UNIVERSITY OF RUHUNA

Faculty of Engineering

End-Semester 6 Examination in Engineering: January 2022

Module Number: EE6206

Module Name: Operating Systems and Programming (N/C)
Part II (Essay)

[Two Hours and Twenty Minutes]

[Answer **all questions**, each question carries 10 marks. Mention all the necessary header Files in each of the program you write.]

(Q1)

a) Write Linux console-based commands for the following sequence of operations?

(i) Go to home directory. Change the directory to **Desktop**. Display the current working directory.

(ii) Assuming there is a text file known as **txt1.text**; create an exact copy of **txt1.text** as **txt2.text** in the previous (home) directory.

(iii) Go back to the **home** directory and copy the **Desktop** directory to inside the **Documents** directory as **Desktop_copy**. Note that **Documents** is a directory inside the home directory.

(iv) Delete **Desktop_copy**, go to the **Home** directory and list the details of **txt2.text** file in long format.

(v) Remove the read privilege of the user for **txt2.text** and add write privilege to users in File's group.

(vi) Create a directory named **mydir2** in **Desktop**. Go to that directory. After that create two empty text files as **Doc1.txt, Doc2.txt** inside **mydir2** using,

(I) **Touch** command

(II) **Cat** command

(vii) Assuming both **Doc1.txt** and **Doc2.txt** has content inside them, copy the content of **Doc1.txt** to **Doc2.txt** and display **Doc2.txt**.

(viii) Observe the statistics of **Doc1.txt.** Change the access time of **Doc1.txt.** Change the modification time of **Doc1.txt**

(ix) Assume there is a C program file called my_program.C. Write instructions to compile and run an object file with same name as my_program.C

[4 marks]

b) Write a code using C programming language for the following.
Create a **line** structure by nesting two (x, y) **point** structures. Get input from user for the 2 points and calculate and display its length using the knowledge of pointers to structures and functions.

[3 marks]

c) Write a code using C programming language for the following.
Using the High-level I/O model only, create and open a file **doc1.txt** with read and write privileges. Then, write text *"Hello, this is OS programming."* to **doc1.txt**. Then, append the content of **doc1.txt** to the end of a file named **doc2.txt**. After that, change the privilege of **doc1.txt** to write only. At last, the program must try to read from **doc1.txt** with error handling.

[3 marks]

(Q2)

a)

(i) State two (2) differences and two (2) similarities between structures and unions.

[1 mark]

(ii) Briefly explain the meaning of **'pass by reference'** for functions using a simple example.

[1 mark]

b)

(i) State two (2) differences and two (2) similarities between pipes and message queues.

[1 mark]

(ii) State and briefly explain different ways to create unique keys for generating unique Inter Process Communication (IPC) objects.

[1 mark]

c)

(i) Write a complete C program to implement the following scenario.

Use Inter process communication using pipes to implement the following scenario for 3 processes **P1, C1, C2** where **P1** is the initial process, **C1, C2** are the children of P1

The root parent **P1** must send the message **"Hello my child"** to each of its children and waits for response. When any child process **(C)** receives the message **"Hello my child"** from its parent process **(P)** and if that received child process **(C)** does not have any children it must immediately respond to the parent process **(P)** as **"Hello mum"** and terminate its process. If that received child process **(C)** has children; then it must send **"Hello my child"** to each of its children and wait until each of the children respond with **"Hello mum"**. After each of children has responded with **"Hello mum"** to C; then only **C** must respond to its parent **(P)** as **"Hello mum"** and terminate its process **(C)**. When each process receives a message; it should print a statement displaying its name (C1, P1 etc.), its process ID, the sender's process ID and the particular message received to standard output. The program must include error handling as well.

[3.0 marks]

(ii) Write a complete C program to implement the following scenario.

An initial process **(P1)** creates a child process **C1**. **C1** creates its child process **C2**. Create a message queue and send the following messages to the queue from the initial process **(P1)**. **P1** must send the messages to the queue in the exact order given in Table Q2 starting from **"MESSAGE 1"** and wait until its child process **(C1)** terminate.

TABLE Q2: Table of messages in the message queue

Type	Text
25	MESSAGE 1
50	MESSAGE 2
75	MESSAGE 3
50	MESSAGE 4
35	MESSAGE 5
110	MESSAGE 6

- Suspend the execution of **C1** for 1 second after **P1** creates the queue. After that 1s, **C1** must read and print first 2 messages of the queue in FIFO manner and print to terminal along with the process ID. After that, **C1** will wait until **C2** terminates. Once **C2** terminates, **C1** will also terminate.
- Suspend the execution of **C2** for 3 seconds after **P1** creates the queue and then read next 2 messages in ascending order of type of messages. Then, **C2** must terminate after printing the 2 messages it read.
- After the child process **C1** terminates, the initial process **(P1)** must read the message with type 110 and it should terminate after printing the read message with process ID.

What is the message remaining in the queue at last?

[3.0 marks]

(Q3)

a)

(i) State the corresponding function used for sending a set of bytes from a memory buffer to a file in each of Low level I/O and High level I/O models. Mention two (2) differences and two (2) similarities between those two functions used in two types of models. You may discuss about the intended task, number and type of arguments, return types, header files etc. for comparison.

[1 mark]

(ii) List two (2) different types of permission identified in file systems.

[0.5 mark]

(iii) A programmer has passed **0666** as the third argument of *open( )* function. Write an alternative way to pass this argument to *open( )* function with the required header file.

[1 mark]

b)

(i) Define what is a thread and a process separately. Specify two (2) differences and two (2) similarities between threads and processes.

[1.5 mark]

(ii) State 3 ways that can be used for Inter process communication and 2 techniques of thread synchronization.

[1 mark]

c)

(i) Write a complete C program to implement the following scenario.

A parent process **P1** will create two child processes **(C1, C2)** simultaneously. Create a shared memory segment storing a long value initialized to zero. **C1** must run a function which increments the value of the shared memory segment by one (1) 10 million times whereas **C2** increments it by two (2) 10 million times. At the same time, the parent process **(P1)** must decrement it by three (3) 10 million times. At last, the parent process must return the value of shared memory segment. Use concurrency control using Semaphores to provide the correct result.

[2.5 marks]

(ii) Write a complete C program to implement the following scenario.

Create a hyper-threaded process with two (2) simultaneous threads. One thread should write as "**T1**" repeatedly forty (40) times and the other thread should write as "**T2**" repeatedly forty (40) times. Thread synchronization should be as follows. When **Thread1** executes, it should print the word "**T1**" ten (10) times and after that wait until the other thread prints "**T2**" five (5) times before **Thread1** prints again. After **Thread1** has finished printing; **Thread2** can print continuously. Use Mutex and condition variables for this program.

[2.5 marks]

(Q4)

a)

(i) Define what is a socket in IPC and describe the *connect( )* function.

[1 mark]

(ii) List down the 7 attributes of a socket.

[1 mark]

(iii) State and briefly explain the two types of byte orders.

[0.5 mark]

(iv) Briefly explain the meaning of the following two commands,

(I) cat /etc/hosts

(II) netstat -ant | grep 22

[1 mark]

(v) State and briefly explain the server only functions in the correct sequence which they should be called in a server machine in client-server communication.

[1.5 mark]

b)

(i) Write a complete C program to implement the following scenario.

Create a client program to connect to the **echo** service of a remote machine to send a user specified string of user specified size and display the returned string on screen. Include error handling for sending only. The dotted decimal notation of the IPv4 address of the remote machine is **148.158.56.521**. The socket's domain is internet and its service type is stream. After receiving the reply from the remote machine, the client program must make sure that the socket is disabled for reading only for any process which has been connected to it.

[2.5 marks]

(ii) Write a complete C program to implement the following scenario.
Create a "simultaneous repetitive calculation" program as follows using C programming language. Use 4 hyper threads (**T1, T2, T3, T4**) each returning values for Addition, Subtraction, Multiplication, Division respectively of user input two floating point numbers to the main thread when each of them terminates causing the main thread to print each returned value along with thread ID. Each thread T1, T2, T3, T4 must respectively print strings **"Addition ", "Subtraction ", "Multiplication", "Division"** concurrently user specified number of times. Use one thread function for all 4 threads where each thread passes different arguments to the thread function. There are no concurrency control measures employed. Assume each thread consumes a constant time of 1 ms to print its string "Addition", "Subtraction" etc.

[2.5 marks]