|  |
| --- |
| **Operating Systems Lab (CL2006)** |
| Date: 07/05/2024 |
| **Course Instructor(s)** |
| Muhammad Monis |

|  |
| --- |
| **Lab Mid Exam (B)** |
| **Total Time: 60 minutes** |
| **Total Marks: 20** |
| **Total Questions**: **01** |
|  |
| **Semester:** SP-2024 |
| **Campus:** Karachi |
| **Dept:** Computer Science |

**Submission Instructions:**

* You are given a separate sheet to solve the question no 1.
* Avoid overwriting on the paper
* All parts of Q1 must be attempted in order with question numbers mentioned.

|  |
| --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Student Name Roll No Section Student Signature |

***CLO # 1|2: Understand and Analyze Command Line tools for Linux OS and Shell scripts for system level programming to automate tasks such as file management, system backups and software installations. Note you have do this on the paper (Total Time for this question 1 hr.)***

**Q1**. [= 20 marks]

1. Write a shell script which takes user input for three items. Your roll number, date and paper type and creates three folders within each other with one notable exception that the folder name should be shuffled each time there is a parent folder (i.e. k224087-05-02-2024-B/B-05-02-2024-k224087/)
2. Write a C/C++ program that take command line arguments for number of files and folders inside the current directory and identify if the number of files are larger than the number of folders or vice versa. If files are larger create a folder to store the files within the C code. If folders are larger create the same number of files to match the ratio of the folders.

| Process | Arrival Time | Burst Time | Priority | I/O Burst Time |
| --- | --- | --- | --- | --- |
| P1 | 0 | 6 | 2 | 1 |
| P2 | 1 | 4 | 1 | 6 |
| P3 | 2 | 5 | 0 | 2 |
| P4 | 3 | 3 | 6 | 3 |

1. The above contains the necessary information required to schedule, solve using the appropriate scheduling algorithms that first schedule based on priority then on I/O burst Time (note this must be small). Calculate the average waiting time for both algorithms. Also mentioned which algorithm you utilized.
2. Create a Make file which makes object as well as executable files in two separate folders. When the make compile O call is initiated the object file should be removed from the folder and a new executable should be made under a different name.
3. If we changed the SIGINT signal, how can we stop the process. Write down the reason as to why we should change the interrupt signal and is there another way to interrupt/stop a process

|  |
| --- |
| **Operating Systems Lab (CL2006)** |
| Date: 07/05/2024 |
| **Course Instructor(s)** |
| Muhammad Monis |

|  |
| --- |
| **Lab Mid Exam (B)** |
| **Total Time: 60 minutes** |
| **Total Marks: 30** |
| **Total Questions**: 01 |
|  |
| **Semester:** SP-2024 |
| **Campus:** Karachi |
| **Dept:** Computer Science |

**Submission Instructions:**

● You must comment your student ID on top of each file. (Line#1 of your code).

● Name the 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\_B** (Your paper Type).

● Submission are on local storage that can be accessed via the other location tab in explorer and then entering the address as **smb://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 Folder2024\teacherName\Your\_Roll\_No”** folder

|  |
| --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Student Name Roll No Section Student Signature |

***CLO # 3: Gain hands on experience in writing code that interacts with operating system services related process and files system, multi-thread programing and different synchronization primitives.***

Q2(a) [15 marks] You are tasked of making a game called Wack a mole, but the operations of the game are changed from the original. Consider 4 threads (being players) and one mole which the players must hit. The hitting criteria of the mole is if the value is between (37 and 55) if the number falls between the range that thread gets the priority to hit the mole. The game would last 10 seconds so come up with a way to close all the threads after this time interval (Note: You can also specify this limit with whichever thread reaches the hit count of 10 first or add a delay after a hit 10 times). The thread with the highest hit number wins. Note: Use the RANDINT function to create random numbers.

Example:

Thread 1 count: 5  
Thread 2 count: 10  
Thread 3 count: 2  
Thread 4 count: 7

Thread 2 wins

**Q2(b)**. [15 marks] Imagine a Formula 1 pit stop scenario where a racing team is preparing to service their car during a race. The pit crew consists of several specialized members, each responsible for a specific task such as changing tires, refueling, and adjusting the car's aerodynamics. You need to utilize semaphores to ensure synchronization between the processes. There exist three processes, Brake Change, Tire change and re-fueling. Each process is dependent on the other but can work concurrently i.e. Fueling can be done alongside with brake change and tire change but tire change and brake change should be synchronized. First the brakes than the tire (obviously). Use semaphores and mutexes to accomplish this tasks

Example:   
Process 1: Refueling  
Process 2: Removing Tires

Process 2: Waiting  
Process 3: Changing Brakes  
Process 3: Done:

Process 1 : Done  
Process 2 : Changing Tires  
Process 2 : Done