

System Programming - Linux

Week 15 - Review of Course Project(Questions)

Lecturer: Dr. Aggrey Obbo (PhD)

June 16, 2025

Contents

1 Section 1: Linux Utilities and Installation	1
2 Section 2: File Systems and Directories	2
3 Section 3: Inter-Process Communication and Redirection	2
4 Section 4: Message Queues	3
5 Section 5: Semaphores	3
6 Section 6: Sockets	3

This exercise is designed to assess the student's practical and theoretical understanding of fundamental Linux system programming concepts. The student is required if possible to use a Linux virtual machine or environment such as Ubuntu to answer the practical questions.

Instructions

1. Complete all sections. For practical questions, execute the commands and briefly describe the outcome or relevant file changes.
2. For theoretical questions, provide clear and concise explanations.
3. Assume you are working in a fresh user directory.
4. The highest obtainable mark is 100

1 Section 1: Linux Utilities and Installation

- (a) Install the htop utility (a powerful interactive process viewer) on your Linux system.
 - (i) Provide the command(s) you would use.
 - (ii) Briefly explain what htop shows that ps doesn't by default.(2.5 Marks for each right answer)
- (b) Display your system's total RAM, used RAM, and free RAM in a human-readable format.
 - (i) Provide the command.
 - (ii) Capture a snippet of its output.(2.5 Marks for each right answer)

- (c) Create a new user named devuser with a home directory and a default shell. Set a password for devuser (e.g., "DevP@pass").
 - (i) Provide the command(s).
 - (ii) Explain how you would verify the user was created.(2.5 Marks for each right answer)

2 Section 2: File Systems and Directories

- (a) Create a directory structure as follows:
 - (i) `~/project_work/module1/src`
 - (ii) `~/project_work/module1/doc`
 - (iii) `~/project_work/module2/src`
 - (iv) `~/project_work/module2/bin`
 - (v) `~/project_work/module3/bin`1 Mark for each right answer
- (b) Provide the single command to achieve this. Inside `~/project_work/module1/src/`, create an empty file named `main.c`. Then, change its permissions so that only the owner can read and write, while group and others have read-only access.
 - (i) Provide the command(s).
 - (ii) Verify the permissions using `ls -l`.(2.5 marks for each right answer)
- (c) Change the ownership of the `~/project_work/module2/bin/` directory and all its contents to devuser (from Section 1) and its primary group.
 - (i) Provide the command(s)
 - (ii) Explain how you would verify the ownership change.(2.5 marks for every true answer)
- (d) (5 Marks - Practical) From your home directory (`~`), create a symbolic link named `my_project` that points to `~/project_work/module1/src/`.
 - (i) Provide the command(s).
 - (ii) Provide the command. Demonstrate that `my_project` behaves like a pointer by trying to `cd` into it.(2.5 Marks for each right answer)

3 Section 3: Inter-Process Communication and Redirection

- (a) Use a pipe (`|`) to count the number of lines in the `/etc/passwd` file that contain the word "bash".
 - (i) Provide the command.
 - (ii) State the output you get.(2.5 marks for every right answer)
- (b) Redirect the standard output of the `ls -l` command to a file named `file_list.txt` in your current directory. Then, append the output of `date` to the same `file_list.txt`.
 - (i) Provide the command(s).
 - (ii) Show the content of `file_list.txt` after both operations.

(2.5 marks for every right answer)

- (c) Execute a command that will intentionally produce an error (e.g., `cat /nonexistent_file`), but redirect its standard error to a file named `error_log.txt`. Ensure no error message appears on your terminal.
 - (i) Provide the command.
 - (ii) Show the content of `error_log.txt`.

(2.5 marks for every right answer)

4 Section 4: Message Queues

- (a) Describe a scenario where using a message queue would be a suitable IPC mechanism. Explain why it's suitable for that scenario compared to shared memory or pipes.
(5 Marks)
- (b) Briefly explain the purpose of the following key POSIX message queue functions: `mq_open()`, `mq_send()`, and `mq_receive()`.
(5 Marks)

5 Section 5: Semaphores

- (a) Explain the primary problem that semaphores are designed to solve in concurrent programming.(5 Marks - Theoretical) Provide a simple analogy.
(10 Marks)
- (b) Describe the behavior of the two fundamental semaphore operations, often referred to as "P" (or `wait()`) and "V" (or `post()`). How do these operations ensure synchronization?
(10 Marks)

6 Section 6: Sockets

- (a) Briefly compare and contrast TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) in terms of reliability and connection state.
(5 Marks)
- (b) Outline the logical sequence of operations (system calls) a TCP client typically performs to establish a connection and send data to a server.
(5 Marks)
- (c) Outline the logical sequence of operations (system calls) a TCP server typically performs to listen for, accept, and handle a single client connection.
(5 Marks)
- (d) You are trying to connect to a web server (e.g., `example.com` on port 80), but your client program consistently receives a "Connection refused" error. List two common reasons for this error and suggest a Linux command-line utility you could use to diagnose each reason.
(5 Marks)

End.