

# Operating Systems (CS2006)

Date: February 28<sup>th</sup> 2024

## Course Instructor(s)

Dr. Nadeem Kafi, Dr. Nausheen Shoaib, Mr.  
Shahbaz Siddiqui, Ms. Anaum Hamid, Ms. Safia  
Baloch, Ms. Mubashra Fayyaz, Ms. Atiya Jokhio,  
Mr. Minhaz Raza, Ms. Saeeda, Ms. Hania Shah

## Sessional-I Exam

Total Time: 1 Hours

Total Marks: 15

Total Questions: 02

Semester: SP-2024

Campus: Karachi

Dept: Computer Science

Student Name

Roll No

Section

Student Signature

**CLO # 1: Describe, discuss, and analyze, services provided by the modern Operating Systems.**

**Q1.** [1.5 marks x 5 = 7.5 marks]

Write short textual answers. Drawings are not allowed.

- Discuss why dual mode of operation is utilized in a multitasking operating system.
- Consider monolithic and microkernel operating system structures. **Which one executes a system call that uses file system and device drivers faster than the other and why?**
- Suppose a process P0 is executing. **What will happen if the next instruction is: i) a read () system call, ii) an interrupt?** Write concise steps for both. Avoid missing key steps or writing them out of order.

Give only labelled diagrams (Use pencil only). Note: textual answers will not be graded.

- For two executing processes, **illustrate how operating system will stop execution of process P0 and allow P1 to execute.** Show all states these processes would take at different times in your diagram. Use technical terms in labels and specify all steps to get full scores.
- Illustrate different (four or more) ways a **process can wait in different queues** after leaving the CPU till admitted back in the ready queue. Assume a multitasking operating system.

**CLO # 2: Understand, design, and implement solutions employing concepts of Processes/Threads.**

**Q2.** [1 marks + 2.5 marks + 4 marks = 7.5 marks]

### Understanding and Design

- Suppose you need to implement an integer circular buffer of size 50 for a producer-consumer scenario. Write **one major reason to reject named pipes** and use shared memory instead. [1]
- Suppose a process executes **fork () system call 2 times**. Draw a labelled diagram with meaningful hints to reflect your understanding of all the resultant processes along with their linkage. Explain which processes should issue wait () system for correct behavior. No grading without appropriate hints to show your understanding. [1.5 + 1]

### Implementation (Use pencil only)

- Write c language snippet that runs both i) **ls -al** command and ii) an executable file **/bin/foo** as two concurrent processes. Use a minimum number of processes. Error check for system call failures. The order of execution of processes is not important. [2+2=4]