The command **killallterms** (**w25shell$**killallterms) must kill all the **w25shell** terminals that are currently open

Rule 3: The argc (includes the name of the executable/command) of any command/ program should be >=1 and < =5

Examples:

- **w25shell$ pwd (argc =1)**
- **w25shell$ grep to sample.txt (argc=3) s**
- **w25shell$ ls -l -t -a (argc =4)**

Rule 4: The argc of induvial commands or programs that are used along with the special characters listed below should be >=1 and <=5

- Ex: **w25shell$ ls -l -t -a |wc -w** //the first command has argc=4 and the second command has argc=2 which are used along with the special | character

Special Characters

The program should handle the following special characters (In accordance to Rule 3 and the additional rules listed below)

- | **Piping** (up to 5 piping operations should be supported)

  Ex **w25shell$ ls -l -t -a |grep *.txt|wc| wc -w**

  // Every command/program can have argc >=1 and <=5 as per Rule 3

- = **Reverse Piping** (up to 5 reverse piping operations should be supported)
  w25shell$ wc -w = wc = ls -1 should transfer the output of ls -1 -> wc-> wc -w -> stdout
  // Every command/program can have argc >=1 and <=5 as per Rule 3

- ~ Append text to both text files (Binary operation-requires two .txt files)

  w25shell$ sample1.txt ~ sample2.txt should append the contents of sample2.txt to sample1.txt and append the contents of sample1.txt to sample2.txt (and save both the files)

- # Counts the number of words present in a particular .txt file
  w25shell$ # sample.txt //should output the number of words in sample.txt

- + Text file (.txt) concatenation (up to 5 operations)
  Ex **w25shell$ sample1.txt + sample2.txt + temp1.txt + temp2.txt**
  // Files must be concatenated in the order in which they are listed and the final result is sent to stdout

- <,>, >> Input Redirection from a text file, Output Redirection to a text file, Output redirection to a text file (with append output)

  w25shell$ grep Windsor <sample.txt

  w25shell$ ls -1 >dirlist.txt

  w25shell$ ls -1 >>dirlist.txt

- ; **Sequential execution** of commands (up to 4 commands) the argc of each command should be >=1 and <=5 as per Rule 3
  Ex: w25shell$ date ;pwd ;ls -l – t -a

- && **Conditional Execution** // up to 5 conditional execution operators should

  be supported and could possibly be a combination of && and ||

  Ex : w25shell$ ex1 && ex2 && ex3 && ex4

  - **w25shell$ c1 && c2 || c3 && c4**

- || **Conditional Execution** // see &&
  - Note in both && and ||, the argc of each command should be >=1 and <=5 as per Rule 3

You are NOT required to combine special characters : ex $ p1 & p2 > list.txt

Note:

- **You must include comments throughout the program reasonably explaining the working of the code.**
- You have to use **fork() and exec() along with other pertinent system calls** to run commands from w25shell //You cannot use the system() function
- Appropriate **error messages** must be displayed by the program based on the specifications.

Submission Instructions:

Plagiarism Detection Tool: MOSS

You need to submit the following:
1. w25shell_fname_lname_SID.c
2. Zoom/Google Drive recording **(with your camera switched on)** link explaining the following (not more than 15 minutes)

- Overall working of the code and various modules (around 8-9 minutes)
- Execution of the code under various inputs/conditions as per the requirements of the assignment (around 6-7 minutes)
- Other form of links/MP4 files will NOT be acceptable.
- Include the link in the COMMENTS section.