National University of Computer and Emerging Sciences

Chiniot-Faisalabad Campus

Operating System (CS206)

Sessional-I Exam

Date: November 5th 2024

Total Time (Hrs): 20 min

Total Marks: 15

Total Questions: 2

Course Instructor(s)

Ms. Juhinah Batool, Ms. Rukshana Zafar, Ms. Ayesha Liaqat, Ms. Mahzaib Younas, Mr. Tahir Farooq, Mr. Haseeb Arshad

_____ _____ _____

Roll No Section Student Signature

Vetted by: _____ **Signature:**_____

Do not write below this line

Attempt all the questions.

CLO 2: Implement solutions employing concepts of Processes and Threads

Q1: Choose the correct one [10 Marks]

Rubrics = 1 Marks for each, cutting not allowed

1.	**Suppose that a process is in "Blocked" state waiting for some I/O service. When the service is completed, it goes to the :**
	a. Running state **b. Ready state** c. Suspended state d. Terminated state
2.	**A process is a program in execution, it exists in main memory and it may be:**
	a. Either OS process or User process b. Either I/O bound process or CPU bound process c. Either Independent process or Cooperating process **d. All the above**
3.	**A scheduling method in which a processes can be interrupted whether they have completed their current task or not is**
	a. Dynamic **b. Preemptive** c. Static d. Non-Preemptive
4.	**A Foreground processes is:**
	a. Detached from the terminal it was started b. Like a background process runs without user interaction **c. Able to receive input and return output from/to the user** d. All of the above
5.	**When a process creates a new process using the fork() operation, which of the following state is shared between the parent process and the child process?**
	a. Stack

	b. Heap **c. Shared Memory segments** d. All of the above
6.	**Which of the following statements about threads is/are TRUE?**
	a. Threads belonging to a process are by default not protected from each other b. All the threads belonging to a process share a common stack c. Each thread has its own file descriptor table for open files d. Threads can only be implemented in kernel space
7.	**In what type of system is the round robin scheduling algorithm particularly effective?**
	a. Time sharing system b. Real-time system c. Batch Processing system d. Embedded System
8.	**Which formula is using to identify the next CPU burst**
	a. $S_{n+1} = (1-\alpha)\ Sn + \alpha\ Tn$ b. $S_{n+1} = (1+\alpha)\ Sn + \alpha\ Tn$ c. $S_{n+1} = (1-\alpha)\ Sn - \alpha\ Tn$ d. $S_{n+1} = (1+\alpha)\ Sn + \alpha\ Tn$
9.	**Which of the following statements are true?** **I. Shortest remaining time first scheduling may cause starvation** **II. Preemptive scheduling may cause starvation** **III. Round robin is better than FCFS in terms of response time**
	a. I only b. I and III only c. II and III only **d. I, II and III**
10.	**Push migration and pull migration are two types of _____ approach**
	a. **Load balancing** b. Processor Affinity c. Symmetric Multithreading d. Synchronous Multithreading

Question No 02: Select the True or False option also justify your answer. [5 Marks]

[1 Mark for each 0.5 for correct option and 0.5 for explanation]

1. Independent process is the process that can affect by the other processes. [True/**False**]

   Because Independent processes does not share any data like temporary or persistent with any other process

2. Inter process Communication allows process to synchronize the activity. [**True**/False]

   IPC allows processes to synchronize activity and enables communication between the processes.

3. Messages sent by a process can be fixed or variable sized. [**True**/False]

   Messages sent by a process can be fixed or variable size. If the message size of the process is fixed then system level implementation is straightforward but it makes the task of programming more difficult.

4. The link between two processes P and Q to send and receive messages is called message-passing link [True/**False**]

The link between two processes P and Q to send and receive messages is called communication link. Two processes P and Q want to communicate with each other

5. Software threads operate at the CPU level, providing parallel execution capabilities within a single core

[True/**False**]

Because Software threads exist at the operating system or application level and are managed by the system's thread scheduler