



# UNIVERSITY OF RUHUNA

## Faculty of Engineering

End-Semester 6 Examination in Engineering: November 2022

Module Number: EE6206

Module Name: Operating Systems Programming

[Three Hours]

[Answer all questions, each question carries 10 Marks]

Q1 a)

- (i) Write three (03) methods to create an empty file named 'doc.txt' using Linux terminal. [0.5 Marks]
- (ii) Write a Linux command to remove write permission for group and others for a file 'a.txt'. [0.5 Marks]
- (iii) What is the symbolic representation of numeric permission 644? [0.5 Marks]
- (iv) Write all the steps that need to follow in order to add an alias named 'shwh' to 'bashrc' for the Linux command to show all-hidden directory contents. [1.0 Mark]
- (v) Write a Linux command to append the content of 'doc1.txt' to an existing 'doc2.txt'? [0.5 Marks]
- (vi) Write a Linux command to compress a directory named '/home/myfolder' by using tar command. [0.5 Marks]
- (vii) What is the Linux command that can be used to search patterns of words in a file? Briefly explain using an example. [1.0 Mark]
- (viii) What is the meaning of permission string 'rw-rw-rw-'? [0.5 Marks]
- (ix) Write four (04) operations performed by the Linux kernel. [1.0 Mark]

b)

- (i) Write a complete C program to find the factorial of an user input number by using a recursion function. (Ex: If user input 4, factorial 4 or  $4! = 24$ ) [3.0 Marks]
- (ii) What is the output of the C code shown in Figure Q1? Explain your answer briefly. [1.0 Mark]

Q2 a)

- (i) Write a statement which can be used to open a file using low level I/O system calls in C programming language. Explain the return value and the

parameters of that system call.

[1.0 Mark]

(ii) Explain an advantage of explicitly closing a file descriptor.

[1.0 Mark]

(iii) Create a 10MB file named 'dataFile' using Linux 'dd' command.

[0.5 Marks]

(iv) Write a complete C program by using low level I/O to create a text file named 'doc1.txt' and write name and age of the user using standard input to 'doc1.txt' with appropriate error handling. If 'doc1.txt' exists; the program should truncate it and rewrite.

[2.0 Marks]

(v) Write a complete C program by using low level I/O to copy the file created in Q2 a) (iii) to a new file named 'copiedFile'.

[2.0 Marks]

b)

(i) Briefly explain the 'fscanf' buffered input system call in C programming language.

[1.0 Mark]

(ii) Write a complete C program with appropriate error handling to read and print the content of 'doc1.txt' (i.e. created in Q2 a) (iv)) to the console using high level file I/O. (Note: You should use fscanf buffered input system call to read the data from the file)

[2.5 Marks]

Q3 a)

(i) Define the meaning of virtual memory?

[1.0 Mark]

(ii) Write three (03) advantages of virtual memory.

[1.0 Mark]

(iii) State three (03) cases where CPU exception page fault occurs.

[1.0 Mark]

b)

(i) Briefly explain the term Hyper Threading.

[0.5 Marks]

(ii) Briefly explain the pthread\_create() function in C programming language with respect to the return value and parameters.

[1.5 Marks]

(iii) Explain the use of condition variables in multi-threaded communication with respect to pthread\_cond\_wait() and pthread\_cond\_signal() functions.

[1.5 Marks]

(iv) Write a C program with appropriate error handling to create two threads. One thread runs a function which increments a process scope global variable by one, 10 million times using a loop and the other thread must decrement it by one, 10 million times using a loop. At last, main thread must return the value of global variable.

[2.5 Marks]

(v) Modify the program written in Q3 b) (iv) with concurrency control using Mutex to provide the correct result.

[1.0 Mark]

Q4 a)

- (i) Briefly explain the fork system call in C programming language. [1.0 Mark]
- (ii) Explain a method to use pipe inter process communication method to make a full duplex communication channel between processes. [1.0 Mark]
- (iii) Write a C program with appropriate error handling to create a pipe to transfer the message "This message is from parent!" from parent process to child process and display using terminal in the child process. The parent should wait until the child terminates, before printing "My child has terminated". [2.5 Marks]

b)

- (i) Compare message queue and pipe inter process communications. [1.0 Mark]
- (ii) Write the typical message structure which is used in message queue inter process communication. [0.5 Marks]
- (iii) What are the two (02) constraints related to the structure of the messages in message queue inter process communication. [1.5 Mark]
- (iv) Write a C program with appropriate error handling to create a message queue to communicate between the parent and the child process. The parent process should send the messages in Table Q4 to the message queue and wait until the child terminates. The child process should only read the message with type 200 and print the content to the console. [2.5 Marks]

Q5 a)

- (i) Explain why Shared memory is considered as the fastest among the inter process communication methods. [0.5 Marks]
- (ii) What are the two (02) basic steps that need to be performed in order to use shared memory inter process communication? [0.5 Marks]
- (iii) Write a C program with appropriate error handling to create a shared memory segment to send "Hello Child" message from parent to the child process. Then the child process should print it and send back "Hello Parent" message to the parent. The parent should wait until the child terminates, before printing the child's message. [2.0 Marks]

b)

- (i) Briefly explain the uses of Semaphores. [0.5 Marks]
- (ii) Briefly explain the functions sem\_wait() and sem\_post() used in C



programming language.

[1.0 Mark]

c)

(i) What are the two (02) socket domains?

[0.5 Marks]

(ii) Write seven (07) attributes of a socket.

[0.5 Marks]

(iii) Draw server-client socket workflow.

[0.5 Marks]

(iv) Write a C program with appropriate error handling to create client and server socket communication. The server should listen until the client is connected to the server and then the server should send the message "This msg is from server!" to the connected client. Then the client should print it to the console and reply with "Hi Server!" to the server. Then the server should print the message received from the client to the console.

[4.0 Marks]

```
#include <stdio.h>
void solve() {
    int i = 4;
    int res = i++ + ++i + i++ + ++i;
    printf("%d", res);
}
int main(){
    solve();
    return 0;
}
```

Figure Q1: Sample C code

Table Q4: Type and Text of the Message Queue

Type	Text
100	I am hungry
200	I am sleepy
300	I am angry
400	I am lonely