VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Fall Semester, 2020-21

FAT Lab QP

Programme and Branch: B.Tech – CSE **Max. Marks: 50**

Course: CSE2005-Operating Systems **Duration: 1:30Hrs**

Slot: L41+L42

Instructions:

1. Write down your name, reg.no, slot and date in the answer sheet.

2. Write down aim ,algorithm in the paper.

4. Start implementing the given question.

5. Save the file with **regno** and upload the same in VTOP on or before 7 p.m.

6. **Upload file must contain the following:**

- Scanned copy of handwritten aim, algorithm
- Copy of the implemented source code with the output screenshot

NOTE:

Write a generic code and the code must run for any values of input. Also a write a code to perform validation of the data.

1) a) Implement menu-driven program to implement FCFS and SSTF disk scheduling algorithms. Take total number of tracks (or cylinders) and current head position as run-time input. Assume head is moving towards 0. Compute and display seek length and average seek length.

b) Consider three processes, all arriving at time zero, with total execution time of 10, 20 and 30 units respectively. Each process spends the first 20% of execution time doing I/O, the next 70% of time doing computation, and the last 10% of time doing I/O again. The operating system uses a shortest remaining compute time first scheduling algorithm and schedules a new process either when the running process gets blocked on I/O or when the running process finishes its compute burst. Assume that all I/O operations can be overlapped as much as possible. Develop an algorithm and write code to find out the percentage of the PU remain idle.

2) a)Implement menu-driven program for page replacement using MFU and LFU. Consider the total number of references made by CPU are 16. Take page reference string and number of page frames as run time input. Compute and display number of page faults, hit ratio and miss ratio. Use stack data structure to record most recent page references. Assume initially all 3 page frames are empty. Display stack content and number of page faults.

b)Implement Banker's Algorithm with additional resource request checking to grant the request or not.

Consider the snapshot of the system as given below:

Process	Allocation				Max				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	2	0	1	1	3	2	1	1	2	2	2	0
P1	1	1	0	0	1	2	0	2				
P2	1	0	1	0	3	2	1	0				
P3	0	1	0	1	2	1	0	1				

i) If a request from process P3 arrives for additional resources of (2,1,0,0), can the Banker's algorithm grant the request immediately?

3) a)Implement menu-driven program to implement Circular-SCAN and Circular-LOOK disk scheduling algorithms. Take total number of tracks (or cylinders) and current head position as run-time input. Assume head is moving towards track (or cylinder) 0 and provides service in the direction of decreasing number. Compute and display seek length and average seek length.

   b)Consider five CPU-intensive processes, which require 6, 9 ,10 , 4 and 5 time units and arrive at times 0, 3, 5,7 and 10, respectively with time quantum of 4 time units. Develop an algorithm and write code to find out the number of context switches, average turnaround time, average waiting time.

4) a)Implement menu-driven program using Optimal and FIFO. Consider CPU creates 20 page references. Take page reference string and number of page frames as run time input. Compute and display number of page faults with respect to 4 and 5 frames, hit ratio and miss ratio. Use stack data structure to record most recent page references. Assume initially all page frames are empty. Display stack content and number of page faults.

   b)Consider six processes are arrived the ready queue at times 0, 4, 5, 6 , 8 and 9 which require 7, 4, 9, 1 , 8 and 10 units of time to complete their execution with the priority of 4, 1, 3 ,4, 5 and 2. Develop an algorithm and write code to find out the average turnaround time, average waiting time and percentage of the CPU remain idle for the non-preemptive priority scheduling algorithm.

5) a)Implement menu-driven program for page replacement using FIFO. Consider the total number of references made by CPU are 20. Take page reference string and number of page frames as run time input. Assume initially all page frames are empty. Display the number of page faults with respect to 3 and 4 frames and check for Belady's Anomaly. Display whether Belady's Anomaly exist or not.

b)Consider three process, all arriving at time zero, with total execution time of 10, 20 and 30 units respectively. Each process spends the first 20% of execution time doing I/O, the next 70% of time doing computation, and the last 10% of time doing I/O again. The operating system uses a shortest remaining compute time first scheduling algorithm and schedules a new process either when the running process gets blocked on I/O or when the running process finishes its compute burst. Assume that all I/O operations can be overlapped as much as possible. Develop an algorithm and write code to find out the percentage of the CPU remain idle.

6) a) Implement menu-driven program to implement FCFS and SSTF disk scheduling algorithms. Take total number of tracks (or cylinders) and current head position as run-time input. Assume head is moving towards 0. Compute and display seek length and average seek length.

b)Consider four processes are arrived the ready queue at times 0, 1, 2 and 3 which require 8, 4, 9 and 5 time units to complete their execution. Develop an algorithm and write code to find out the number of context switches are needed, average turnaround time and average waiting time for the shortest job first remaining algorithm.

7) a)Implement menu-driven program for page replacement using MFU and LFU. Consider the total number of references made by CPU are 16. Take page reference string and number of page frames as run time input. Compute and display number of page faults, hit ratio and miss ratio. Use stack data structure to record most recent page references. Assume initially all 3 page frames are empty. Display stack content and number of page faults.

b)Write a program that should combine the Shortest Job First and Priority Scheduling by a logic also show it with arrival time variance in non-pre-emptive mode.

8) a)Implement menu-driven program to implement Circular-SCAN and Circular-LOOK disk scheduling algorithms. Take total number of tracks (or cylinders) and current head position as run-time input. Assume head is moving towards track (or cylinder) 0 and provides service in the direction of decreasing number. Compute and display seek length and average seek length.

b)Write a shell script that computes the gross salary of an employee according to the following rules

i. if basic salary is < 1500 then HRA=10% of the basic salary and DA=90% of basic
ii. if basic salary is >=1500 then HRA=Rs 500 and DA=98% of basic The basic salary is entered interactively through the keyboard

9) a)Implement menu-driven program using Optimal and FIFO. Consider CPU creates 20 page references. Take page reference string and number of page frames as run time input. Compute and display number of page faults with respect to 4 and 5 frames, hit ratio and miss ratio. Use stack data structure to record most recent page references. Assume initially all page frames are empty. Display stack content and number of page faults.

b)Banker's Algorithm with additional resource request checking to grant the request or not.

Consider the snapshot of the system as given below:

Process	Allocation				Max				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	2	0	1	1	3	2	1	1	2	2	2	0
P1	1	1	0	0	1	2	0	2				
P2	1	0	1	0	3	2	1	0				
P3	0	1	0	1	2	1	0	1				

i) If a request from process P3 arrives for additional resources of (2,1,0,0), can the Banker's algorithm grant the request immediately?

10) a) Implement menu-driven program for page replacement using FIFO. Consider the total number of references made by CPU are 20. Take page reference string and number of page frames as run time input. Assume initially all page frames are empty. Display the number of page faults with respect to 3 and 4 frames and check for Belady's Anomaly. Display whether Belady's Anomaly exist or not.

b)Consider six processes are arrived the ready queue at times 0, 4, 5, 6 , 8 and 9 which require 7, 4, 9, 1 , 8 and 10 units of time to complete their execution with the priority of 4, 1, 3 ,4, 5 and 2. Develop an algorithm and write code to find out the average turnaround time, average waiting time and percentage of the CPU remain idle for the non-preemptive priority scheduling algorithm.