**LOVELY PROFESSIONAL UNIVERSITY**

**Academic Task-3 (Operating System)**

School of Computer Science and Engineering  Faculty of Technology And Sciences

Name of the faculty member  Neha Sharma

Course Code: CSE 316  Course Title: Operating System

Term: 18192

Max. Marks:30  Date of Allotment: 11/02/2018  Date of Submission: 17/04/2018

Time Duration: 09 Weeks

## 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.

## Simulation Based Assignment Assessment Rubric

### 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 need to specify the algorithms and the OS concepts they are applying on the given scenario based problem. |
| **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. |

**Note: 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**

Dear Students,

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.

**Format of the report:**

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

**Mention the below in header of the word document**

**Student Name:**
**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 Link:**

## List of Questions:

**Q1.** 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 |

Develop a scheduler which submits the processes to the processor in the above 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.

**Q2.** 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.

**Q3.** 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.

**Q4**. 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.

**Q5**. 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 |

**Q6.**. 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.

**Q7**. 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.

**Q8.** Mahesh 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.

**Q9**. 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.

**Q10**. 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.

**Q11.**. 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.

**Q12**. 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.

**Q13**. 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.

**Q14.**. 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.

**Q15**. 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.

**Q16**. 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.

**Q17**. 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.

**Q18**. 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?

**Q19**. 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 |

**Q20**. 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.

**Q21**. 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

**Q22**. 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.

**Q23**. 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 |

**Q24.** 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.

**Q25.** Design 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.

**Q26.** Write a program in C which will accept5 positive integers as command line arguments. Performs check whether there are 5 arguments or not. If number of arguments are lesser or greater than 5 then it will print a relevant message to the user. If you enter 5 positive integers from command line then the program treat those argument as required CPU bust for P1, P2, P3, P4, andP5 process and calculate average waiting time and average turnaround time. Consider used scheduling algorithm as FCFS and same arrival time for all the processes.

**Q27.** Write a program in C which reads input CPU bursts from a 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 treat those argument as required CPU bust for P1, P2, P3, P4, and P5 process and calculate average waiting time and average turnaround time. Consider used scheduling algorithm as SJF and same arrival time for all the processes.

**Q28**. Given five memory partitions of 100 KB, 500 KB, 200 KB, 300 KB, and 600 KB (in order), how would each of the first-fit, best-fit, and worst-fit algorithms place processes of 212 KB, 417 KB, 112 KB, and 426 KB (in order)? Write a program in C which will print the mapping of processes with memory partitions for both the algorithms.(for conceptual clarity refer the textbook)

**Q29**. Write a C program to create a page table for a program of 5MB. Consider page size as 2KB. Assign frame numbers randomly in page table. (for conceptual clarity refer the textbook)

**Q30.** Write a C program to solve the following problem:

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 FCFS disk-scheduling algorithms? (for conceptual clarity refer the textbook)

**Q31.** Write a C program to solve the following problem:

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 SSTF disk-scheduling algorithms? (for conceptual clarity refer the textbook)

**Q32**. Write a C program to solve the following problem:

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?

**Q33**. Write a C program to solve the following problem:

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 CSCAN disk-scheduling algorithms?

**Q34.** Write a C program to solve the following problem:

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 LOOK disk-scheduling algorithms?

**Q35.** Write a C program to solve the following problem:

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 C-LOOK disk-scheduling algorithms?

**Q36.** Consider the following set of processes, with the length of the CPU burst given in milliseconds:

| Process | Burst Time |
|---|---|
| $P_1$ | 6 |
| $P_2$ | 8 |
| $P_3$ | 7 |
| $P_4$ | 3 |

Schedule the processes using SJF scheduling, and print the outlook of Gant Chart on the computer screen.

Example:

|_____|_____|_____|___|

**Q37.** Consider the following four processes, with the length of the CPU burst given in milliseconds

| Process | Arrival Time | Burst Time |
|---|---|---|
| $P_1$ | 0 | 8 |
| $P_2$ | 1 | 4 |
| $P_3$ | 2 | 9 |
| $P_4$ | 3 | 5 |

| Process | Burst Time | Priority |
|---|---|---|
| $P_1$ | 10 | 3 |
| $P_2$ | 1 | 1 |
| $P_3$ | 2 | 4 |
| $P_4$ | 1 | 5 |
| $P_5$ | 5 | 2 |

Write a C program to calculate average waiting time using shortest-remaining-time-first scheduling.

**Q38**. consider the following set of processes, assumed to have arrived at time 0, in the orderP1, P2, • • • , P5, with the length of the CPU burst given in milliseconds:

Write a C program to schedule the processes using priority scheduling and calculate average waiting time. Consider Lower the number higher the priority.

**Q39.**Thereare five processes in the system. All five processes arrive at time 0, in the order given, with the length of the CPU burst given in milliseconds:

| Process | Burst Time |
|---|---|
| $P_1$ | 10 |
| $P_2$ | 29 |
| $P_3$ | 3 |
| ***Pi*** | 7 |
| $P_5$ | 12 |

Write a C program which will incorporate SFJ scheduling and print the ending time of process Pi.

**Q40.** Consider the following set of processes, with the length of the CPU burst given in milliseconds:

| Process | Burst Time | Priority |
|---|---|---|
| $P_1$ | 10 | 3 |
| $P_2$ | 1 | 1 |
| $P_3$ | 2 | 3 |
| $P_4$ | 1 | 4 |
| $P_5$ | 5 | 2 |

The processes are assumed to have arrived in the order Pi, P2, P3, P4, P5, all at time 0. Write a C program to calculate the turnaround time of each process by incorporating SJF scheduling.

**Q41**. Write a program which incorporate Peterson's solution for synchronizing two processes those are simultaneously trying to write on a shared file named as File.txt. (use system calls only)

**Q42.** The Sleeping-Barber Problem. 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.

**Q43.** Develop a scheduler which submits 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.

Considering the arrival time and the burst time requirement of the processes the scheduler schedules the processes by interrupting the processor after every 3units of time and does consider the completion of the process in this iteration. The scheduler 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 these 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**.

**Q44**. Develop a scheduler which submits 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.

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 then 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.**

**Q45.** If a teacher is being served and during the period when he is being served another teacher comes, then that teacher would get the service next. This process might continue leading to increase in waiting time of students. Write a C program to ensure in your program that the waiting time of students is minimized. Assume values of arrival time, burst time by your own.

**Q46.** Consider a scheduling approach which is nonpre-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

Write a C program which will use the data given below and compute the waiting time and turnaround time for each process and average waiting time and average turnaround time.

| Process | Arrival time | Burst time |
|---|---|---|
| P1 | 0 | 20 |
| P2 | 5 | 36 |
| P3 | 13 | 19 |
| P4 | 17 | 42 |

**Q47.** For SJF algorithm,

(i) We randomly generate the number of jobs. There must be a limit on the number of jobs in a system. (ii)The execution time of the generated jobs is also not known. Here, we are generating the CPU burst of each job making use of the past history.

(iii)All the jobs are then arranged in a queue where searching is done to find the one with the least CPU burst. There may be two jobs in queue with the same execution time then FCFS approach is to be performed.

Case a) If the algorithm is non-preemptive in nature, then the newly arriving job is to be added to the job queue even though it is of lesser execution time than the one running on the processor.

Case b) Otherwise pre-emption is performed.

Assume values of arrival time, burst time (when required) by your own.

**Q48.**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 pre-empted by a higher priority process, the pre-empted 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.

**Q49.** 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 than10 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.

**Q50.** Design a scheduling program to implements a Queue with two levels:

Level 1: Fixed priority pre-emptive Scheduling

Level 2: Round Robin Scheduling

For a Fixed priority preemptive Scheduling (Queue1), thePriority 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 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 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.

**Q51.** Design a scheduler that uses a pre-emptive 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):

| Process | AT | Service Time (BT) |
|---|---|---|
| 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.

**Q52** .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 queue2. Consider the following set of processes (for reference) with the arrival times and the CPU burst times in milliseconds.

| Pro | AT | BT |
|---|---|---|
| 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.

**Q53**. consider a system with five processes P0 through P4 and three resource types A, B and C. Resource type A has10 instances, resource type B has 5 instances, and resource type C has 7 instances.

Suppose that, at time T0, the following snap shot of the system has been taken

| | *Allocation* | *Max* | *Available* |
|---|---|---|---|
| | *ABC* | *A B C* | *ABC* |
| $P_0$ | 0 1 0 | 7 5 3 | 3 3 2 |
| p, | 2 0 0 | 3 2 2 | |
| $P_2$ | 3 0 2 | 9 02 | |
| $P_3$ | 21 1 | 2 2 2 | |
| ***Pi*** | 0 0 2 | 4 3 3 | |

Implement the Banker's algorithm to claim that the system is in safe state or not?

**Q54.** ASSIGNMENT COMPLETION PROBLEM

There are 3 student processes and 1 teacher process. Students are supposed to do their assignments and they need 3 things for that-pen, paper and question paper. The teacher has an infinite supply of all the three things. One students has pen, another has paper and another has question paper. The teacher places two things on a shared table and the student having the third complementary thing makes the assignment and tells the teacher on completion. The teacher then places an other two things out of the three and again the student having the third thing makes the assignment and tells the teacher on completion. This cycle continues. WAP to synchronise the teacher and the students.

| RollNumber | Name | Ques 1 | Ques 2 |
|---|---|---|---|
| RK17EBA01 | Kartik Tyagi | 1 | 28 |
| RK17EBA02 | Dheeraj Kumar | 2 | 29 |
| RK17EBA03 | Adapala Suneel Kumar | 3 | 30 |
| RK17EBA04 | Yash Kumar Duhan | 4 | 31 |
| RK17EBA05 | Amil Muhammed Hamza | 5 | 32 |
| RK17EBA06 | Abhinabh Parida | 6 | 33 |
| RK17EBA07 | Deepak Saraswat | 7 | 34 |
| RK17EBA08 | Vishal Batra | 8 | 35 |
| RK17EBA09 | Bolusani Santhosh | 9 | 36 |
| RK17EBA10 | Abhishek Sharma | 10 | 37 |
| RK17EBA11 | Arun Kumar Yadav | 11 | 38 |
| RK17EBA12 | Shamim Ahmed | 12 | 39 |
| RK17EBA13 | Mitanshu Baranwal | 13 | 40 |
| RK17EBA14 | Abhijeet Kumar | 14 | 41 |
| RK17EBA15 | Sandeep Kumar | 15 | 42 |
| RK17EBA16 | Anurag Srivastava | 16 | 43 |
| RK17EBA17 | Sahid Anowar Hussain | 17 | 44 |
| RK17EBA18 | Vikash Kumar Gupta | 18 | 45 |
| RK17EBA19 | Pratik  Sawalkar | 19 | 46 |
| RK17EBA20 | Motiranjan Nayak | 20 | 47 |
| RK17EBA21 | Akhil Singh | 21 | 48 |

| | | | |
|---|---|---|---|
| RK17EBA22 | Abhishek | 22 | 49 |
| RK17EBA23 | Velpula Hemanth Kumar | 23 | 50 |
| RK17EBA24 | Rotta Yosep Deven Kumar | 24 | 51 |
| RK17EBA25 | Lakkireddy Leelakar Reddy | 25 | 52 |
| RK17EBA26 | Budugu Sunil Kumar Reddy | 26 | 53 |
| RK17EBA27 | Mrityunjay Mishra | 27 | 54 |
| RK17EBA28 | Rishabh Kumar Tiwari | 1 | 28 |
| RK17EBA29 | Deepanshu Sharma | 2 | 29 |
| RK17EBA30 | Vikas Bunkar | 3 | 30 |
| RK17EBA31 | Rajesh Gupta | 4 | 31 |
| RK17EBA32 | Gaurav Singhal | 5 | 32 |
| RK17EBA33 | Pidikiti Sasidhar | 6 | 33 |
| RK17EBA34 | Vikas Munna Sharma | 7 | 34 |
| RK17EBB35 | Anurag Giri | 8 | 35 |
| RK17EBB36 | Mirthipati Kesavarao | 9 | 36 |
| RK17EBB37 | Manish Kumar | 10 | 37 |
| RK17EBB38 | Siddharth Gupta | 11 | 38 |
| RK17EBB39 | Yallapu Sai Ankit | 12 | 39 |
| RK17EBB40 | Vishal Sharma | 13 | 40 |
| RK17EBB41 | Pranav Sharma | 14 | 41 |
| RK17EBB42 | Vishal Das | 15 | 42 |
| RK17EBB43 | Soumyajyoti Mahalanobish | 16 | 43 |
| RK17EBB44 | Yuvraj Singh | 17 | 44 |
| RK17EBB45 | Manjeet Singh Maan | 18 | 45 |
| RK17EBB46 | Shalini Kumari Rai | 19 | 46 |
| RK17EBB47 | Sandeep Kaushik | 20 | 47 |
| RK17EBB48 | Gundeti Chinnesh | 21 | 48 |
| RK17EBB49 | Adnan Khan | 22 | 49 |
| RK17EBB50 | Ankit Sharma | 23 | 50 |
| RK17EBB51 | Rishabh Sharma | 24 | 51 |
| RK17EBB52 | Gaade Rakesh Reddy | 25 | 52 |
| RK17EBB53 | Battu Mercy Anns | 26 | 53 |
| RK17EBB54 | Simran Gujrati | 27 | 54 |
| RK17EBB55 | Arnav | 1 | 28 |
| RK17EBB56 | Nikhil Kumar Singh | 2 | 29 |
| RK17EBB57 | Abdullah | 3 | 30 |
| RK17EBB58 | Kishor Mitra | 4 | 31 |
| RK17EBB59 | Pragya Srivastava | 5 | 32 |
| RK17EBB60 | Arjun Chaudhary | 6 | 33 |
| RK17EBB61 | Anmol Singh | 7 | 34 |
| RK17EBB62 | Inturi Ganesh | 8 | 35 |

| | | | |
|---|---|---|---|
| RK17EBB63 | Appidi Moni | 9 | 36 |
| RK17EBB64 | Shrikant Prasad | 10 | 37 |
| RK17EBB65 | Pittala Deepak | 11 | 38 |
| RK17EBB66 | Deepesh Kumar Pandey | 12 | 39 |
| RK17EBB67 | Shruti Pandey | 13 | 40 |
| RK17EBB68 | Gaurav Barde | 14 | 41 |