LOVELY PROFESSIONAL UNIVERSITY

Academic Task-3 (Compulsory Assignment)

School of Computer Science and Engineering Faculty of Technology And Sciences

Name of the faculty member :- Amanpreet Kaur Section: 1651

Course Code: CSE316 Course Title: Operating Systems

Term: 17182
Max. Marks : 30 Date of Allotment:13-03-2018 Date of Submission:28-04-2018

Instructions for 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.

Question Allocation Table:

Roll No.	Question No.	Roll No.	Question No.	Roll No.	Question No.	Roll No.	Question No.
1	4	18	6	34	7	50	4
2	18	19	12	35	22	51	7
3	16	20	21	36	20	52	9
4	13	21	20	37	18	53	18
5	24	22	25	38	12	54	21
6	9	23	7	39	11	55	22
7	5	24	4	40	10	57	14

8	11	25	8	41	17	58	3
9	14	26	9	42	21		
11	22	27	13	43	24		
12	19	28	14	44	3		
13	13	29	23	45	5		
14	8	30	25	46	15		
15	1	31	21	47	16		
16	7	32	3	48	12		
17	2	33	5	49	2		

List of Questions:

Develop a scheduler which submits the processes to the processor in the following scenario, and compute the scheduler performance by providing the waiting time for process, turnaround time for process and average waiting time and turnaround time.

Ques. 1. Considering 4 processes with the arrival time and the burst time requirement of the processes the scheduler schedules the processes by interrupting the processor after every 3 units of time and does consider the completion of the process in this iteration. The schedulers then checks for the number of processes waiting for the processor and allots the processor to the process but interrupting the processor after every 6 units of time and considers the completion of the process in this iteration. The scheduler after the second iteration checks for the number of processes waiting for the processor and now provides the processor to the process with the least time requirement to go in the terminated state.
The inputs for the number of requirements, arrival time and burst time should be provided by the user.

Consider the following units for reference.

Process	Arrival time	Burst time
P1	0	18
P2	2	23
P3	4	13
P4	13	10

Ques. 2. Considering the arrival time and burst time requirement of the process the scheduler schedules the processes by interrupting the processor after every 6 units of time and does consider the completion of the process in this iteration. The scheduler than checks for the number of process waiting for the processor and allots the processor to the process but interrupting the processor every 10 unit of time and considers the completion of the processes in this iteration.

The scheduler checks the number of processes waiting in the queue for the processor after the second iteration and gives the processor to the process which needs more time to complete than the other processes to go in the terminated state.

The inputs for the number of requirements, arrival time and burst time should be provided by the user.

Consider the following units for reference.

Process	Arrival time	Burst time
P1	0	20
P2	5	36
P3	13	19
P4	26	42

Develop a scheduler which submits the processes to the processor in the defined scenario, and compute the scheduler performance by providing the waiting time for process, turnaround time for process and average waiting time and turnaround time.

Ques. 3. Consider a scheduler which schedules the job by considering the arrival time of the processes where arrival time if given as 0 is discarded or displayed as error. The scheduler implements the shortest job first scheduling policy, but checks the queue of the processes after the every process terminates and time taken for checking and arranging the process according to the shortest job is 2 time unit. Compute the waiting time, turnaround time and average waiting time and turnaround time of the processes. Also compute the total time taken by the processor to compute all the jobs.

The inputs for the number of requirements, arrival time and burst time should be provided by the user.

Consider the following units for reference.

Process	Arrival time	Burst Time
1	0	6
2	3	2
3	5	1
4	9	7
5	10	5
6	12	3
7	14	4
8	16	5
9	17	7
10	19	2

Develop a scheduler which submits the processes to the processor in the defined scenario, and compute the scheduler performance by providing the waiting time for process, turnaround time for process and average waiting time and turnaround time.

Ques. 4. Consider a scheduling approach which is non pre-emptive similar to shortest job next in nature. The priority of each job is dependent on its estimated run time, and also the amount of time 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
Using the data given below compute the waiting time and turnaround time for each process and average waiting time and average turnaround time.

Process	Arrival time	Burst time
P_1	0	20
P_2	5	36
P_3	13	19
P_4	17	42

Ques. 5. 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.

Ques. 6. Design a scheduling program that is capable of scheduling many processes that comes in at some time interval and are allocated the CPU not more that 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.

Ques. 7. Design a scheduling program to implements a Queue with two levels:
Level 1 : Fixed priority preemptive Scheduling
Level 2 : Round Robin Scheduling
For a Fixed priority preemptive Scheduling (Queue 1), the Priority 0 is highest priority. If one process 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 P1 will 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 round robin 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.

Ques. 8. 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.

Ques. 9. Design a scheduler that uses a preemptive priority scheduling algorithm based on dynamically changing priority. Larger number for priority indicates higher priority.
Assume that the following processes with arrival time and service time wants to execute (for reference):

ProcessID	Arrival Time	Service Time
P1	0	4
P2	1	1
P3	2	2
P4	3	1

When the process starts execution (i.e. CPU assigned), priority for that process changes at the rate of m=1.When the process waits for CPU in the ready queue (but not yet started execution), its priority changes at a rate n=2. All the processes are initially assigned priority value of 0 when they enter ready queue for the first time . The time slice for each process is q = 1. When two processes want to join ready queue simultaneously, the process which has not executed recently is given priority. Calculate the average waiting time for each process. The program must be generic i.e. number of processes, their burst time and arrival time must be entered by user.

Ques. 10. Design a scheduler with multilevel queue having two queues which will schedule the processes on the basis of pre-emptive shortest remaining processing time first algorithm (SROT) followed by a scheduling in which each process will get 2 units of time to execute. Also note that queue 1 has higher priority than queue 2. Consider the following set of processes (for reference)with their arrival times and the CPU burst times in milliseconds.

Process	Arrival-Time	Burst-Time
P1	0	5
P2	1	3
P3	2	3
P4	4	1

Calculate the average turnaround time and average waiting time for each process. The input for number of processes and their arrival time, burst time should be given by the user.

Ques. 11. Reena’s operating system uses an algorithm for deadlock avoidance to manage the allocation of resources say three namely A, B, and C to three processes P0, P1, and P2. Consider the following scenario as reference .user must enter the current state of system as given in this example :

Suppose P0 has 0,0,1 instances , P1 is having 3,2,0 instances and P2 occupies 2,1,1 instances of A,B,C resource respectively.

Also the maximum number of instances required for P0 is 8,4,3 and for p1 is 6,2,0 and finally for P2 there are 3,3,3 instances of resources A,B,C respectively. There are 3 instances of resource A, 2 instances of resource B and 2 instances of resource C available. Write a program to check whether Reena’s operating system is in a safe state or not in the following independent requests for additional resources in the

current state:

1. Request1: P0 requests 0 instances of A and 0 instances of B and 2 instances of C.
2. Request2: P1 requests for 2 instances of A, 0 instances of B and 0 instances of C.

All the request must be given by user as input.

Ques. 12. Three students (a, b, c) are arriving in the mess at the same time. The id numbers of these students are 2132, 2102, 2453 and the food taken time from the mess table is 2, 4 and 8 minutes. If the two students have same remaining time so it is broken by giving priority to the students with the lowest id number. Consider the longest remaining time first (LRTF) scheduling algorithm and calculate the average turnaround time and waiting time.

Ques. 13. 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 i.e. to apply round robin algorithm OF 10 seconds on over all structure.
Calculate Waiting time and turnaround time for every process. The input for number of processes should be given by the user.

Ques. 14. Write a program to implement priority scheduling algorithm with context switching time. Prompt to user to enter the number of processes and then enter their priority, burst time and arrival time also. Now whenever operating system preempts a process and shifts cpu's control to some another process of higher priority assume that it takes 2 seconds for context switching(dispatcher latency).Form a scenario, where we can give the processes are assigned with priority where the lower integer number is higher priority and then context switch .. as the process waits the priority of the process increase at rate of one per 2 time units of wait.
Calculate waiting time and turnaround time for each process.

Ques. 15. A uniprocessor system has n number of CPU intensive processes, each process has its own requirement of CPU burst. The process with lowest CPU burst is given the highest priority. A late-arriving higher priority process can preempt a currently running process with lower priority. Simulate a scheduler that is scheduling the processes in such a way that higher priority process is never starved due to the execution of lower priority process. What should be its average waiting time and average turnaround time if no two processes are arriving are arriving at same time.

Ques. 16. Design a scheduler that can schedule the processes arriving system at periodical intervals. Every process is assigned with a fixed time slice t milliseconds. If it is not able to complete its execution within the assigned time quantum, then automated timer generates an interrupt. The scheduler will select the next process in the queue and dispatcher dispatches the process to processor for execution. Compute the total time for which processes were in the queue waiting for the processor. Take the input for CPU burst, arrival time and time quantum from the user.

Ques. 17. Design a scheduler following non-preemptive scheduling approach to schedule the processes that arrives at different units and having burst time double the arrival time. Scheduler selects the process with largest burst time from the queue for the execution. Process is not being preempted until it finishes its service time. Compute the average waiting time and average turnaround time. What should be the average waiting time if processes are executed according to Shortest Job First scheduling approach with the same attribute values.

Ques. 18. Ten students (a,b,c,d,e,f,g,h,i,j) 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 or Java program to define order of billed students?

Ques. 19. There are 5 processes and 3 resource types, resource A with 10 instances, B with 5 instances and C with 7 instances. Consider following and write a c code to find whether the system is in safe state or not?

Available			Processes	Allocation			Max		
A	B	C		A	B	C	A	B	C
3	3	2	P0	0	1	0	7	5	3
			P1	2	0	0	3	2	2
			P2	3	0	2	9	0	2
			P3	2	1	1	2	2	2
			P4	0	0	2	4	3	3

Ques. 20. Consider that a system has P resources of same type. These resources are shared by Q processes time to time. All processes request and release the resources one at a time. Generate a solution to demonstrate that, the system is in safe state when following conditions are satisfied.

Conditions:

1. Maximum resource need of each process is between 1 and P.
2. Summation of all maximum needs is less than P+Q

Ques. 21. Consider a scenario of demand paged memory. Page table is held in registers. It takes 8 milliseconds to service a page fault if an empty page is available or the replaced page is not modified and 20 milliseconds if the replaced page is modified. Memory access time is 100 nanoseconds. Assume that the page to be replaced is modified 70 percent of the time. Generate a solution to find maximum acceptable page-fault rate for access time that is not more than 200 nanoseconds.

Ques. 22. Consider following and Generate a solution to find whether the system is in safe state or not?

Available				Processes	Allocation				Max			
A	B	C	D		A	B	C	D	A	B	C	D
1	5	2	0	P0	0	0	1	2	0	0	1	2
				P1	1	0	0	0	1	7	5	0
				P2	1	3	5	4	2	3	5	6
				P3	0	6	3	2	0	6	5	2
				P4	0	0	1	4	0	6	5	6

Que 23. Consider a concurrent program that includes two functions, called funcA and funcB. This program has the following synchronization requirements, both of which must be satisfied.

Req. 1: At most one thread at a time may be running funcB.

Req. 2: At most two threads at a time may be running any combination of funcA or funcB.

These requirements are summarized in the table on the right, which shows which combinations of funcA and funcB may be executed concurrently. Note that it is never OK for more than two threads to be running any combination of these functions concurrently. Your task is to determine how to enforce these synchronization requirements using semaphores. You must not use any other synchronization primitives, e.g., spinlocks, locks, wait queues, condition variables. Your solution should not be more restrictive than necessary, and it should ensure that deadlock is not possible.

	Func A	Func B
Func A	OK	OK
Func B	OK	NO

Concurrent Function Execution Requirements

Que 24. You have been hired by Snowflake entertainment to design the matchmaking system for their new multiplayer online game. In this game, a match consists of 3 players and can only start when all 3 players are available. The company owns only one server and the server can only host one match at a time. A new match can start on the server only when (a) the previous match has finished, and (b) three players are available to play. Implement the following three functions to satisfy the specified constraints.

Que 25. Suppose that a system initially has only a single thread (T1), which is running the function f1(). f1() calls pthread_create to create a new thread, T2, to run function f2() concurrently with T1. These two threads and the functions they execute are illustrated in the two columns of the figure below. As illustrated in the figure, thread T1 calls function fb(), and thread T2 calls functions fa() and fc(). Your task is to synchronize the calls to fa(), fb(), and fc() so that the following synchronization rules are enforced:

(1) function fa() finishes before function fb() is called, and

(2) function fb() finishes before function fc() is called.

Show how to use semaphore(s) to enforce these sychronization rules by adding P() and V() calls in suitable places in functions f1() and f2() in the diagram below. You must also declare (at the top of the diagram) global variables to point to any semphores you use in your solution. Finally, you must include sem create calls in suitable places in f1() and/or f2() to create and initialize the semaphores you need. Be sure that your sem create() calls show the intial semaphore value for each newly created semaphore. Do not use any synchronization primitives or techniques other than sempahores. Keep your solution as simple as possible.