



National University of Computer & Emerging Sciences, Karachi
CS-Department
FAST School of Computing
Lab Final Examination Fall 2023
11th Jan 2024, 10:00 am – 12:00 pm

Course Code: CL2006	Course Name: Operating Systems Lab
Instructor Names: Mr. Muhammad Monis	
Student Roll No:	Section No:

General Instructions: Carefully read the following instructions before attempting the paper.

- Except your Roll No and Section, **DO NOT WRITE** anything on this paper.
- In case of any ambiguity, you may make assumptions, but your assumption must not contradict any statement in the question paper.
- **DON'T** share your program, if your code is matched to any member of your class, both will get **straight F** in the course without asking who shared or who magically copied.

Submission Instructions:

- You must comment your student ID on top of each file. (Line#1 of your code).
- Name the .c file for each question according to Roll_No e.g. **k22-xxxx_Q1.c, k22- xxxx_Q2.c** etc.
- Create a ZIP folder of all your solutions and copy it in the local storage with the title **K22-xxxx_A**.
- Submission are on local storage that can be accessed using win+r keys and entering [\\172.16.5.43](http://172.16.5.43) address in the dialog box.
- Enter your username as **khifast\K22xxxx** and its assigned password (Default is **Fast1234**).
- Zip folder needs to be pasted in the “**Exam Folder\teacherName\Your_Roll_No**” folder

Total Time: 110 minutes

Maximum Points: 50

Question No 01(Shell Script + Scheduling Algorithm)

LLO: 02 [Marks: 10 Points]

Consider the processes as shell scripts with arbitrary codes in them (You can consider one finding files with special characters {Files that start with some character either than alphanumeric})

1. Create 5 shell scripts files (One finds the files, One copies the file the previous one found, one moves the original files to a test directory, One deletes the copy files, One just prints hello world) 3 points
2. You would have to schedule these files via an scheduling algorithm so take arrival time and burst time as an assumption. 2 points
3. When scheduling you must run these scripts as well (Note: Scripts are very quick to execute so take sleep command with it when executing the tasks). 2 points
4. Scheduling should be done in a Shortest time remaining next fashion (SJF preemptive) 2 points.
5. Calculate the turnaround time for the processes as well. 1 point.

Question No 02(OpenMP + Signals)

LLO: 02 [Marks: 10 Points]

OpenMP is utilized to run programs in parallel so that the resources of the CPU can be utilized efficiently. Write a C/C++ code that generates a random array which should be sorted. (Note: when generating an array each value should be placed in the array in a sorted manner otherwise you are attempting the question incorrectly). You can use any sorting algorithm as long as it fulfills the requirements. All of this work needs to be under the OMP umbrella. Use appropriate methods as to how the data between threads should be shared. While inserting an interrupt of SIGTERM would be called and the array which you were inserting values in an sorted manner so far must be reversed (Note: You can use another array for this without disrupting the original one) an print the array then the system should resume back to sorting.

Note: Generate an array of at least 30 values.

Consider a scenario where there exists a Linked Lists (Note: You don't have make an actual linked list, it's just an assumption and you can use any data structure) where each node is connected to one another. You are tasked to make this data structure via the fork() system call in such a way that the message only goes to the connected node and no other node. Once all the nodes are traversed the last node must send a message traversed to the parent node from where it started.

Below is the depiction of this approach

Node 1-> Node->3 ->Node->5 ->Node->2 -> Node1

Use an appropriate IPC Method to solve the question (Hint: No Unnamed Pipe)

(For message take a number and add it corresponding to the node number and pass it to the other node where it should receive it)

Question No 04(Semaphores)**LLO:04[Marks: 15 Points]**

Write a C program that simulates a quiz game involving multiple players. The program should ask the user for the number of players, the number of questions in the quiz, and the time limit for each question. Each player will be represented by a separate thread. The program should use a semaphore to ensure that only one player can answer a question at a time. For each question, each thread will generate a random number, after which it will be ready to answer the question. This means that the lowest number generated will be the thread which gets to answer that question first. Assume that the first player to answer the question gets 10 points, 2nd gets 5 points, and the rest get 1 point. A player will receive 0 points if the time limit is lesser than the generated random number. The program should output the order in which the players finish each question and their respective scores. Additionally, the program should output the overall winner of the quiz based on the highest score.

Instructions:

1. Implement a C program that takes user input for the number of players, the number of questions, and the time limit for each question.
2. Create a separate thread for each player participating in the quiz.
3. Use a semaphore to ensure that only one player can answer a question at a time.
4. Each player should generate a random number and use it to determine their position in the answering order. The player with the lowest number gets to answer first.
5. Award points to the players based on their answering order: the first player gets 10 points, the second player gets 5 points, and the rest get 1 point. 0 points if the time limit is exceeded.
6. Output the order in which players finish each question and their respective scores.
7. Determine the overall winner of the quiz based on the highest score and print the winner's details.