# International Institute of Information Technology, Hyderabad

**(Deemed to be University)**

## CS3.301 Operating Systems and Networks – Monsoon 2025

## Mid-Semester Examination

**Max. Time: 1.5 Hr** **Max. Marks: 30**

**Instructions to the students:**

1. Answers written with pencils won't be considered for evaluation.
2. Please **read the descriptions** of the questions (scenarios) **carefully**.
3. There are a total of six questions, with three MCQs, and they carry a total of 30 marks. For MCQs, you can select one or more options (if necessary). Please refrain from writing long explanations for MCQ questions. **Keep the explanations short and to the point.**
4. In addition, there are two bonus questions. These are not mandatory, but solving them will contribute towards the overall course bonus.
5. In general, for descriptive questions, keep the explanations short and to the point. Long answers do not necessarily indicate higher marks.
6. Feel free to make **meaningful assumptions**. However, please **state them clearly**.

**Good Luck**

---

**Welcome again to the LangOS team!!**

Welcome to Project LangOS: The AI-Powered OS and Networks Initiative. A team of AI engineers and data scientists is looking to onboard system designers to build the world's first natural language-powered Operating System! While users interact with the system through natural language ("Can you run this code?" or "send this data to machine A"), the magic happens through traditional OS and Network concepts working behind the scenes.

Your role is to interact with the LangOS design team and tackle key challenges in CPU scheduling, memory management, and transport protocols. You have 90 minutes to help the team with different problems, which will help in upskilling your team. Each task is assigned a specific number of points, and the cumulative score of all the tasks will determine the skill level of the employees (based on total points), with a maximum of 35 points.

1. As a first step to process virtualisation, the team has identified that a process at any point can be in one of the three states: ready, running or blocked. Since you know Unix OS principles, they wanted to check with you when a process should alternate between these states.

   Considering the scenario of two different processes, i) Interactive collaborative coding sessions (user-driven, lots of I/O); and ii) Periodic retraining of language models (*assume these are purely CPU-intensive jobs with no disk or network I/O*), please explain briefly (with a diagram) the states and their transitions. Briefly describe the conditions under which transitions occur, considering the above two examples **(5 points)**

2. The LangOS process virtualisation team is considering the different workload types in the system to decide on a scheduling policy. The team is planning to build a scenario-based scheduling policy selector that dynamically selects a policy based on a given scenario. The team has identified two broad scenarios:

a. Scenario A: Three identical system processes, each requiring 20ms execution time, arrive simultaneously.
b. Scenario B: Three processes with different execution requirements: Process 1 (10ms), Process 2 (20ms), Process 3 (30ms) arrive simultaneously.

Based on the above scenarios, the team wants your help and support in deciding on which scheduling policy to use **(8 points)**:

a. In your opinion, what will be the benefit of using SJF and FIFO schedulers for scenario A, considering the different metrics?
b. For Scenario B, express your opinion by calculating the metrics for SJF, FIFO, and Round Robin (time slice = 1ms) schedulers. Show your work clearly.
c. According to you, under what characteristics would SJF deliver the same turnaround times as FIFO in LangOS?
d. For what types of workloads and quantum lengths would SJF deliver the same response times as Round Robin?

3. You observed that, like many teams, another team has also used paging for memory management, where they have devised a technique using swap space to manage even processes with large virtual address spaces. However, the exact policy for page replacement has yet to be decided. The team has two types of traces of page access in mind, and they are looking at the main memory as a cache with a size of 3 pages:

**Trace 1:** 0, 1, 2, 0, 1, 3, 0, 3, 1, 2, 1

**Trace 2:** 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5

The team is deliberating between different memory management policies, *but they don't want to consider any random policy, and they do understand that an optimal policy cannot be implemented.* Demonstrate using the necessary metrics and the trace table for the different policies – FIFO and LRU, and further, which among them would you recommend and why? What if the main memory can hold 4 pages? **(8 points)** check

4. The LangOS networking group is debugging a connection and prints out part of the TCP header. One of the junior engineers claims **(3 points)**:

*"The TCP header must always include an ACK number, otherwise the receiver won't know which byte to expect next."*

They wanted to know the correct explanation from you.

a. The ACK field is always present in the TCP header, but the ACK bit in the flags determines if the value is valid.
b. The ACK field is included only when the ACK bit is set; otherwise, it is not part of the header.
c. The ACK number exists only in the three-way handshake and is dropped after connection setup.
d. The ACK field is optional and used only if the application enables reliability.

5. In LangOS, a user process issues a `send()` system call to transmit data over a TCP socket. Considering the following possibilities, which among these do you think captures what may happen in such a scenario **(3 points)**:
   a. A context switch always happens immediately because every system call requires the process to be de-scheduled. ✓
   b. A context switch may happen if the kernel's TCP send buffer is full, causing the process to block until space is available.
   c. A context switch never happens because TCP send is handled entirely in user space.
   d. A context switch occurs only if the receiver is on the same machine, as inter-process communication requires switching between processes.

6. The LangOS network team wants to check the end-to-end network communication capabilities. The team is investigating a scenario of a process on Machine A with IP 192.168.1.10 and MAC AA:BB:CC:DD:EE:01 running on Port number 5000, sends data to a process on Machine B with an IP address of 10.0.0.20, MAC AA:BB:CC:DD: EE:02 on Port number 6000. The packet travels through a router **R** with two interfaces:

   - LAN side: IP = 192.168.1.1, MAC = AA:BB:CC:DD:EE:11
   - WAN side: IP = 10.0.0.1, MAC = AA:BB:CC:DD:EE:22

   Given this scenario, in your opinion, which of the following statements is correct with respect to the OSI layer? **(3 points)**

   a. MAC source/destination addresses are rewritten at every hop ✓
   b. ~~The IP addresses of the source and destination change at each router, but others remain the same~~ ✗
   c. The port numbers are reassigned by the router (to avoid conflicts), while the IP and MAC addresses remain the same. ✗
   d. Both IP and MAC addresses change at every hop, but the port numbers remain unchanged.

**Bonus Questions:**

1. The LangOs team has been thinking of using IPV6 and would like to know some details from you. What part of the IPv6 header can be used to track the path of a package in a network? How is it able to track it? **(2 points)**
2. The team is considering using a CFS scheduler, but they have a curiosity. What was the main reason the CFS scheduler replaced the Linux O(1) scheduler? Justify your answer **(3 points)**
   a. Scheduling a task under high loads took an unpredictable amount of time
   b. Low-priority task could wait indefinitely and starve
   c. Interactive tasks could wait for unpredictable amounts of time to be scheduled

************** Now it's time to wait for the final points to see the skill level****************