CSE316 – OPERATING SYSTEMS

Section: K18BY | **Instructor**: Mr. Sumit Mittu (12735) | **Term:** 19202

Academic Task 3 – Assignment (Simulation Based)

(COMPULSORY)

Last Date of Submission (online): Monday, 6th April 2020 **Max. Marks: 30**

Nature of Task: The Academic Task #3 will be a simulation-based assignment to be implemented using computer programming. In this academic task, the students' will be writing and simulating different standard algorithms of operating system in C.

Instructions for the assignment submission

1. This assignment is a compulsory CA component.
2. The assignment is to be done on individual basis (no groups). Each student will submit the assignment of questions that are assigned individually.
3. The assignment submission mode is Online only. Student has to upload the assignment on or before the last date on UMS only. No submission via e-mail or pen-drive or 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.

Assignment Evaluation Rubrics/Parameters

Assessment Criteria

Parameter	Weightage in %	Description
Test Cases	**20**	The code must satisfy the sample test cases i.e. for each set of input it must generate the desirable output.
Concept Clarity	**10**	The student needs to specify the algorithms and the OS concepts they are applying on the given scenario-based problem.

Parameter	Weightage in %	Description
Functional Requirements (boundary conditions, constraint satisfaction, etc.)	**50**	Code has to fulfill all the constraints and will satisfy the boundary conditions mentioned in the problem. It is mandatory to use C language. Solution should be implemented using OS concepts (System calls)
Report submission	**10**	The student has to submit the report as per the format specified.
Use of GitHub Repository	**10**	Student should upload the project on GitHub repository, every week at least one revision should be done with a total of minimum 5 revisions during project lifecycle. Students uploading project in GitHub in the last week of submission would be subjected to DEDUCTION OF MARKS.

Marks deduction on the basis of similarity index

Plagiarism Weightage	Marks Deducted
50-60%	03
60-70%	06
70-80%	09
80-90%	12
Above 90%	27

Report Format

The individual project report template is built for letting us evaluate your individual understanding, capability and retention of the assigned project. While building answers to the questions from the template, you should be relating the project assigned to you and keep in mind the below three points and submit a write up to for the project.

- **Text Size**: 12
- **Text Style**: Times New Roman
- **Line Spacing**: 1.5 maximum

Mention the below in header of the word document

Student Name: **Registration No.:**

Student ID:

Email Address:

GitHub Link:

Code: Mention solution code assigned to you

1. Explain the problem in terms of operating system concept? (Max 200 word)

Description:

2. Write the algorithm for proposed solution of the assigned problem.

Algorithm:

3. Calculate complexity of implemented algorithm. (Student must specify complexity of each line of code along with overall complexity)

Description (purpose of use):

4. Explain all the constraints given in the problem. Attach the code snippet of the implemented constraint.

Code snippet:

5. If you have implemented any additional algorithm to support the solution, explain the need and usage of the same.

Description:

6. Explain the boundary conditions of the implemented code.

Description:

7. Explain all the test cases applied on the solution of assigned problem**.**

Description:

8. Have you made minimum 5 revisions of solution on GitHub?

GitHub Links:

Question Allocation

Sr	Question Statement	Allocated to Roll Nos.
1.	A university computer science department has a teaching assistant (TA) who helps undergraduate students with their programming assignments during regular office hours. The TA's office is rather small and has room for only one desk with a chair and computer. There are three chairs in the hallway outside the office where students can sit and wait if the TA is currently helping another student. When there are no students who need help during office hours, the TA sits at the desk and takes a nap. If a student arrives during office hours and finds the TA sleeping, the student must awaken the TA to ask for help. If a student arrives and finds the TA currently helping another student, the student sits on one of the chairs in the hallway and waits. If no chairs are available, the student will come back at a later time. Using threads, mutex locks, and semaphores, implement a solution that coordinates the activities of the TA and the students.	**1** **2** **3** **4**
2.	An auto driver charges Rs 10 per customer irrespective of the distance the customer has to travel within the city. All customers are picked up from common point where the auto driver has to return after dropping customer to destination. Considering N customers request for auto service mentioning the distance (measured equivalent to time units) each of them wants to travel, write a computer program that will help the auto driver earn maximum amount of money by travelling least possible distance (equivalent to time units) and that average time for which customers wait for their pickup by auto driver is also as least as possible.	**5** **6** **7** **8**
3.	If a teacher is being served at the food mess and during the period when he is being served, another teacher comes, then that teacher would get the service (food) next. This process might continue leading to increase in waiting time of students to get food. Propose an algorithm and implement it using a C program to ensure average waiting time of students is minimized and that students do not get starved.	**9** **10** **11** **12**
4.	Suppose that a disk drive has 5,000 cylinders, numbered 0 to 4999. The drive is currently serving a request at cylinder143, and the previous request was at cylinder 125. The queue of pending requests, in FIFO order, is: 86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130 starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests for each of the SCAN disk-scheduling algorithms?	**13** **14** **15** **16**
5.	Ten students (s1,s2,s3,s4,s5,s6,s7,s8,s9,s10) are going to attend an event. There are lots of gift shops, they all are going to the gift shops and randomly picking the gifts. After picking the gifts they are randomly arriving in the billing counter. The accountant gives the preference to that student who has maximum number of gifts. Create a C program to define order of billed students?	**17** **8** **9** **20**

Sr	Question Statement	Allocated to Roll Nos.
6.	A set of processes are being scheduled using a preemptive, round robin scheduling algorithm. Each process is assigned a numerical priority, with a higher number indicating a higher relative priority. In addition to the processes listed below, the system also has an idle task (which consumes no CPU resources and is identified as P_idle). This task has priority 0 and is scheduled whenever the system has no other available processes to run. The length of a time quantum is 10 units. If a process is preempted by a higher-priority process, the preempted process is placed at the end of the queue. Use following data as one of the test cases. **Thread Priority Burst Arrival** P1 40 20 0 P2 30 25 25 P3 30 25 30 P4 35 15 60 P5 5 10 100 P6 10 10 105 Write a C code to a. Show the scheduling order of the processes using a Gantt chart. b. What is the turnaround time for each process? c. What is the waiting time for each process? d. What is the CPU utilization rate?	**21** **22** **23** **24**
7.	Write a program in C which reads input CPU bursts from the first line of a text file named as CPU_BURST.txt. Validate the input numbers whether the numbers are positive integers or not. Consider the numbers as CPU burst. If there are 5 positive integers in the first line of the text file then the program treats those arguments as required CPU bust for P1, P2, P3, P4, and P5 process and calculate average waiting time and average turn-around time. Consider used scheduling algorithm using preemptive shortest job first and arrival time of a process every 3 units of time.	**25** **26** **27** **28**
8.	Design a scheduling program capable of scheduling many processes that arrive at some time interval and are allocated the CPU not more than 10 time units. CPU must schedule processes having short execution time first. CPU is idle for 3 time units and does not entertain any process prior this time. Scheduler must maintain a queue that keeps the order of execution of all the processes. Compute average waiting and turnaround time.	**29** **30** **31** **32**
9.	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 4 on queue with highest priority range. Apply priority scheduling algorithm on the queue with medium range of priority and First come first serve 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.	**33** **34** **35** **36**

Sr	Question Statement	Allocated to Roll Nos.
10.	Write a program in C which reads input CPU bursts from the first line of a text file named as CPU_BURST.txt. Validate the input numbers whether the numbers are positive integers or not. Consider the numbers as CPU burst. If there are 5 positive integers in the first line of the text file then the program treats those arguments as required CPU bust for P1, P2, P3, P4, and P5 process and calculate average waiting time and average turn-around time. Consider used scheduling algorithm using non-preemptive shortest job first and arrival time of a process every 3 units of time.	**37** **38** **39** **40**
11.	Design a Prioritized Round Robin (PRR) algorithm which is a combination of priority scheduling and round robin scheduling algorithm such that if the time round-robin time quantum is Q time units then the actual CPU time slot allocated to any process will be determined by scaling Q by priority level of the process. The process can have integer priority defined in the range from 1 (lowest) to 5 (highest). Implement this algorithm using a C program and test run it over a set of processes with given CPU burst time requirement. Generate the Gantt chart and compute average waiting time using your program.	**41** **42** **43** **44**
12.	CPU schedules N processes which arrive at different time intervals and each process is allocated the CPU for a specific user input time unit, processes are scheduled using a preemptive round robin scheduling algorithm. Each process must be assigned a numerical priority, with a higher number indicating a higher relative priority. In addition to the processes one task has priority 0. The length of a time quantum is T units, where T is the custom time considered as time quantum for processing. If a process is preempted by a higher priority process, the preempted process is placed at the end of the queue. Design a scheduler so that the task with priority 0 does not starve for resources and gets the CPU at some time unit to execute. Also compute waiting time, turn around.	**45** **46** **47** **48**
13.	A number of cats and mice inhabit a house. The cats and mice have worked out a deal where the mice can steal pieces of the cats' food, so long as the cats never see the mice actually doing so. If the cats see the mice, then the cats must eat the mice (or else lose face with all of their cat friends). There are NumBowls cat food dishes, NumCats cats, and NumMice mice. Your job is to synchronize the cats and mice so that the following requirements are satisfied: No mouse should ever get eaten. You should assume that if a cat is eating at a food dish, any mouse attempting to eat from that dish or any other food dish will be seen and eaten. When cats aren't eating, they will not see mice eating. In other words, this requirement states that if a cat is eating from any bowl, then no mouse should be eating from any bowl. Only one mouse or one cat may eat from a given dish at any one time. Neither cats nor mice should starve. A cat or mouse that wants to eat should eventually be able to eat. For example, a synchronization solution that permanently prevents all mice from eating would be unacceptable. When we actually test your solution, each simulated cat and mouse will only eat a finite number of times; however, even if the simulation were allowed to run forever, neither cats nor mice should starve.	**49** **50** **51** **52**

Sr	Question Statement	Allocated to Roll Nos.
14.	Sudesh Sharma is a Linux expert who wants to have an online system where he can handle student 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 10am to 12am only. He wants to have separate requests queues for students and faculty. 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.	**53** **54** **55** **56**
15.	A barbershop consists of a waiting room with n chairs and a barber room with one barber chair. If there are no customers to be served, the barber goes to sleep. If a customer enters the barbershop and all chairs are occupied, then the customer leaves the shop. If the barber is busy but chairs are available, then the customer sits in one of the free chairs. If the barber is asleep, the customer wakes up the barber. Write a program to coordinate the barber and the customers.	**57** **58** **59** **60**
16.	Write a program that implements the FIFO page replacement algorithm. First, generate a random page-reference string where page numbers range from 0 to 9. Apply the random page-reference string to each algorithm, and record the number of page faults incurred by each algorithm. Implement the replacement algorithm so that the number of page frames can vary from 1 to 7. Assume that demand paging is used.	**61** **62** **63**
17.	Write a program that implements the least frequently used page replacement algorithm. First, generate a random page-reference string where page numbers range from 0 to 9. Apply the random page-reference string to each algorithm, and record the number of page faults incurred by each algorithm. Implement the replacement algorithm so that the number of page frames can vary from 1 to 7. Assume that demand paging is used.	**64** **65** **66**
18.	Write a program that implements the most recently used page replacement algorithm. First, generate a random page-reference string where page numbers range from 0 to 9. Apply the random page-reference string to each algorithm, and record the number of page faults incurred by each algorithm. Implement the replacement algorithm so that the number of page frames can vary from 1 to 7. Assume that demand paging is used.	**67** **68** **69**
19.	Implement the preemptive Priority CPU scheduling algorithm. Compare the average waiting time per process considering context switch time over head with average waiting time per process not considering context switch time over head.	**70** **71** **79**

****