CA -2 **CSE316(OPERATING SYSTEM** **K21CS and K21FG**

Instructions

1. This assignment is a compulsory CA component.
2. The assignment is to be done on individual basis (no groups)
3. The assignment submission mode is **Online** only. Student has to upload the assignment on or before the last date on UMS. No submission via e-mail or pen-driveor any media will be accepted.
4. Non-submission of assignment on UMS till the last date will result in **ZERO** marks.
5. The student is supposed to solve the assignment on his/her own. If it is discovered at any stage that the student has used unfair means like copying from peers or copy pasting the code taken from internet etc. **ZERO** marks will be awarded to the student.
6. The student who shares his assignment with other students (either in same section or different section) will also get **ZERO** marks.
7. Prepare a report which contains:
    a. The methodology adopted to solve the problem
    b. Code
    c. Output snapshots
8. You are supposed to upload the progress of the code on the github. Mention the github link in the report file to be uploaded.

Question Allotment Table

K21CS

Roll No.	Question No.	Roll No.	Question No.
1	1	19	3
2	2	20	4
3	3	21	5
4	4	22	6
5	5	23	7
6	6	24	8
7	7	25	1
8	8	26	2
9	1	27	3
10	2	28	4
11	3	29	5
12	4	30	6
13	5	31	7
14	6	32	8
15	7	33	1
16	8		
17	1		
18	2		

K21FG

Roll No.	Question No.	Roll No.	Question No.
1	1	23	7
2	2	24	8
3	3	25	1
4	4	26	2
5	5	27	3
6	6	28	4
7	7	29	5
8	8		
9	1		
10	2		
11	3		
12	4		
13	5		
14	6		
15	7		
16	8		
17	1		
18	2		
19	3		
20	4		
21	5		
22	6		

Questions

1. You are a computer systems engineer working at a large technology company. Your manager has tasked you with creating a simulation program to test the performance of the Round Robin scheduling algorithm. The simulation program should generate aset of "processes" with random arrival times and CPU burst times, and should run the Round Robin algorithm for a set amount of time (e.g. 100 time units). The program should record the average waiting time and turnaround time for each process, and should compare the results with the ideal scenario of a perfect scheduler. Your manager is interested in the results of the simulation to evaluate how well the Round Robin algorithm would perform in a real-world scenario, and to identify any potential issues that need to be addressed. She has given you one week to completethe simulation and to prepare a report of your findings and conclusions.

   As a computer systems engineer, you will need to:
   a. Design and implement the simulation program using a programming language of your choice.
   b. Generate a set of "processes" with random arrival times and CPU burst times

using a random number generator.
c. Implement the Round Robin scheduling algorithm in the simulation program.
d. Have the simulation program run for a set amount of time (e.g. 100 time units) and record the average waiting time and turnaround time for each process.
e. Compare the results of the simulation with the ideal scenario of a perfect scheduler.
f. Write a report of the findings and conclusion with the comparison of the results of the round robin scheduling algorithm with other scheduling algorithms such as First Come First Serve (FCFS)

2. Consider a scheduling approach which is pre-emptive similar to shortest remaining time first in nature. The priority of each job is dependent on its estimated run time, and also the amount oftime it has spent waiting. Jobs gain higher priority the longer they wait, which prevents indefinite postponement. The jobs that have spent a long time waiting compete against those estimated to have short run times. The priority can be computed as :
   Priority = 1+ Waiting time / Estimated run time
   Write a program to implement such an algorithm. Ensure

   1. The input is given dynamically at run time by the user
   2. The priority of each process is visible after each unit of time
   3. The gantt chart is shown as an output
   4. Calculate individual waiting time and average waiting time

3. Student to implement a simulation program that simulates a computer's memory management system. The program should include a memory manager that uses a specific memory allocation algorithm (e.g. First-Fit, Best-Fit, or Worst-Fit) to allocate blocks of memory to processes. The students should also include a mechanism for deallocating memory when a process completes. The simulation should run for a set amount of time and record the average amount of fragmentation and the number of wasted memory blocks at the end of each time unit. The below mentioned outcomes are expected:

   a. **Memory allocation:** The program should demonstrate the ability to allocate blocks of memory to processes using the chosen memory allocation algorithm (e.g. First-Fit, Best-Fit, or Worst-Fit).
   b. **Memory deallocation:** The program should include a mechanism for deallocating memory when a process completes, such as a "garbage collector" or a "free list."
   c. **Fragmentation:** The program should record the average amount of fragmentation at the end of each time unit. Fragmentation occurs when there are small, unused blocks of memory scattered throughout the memory space.
   d. **Wasted memory:** The program should also record the number of wasted memory blocks at the end of each time unit. Wasted memory refers to blocks of memory that are no longer being used by any processes.
   e. **Simulation results:** The program should run the simulation for a set amount of time and display the results, including the average amount of fragmentation and the number of wasted memory blocks, at the end of each time unit. The students should also experiment with different input scenarios and algorithms to compare the results and see how different factors affect the performance of the memory manager.

4. Design a scheduling program to implements a Queue with two levels:
   Level 1 : Fixed priority preemptive Scheduling Level 2 : Shortest Remaining Time First

   For a Fixed priority preemptive Scheduling (Queue 1), the Priority 0 is highest priority. If oneprocess P1 is scheduled and running, another process P2 with higher priority comes. The New process (high priority) process P2 preempts currently running process P1 and process P1will go to second level queue. Time for which process will strictly execute must be considered in the multiples of 2.

   All the processes in second level queue will complete their execution according to SRTF scheduling.
   Consider: 1. Queue 2 will be processed after Queue 1 becomes empty.
   2. Priority of Queue 2 has lower priority than in Queue 1.

5. **Scenario:** A company has a large number of employees who work on various projects and create different types of files such as documents, spreadsheets, and images. The company wants to implement a new file system that can manage the disk space efficiently and handle the high volume of file operations.

   Problem: Create a simulation program that simulates a file system for the company. The program should include a file system manager that uses a specific file allocation algorithm (e.g. Contiguous Allocation or Linked Allocation) to allocate space for files on a simulated disk. The students should also include a mechanism for deleting, renaming and moving files. The simulation should run for a set amount of time and record the average amount of fragmentation and the number of wasted disk blocks at the end of each time unit.

   Consider the following factors in their simulation:
   • The employees will be creating and editing different types of files (e.g. documents, spreadsheets, images) with varying file sizes.
   • The employees will be frequently adding and deleting files.
   • The company wants to minimize the amount of wasted disk space.

   Provide a report on their findings and observations, including the performance of the file system under different scenarios and the trade-offs involved in the different file allocation algorithms (Contiguous or Linked Allocation).

   Expected outcomes:
   1. File allocation: The program should demonstrate the ability to allocate space for files on a simulated disk using the chosen file allocation algorithm (e.g. Contiguous Allocation or Linked Allocation).
   The program should run the simulation for a set amount
   of time and display the results, including the average amount of fragmentation and the number of wasted disk blocks, at the end of each time unit.

6. Ajay Kalia is a Banking Expert who wants to have an online system where he can handle customer queries. Since there can be multiple requests at any time he wishes to dedicate a fixed amount of time to every request so that everyone gets a fair share of his time. He will log into the system from 9am to 1pm only. He wants to have separate requests queues for regular customers and new customers. Implement a strategy for the same. The summary at the end of the session should include the total time he spent on handling queries and average query time.

7. Write a multithreaded program that implements the banker's algorithm. Create n threads that request and release resources from the bank. The banker will grant the request only if it leaves the system in a safe state. It is important that shared data be safe from concurrent access. To ensure safe access to shared data, you can use mutex locks.

   Ensure:
   The program should be dynamic such that the threads are created at run time based on the input from the user.
   The resources must be displaced after each allocation. The system state should be visible after each allocation

8. Write a program for multilevel queue scheduling algorithm. There must be three queues generated. There must be specific range of priority associated with every queue. Now prompt the user to enter number of processes along with their priority and burst time. Each process must occupy the respective queue with specific priority range according to its priority. Apply Round Robin algorithm with quantum time 3 on queue with highest priority range. Apply priority scheduling algorithm on the queue with medium range of priority and SJF algorithm on the queue with lowest range of priority. Each and every queue should get a quantum time of 10 seconds. CPU will keep on shifting between queues after every 10 seconds.