LOVELY PROFESSIONAL UNIVERSITY
Academic Task-3 (Operating System)

School of Computer Science and Engineering Faculty of Technology And Sciences

Name of the faculty member Shivali Chopra

Course Code: CSE 316 Course Title: Operating System

Term: 18192
Max. Marks:30 Date of Submission: 10/04/2019

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:

Roll Number of Students:

S.No.	Roll No.	Ques. No. Assigned
1	1,14,27,40,53,66	1, 26
2	2,15,28,41,54,67	2, 25
3	3,16,29,42,55	3, 24
4	4,17,30,43,56	4,23
5	5,18,31,44,57	5, 22
6	6,19,32,45,58	6, 21
7	7,20,33,46,59	7, 20
8	8,21,34,47,60	8, 19
9	9,22,35,48,61	9, 18
10	10,23,36,49,62,69	10,17
11	11,24,37,50,63	11,14
12	12,25,38,51,64	12,15
13	13,26,39,52,65,68	13,16

List of Questions:

1. Write a C program using the fork() system call that generates this sequence in the child process. The starting number will be provided from the user. For example, if 8 is passed as a parameter on the command line, the child process will output 8, 4, 2, 1. Because the parent and child processes have their own copies of the data, it will be necessary for the child to output the sequence. Have the parent invoke the wait() call to wait for the child process to complete before exiting the program. Perform necessary error checking to ensure that a positive integer is passed on the command line.

2. Write a multithreaded program that calculates various statistical values for a list of numbers. This program will be passed a series of numbers on the command line and will then create three separate worker threads. One thread will determine the average of the numbers, the second will determine the maximum value, and the third will determine the minimum value. For example, suppose your program is passed the integers
   90 81 78 95 79 72 85
   The program will report
   The average value is 82
   The minimum value is 72
   The maximum value is 95

   The variables representing the average, minimum, and maximum values will be stored globally. The worker threads will set these values, and the parent thread will output the values once the workers have exited.

3. Write a multithreaded program in C that outputs prime numbers. This program should work as follows: The user will run the program and will enter a number on the command line. The program will then create a separate thread that outputs all the prime numbers less than or equal to the number entered by the user.
4. The Fibonacci sequence is the series of numbers 0, 1, 1, 2, 3, 5, 8, ... Formally, it can be expressed as:

$fib0 = 0$
$fib1 = 1$
$fibn = fibn-1 + fibn-2$

Write a multithreaded program that generates the Fibonacci sequence. This program should work as follows: On the command line, the user will enter the number of Fibonacci numbers that the program is to generate. The program will then create a separate thread that will generate the Fibonacci numbers, placing the sequence in data that can be shared by the threads (an array is probably the most convenient data structure). When the thread finishes execution, the parent thread will output the sequence generated by the child thread. Because the
parent thread cannot begin outputting the Fibonacci sequence until the child thread finishes, the parent thread will have to wait for the child thread to finish.

5. 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.

6. Suppose that the following processes arrive for execution at the times indicated. Each process will run for the amount of time listed. In answering the questions, use nonpreemptive scheduling, and base all decisions on the information you have at the time the decision must be made.

   Process Arrival Time Burst Time

   *P*1 0.0 8

   *P*2 0.4 4

   *P*3 1.0 1

   a. What is the average turnaround time for these processes with the FCFS scheduling algorithm?

   b. What is the average turnaround time for these processes with the SJF scheduling algorithm?

   c. Compute what average turnaround time will be if the CPU is left idle for the first 1 unit and then SJF scheduling is used. Remember that processes *P*1 and *P*2 are waiting during this idle time, so their waiting time may increase.

7. Researchers designed one system that classified interactive and noninteractive processes automatically by looking at the amount of terminal I/O. If a process did not input or output to the terminal in a 1-second interval, the process was classified as noninteractive and was moved to a lower-priority queue. In response to this policy, one programmer modified his programs to write an arbitrary character to the terminal at regular intervals of less than 1 second. The system gave his programs a high priority, even though the terminal output was completely meaningless.

8. The following 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.

Thread Priority Burst Arrival

*P*1 40 20 0

*P*2 30 25 25

*P*3 30 25 30

*P*4 35 15 60

*P*5 5 10 100

*P*6 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?

9. 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 intergers 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 turn around time. Consider used scheduling algorithm as SJF and same arrival time for all the processes.

10. 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?

11. Write a C program to solve the following problem:

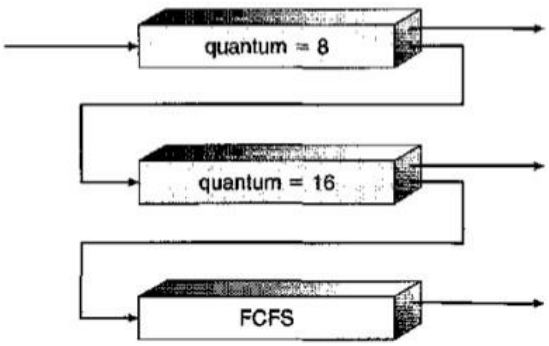
Suppose that a disk drive has 5,000 cylinders, numbered 0 to 4999. The drive is currently serving a request at cylinder143, and the previous request was at cylinder 125. The queue of pending requests, in FIFO

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?

12. Implement the multi-level feedback queue scheduling algorithm by considering the following diagram: You can use the code of others to implement Roud-Robin, and FCFS but implement aging by your own self.



13. 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.

14. 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. Ensure in your program that the waiting time of students is minimized.

15. 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.

16. 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 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.
17. 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 (Queue1), 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.

18. 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 at same time.

19. 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?

20. 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,an other 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 another 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 synchronize the teacher and the students.

21. 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.

22. 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.
    (For any reference refer Text Book)
23. 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.
24. Consider following and Generate a solution in C 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

25. Design a program using concepts of inter-process communication ordinary pipes in which one process sends a string message to a second process, and the second process reverses the case of each character in the message and sends it back to the first process. For example, if the first process sends the message Hi There, the second process will return hI tHERE. This will require using two pipes, one for sending the original message from the first to the second process and the other for sending the modified message from

the second to the first process. You can write this program using either UNIX or Windows pipes.

26. Design a file-copying program named filecopy using ordinary pipes. This program will be passed two parameters: the name of the file to be copied and the name of the copied file. The program will then create an ordinary pipe and write the contents of the file to be copied to the pipe. The child process will read this file from the pipe and write it to the destination file. For example, if we invoke the program as follows:
    filecopy input.txt copy.txt
    The file input.txt will be written to the pipe. The child process will read the contents of this file and write it to the destination file copy.txt.
27. Write a C program to create two zombie processes and two orphan processes using system calls.