UNIVERSITY OF RUHUNA

Faculty of Engineering

End-Semester 6 Examination in Engineering: April 2025

Module Number: EE6253 **Module Name: Operating Systems and Network Programming**

[Three Hours]

[Answer **all questions**, each question carries **10 marks**]

Part II: Essay Questions

Q2 In a multi-programmed system, effective resource management is essential to avoid deadlock situations and ensure the system remains in a safe state. The Banker's Algorithm, proposed by Edsger Dijkstra, is a deadlock avoidance algorithm used to determine whether resource allocation leaves the system in a safe state.

a) Explain the data structures used in Banker's algorithm.

[2.0 Marks]

b) Describe a real-world deadlock problem.

[1.0 Mark]

c) Explain the 4 conditions for the deadlock.

[1.0 Mark]

d) Consider a system with 5 processes (P_0 to P_4) and 3 resource types (A, B, and C). The total number of instances for each resource type is as follows:

- Resource A: 10 instances
- Resource B: 5 instances
- Resource C: 7 instances

The number of available instances of each resource type is [3, 3, 2].
At a specific moment T_0, the system is in the state, as represented in Table Q2-d).

Table Q2 d)

Process	Allocation [A B C]	Maximum [A B C]
P_0	0 1 0	7 5 3
P_1	2 0 0	3 2 2
P_2	3 0 2	9 0 2
P_3	2 1 1	2 2 2
P_4	0 0 2	4 3 3

i) Apply the Banker's algorithm and find out whether the above system is in a safe state or not.

[5.0 Marks]

ii) To keep the system in a safe state, find out the order of the processes to be executed.

[1.0 Mark]

Q3 CPU scheduling is a fundamental aspect of operating system design that determines the order in which processes access the central processing unit. Efficient scheduling algorithms are crucial for maximizing CPU utilization, ensuring fairness and enhancing overall system performance.

a) List four key circumstances under which CPU scheduling decisions are made in a typical operating system.

[1.0 Mark]

b) Describe the following terms in CPU scheduling.

[2.0 Marks]

i) Non-Preemptive
ii) Convoy Effect
iii) Dispatcher
iv) Throughput

c) In a time-sharing system, which CPU scheduling algorithm would be most appropriate and why?

[1.0 Mark]

d) Suppose there is a laundry shop which provides ironing services to its customers. Consider the data in Table Q3-d).

Table Q3 d)

Job	Arrival Time	Execution Time
Ironing a school uniform	8.30 am	8 minutes
Ironing a trouser	8.31 am	4 minutes
Ironing a silk saree	8.32 am	9 minutes
Ironing a frock	8.33 am	5 minutes

i) Find the average waiting time of each job when the shortest job is first carried out.

[2.0 Marks]

ii) Find the average waiting time of each job when shortest remaining time job is first carried out.

[2.0 Marks]

iii) Draw Gantt charts separately for above scenarios.

[1.0 Mark]

iv) Discuss the performance of both the above scheduling methods.

[1.0 Mark]

Q4 a) Mastering basic Linux terminal commands is essential for users seeking to harness the full potential of the operating system, as these commands provide the foundation for performing a wide range of tasks from simple file operations to complex system configurations.

i) What is the purpose of the command **mkdir folder1**? What happens if a directory named **folder1** already exists?

[0.5 Marks]

ii) Explain the difference between **mv folder1 ~/user/** and **cp -r folder1 ~/user/**.

[0.5 Marks]

iii) What does **chmod u-w,g+x ~/Desktop/txt2.txt** do? How would you reverse these changes?

[0.5 Marks]

iv) Why might a user be unable to edit **txt2.txt** after running **chmod u-w txt2.txt**?

[0.5 Marks]

v) What is the purpose of **stat Doc1.txt**? List two pieces of information it displays.

[0.5 Marks]

vi) What is the difference between the following codes.

- **cat Doc1.txt > Doc2.txt**
- **cat Doc1.txt >> Doc2.txt**

[0.5 Marks]

vii) Write a command to display the content of both **Doc1.txt** and **Doc2.txt** in the terminal.

[0.5 Marks]

viii) What is the purpose of **sudo init** 0 and **sudo init 6**?

[0.5 Marks]

ix) What is the difference between the following commands?

- **ls ~**
- **ls -l ~**

[0.5 Marks]

x) What is the purpose of **ls -la** ~ command?

[0.5 Marks]

b) Low-level system calls provide a direct interface between user applications and the operating system kernel, enabling precise control over hardware and system resources.

i) Write the code line for following Low Level System Calls and define all the arguments in each system call properly.

- open()
- close()
- write()
- perror()

[2.0 Marks]

ii) Write a program to perform the following task using Low Level File I/O System Calls.

Create a text file named as **doc1.txt** and write name and age the user input using standard input to **doc1.txt** with possible error handling. If **doc1.txt** exists, the program should truncate it and rewrite.

[3.0 Marks]

Q5 a) Explain the life cycle of a Thread using a diagram.

[1.5 Marks]

b) Write down the Java methods used to implement the different states in the thread life cycle.

[1.5 Marks]

c) When can a thread reach the terminated/end state?

[1.0 Mark]

d) State two advantages of implementing the Runnable interface over extending the Thread class to create a thread object in Java.

[2.0 Marks]

e) Write down the steps and corresponding Java snippets involved in establishing a TCP connection between a client and a server using socket programming. Assume that the client sends a message, and the server reads it.

[3.5 Marks]

f) How is a UDP socket created in Java?

[0.5 Marks]