OS Assignment 1

1.

Show the execution of the following processes on Gantt Charts for the processor and the I/O device if Non-preemptive Shortest Job First Algorithm for the Processor is used. Assume that the processes in the I/O queues are processed in First Come First Served manner. Find also the average waiting time in ready queue.

CPU and I/0 Bursts

Arrival	Process	CPU Burst	I/O Burst	CPU Burst	I/O Burst	CPU Burst
0	A	4	4	4	4	4
2	B	8	1	8	1	-
3	C	2	1	2	1	-
7	D	1	1	1	1	1

2.

Which algorithm will be suitable for CPU scheduling in the following types of Operating systems? Why?

i) Batch OS ii) Interactive OS iii) Real time OS

What is the purpose of a Command line interpreter? Why is it usually separated from kernel?

What happens on a context switch? Should context switches happen frequently or infrequently? Justify.

What is blocked suspended and ready suspended state of a process? Why these states are needed?

3.

Draw the process state transition diagram of an OS in which i) each process is in one of the five states: created, ready, running, blocked (i.e. sleep or wait), or terminated, and ii) only non-preemptive scheduling is used by the OS. Explain the state transitions.

The table below is a representation of the OS's internal process table. Each process has a stack, PID, a status, a priority and a next field. The next field is used to store the index of the process table using two circular linked list. One list contains all the ready and running processes (0 -> 4 > 3 -> 2 ->0) and another list contains all the blocked processes (1->1). Assume that a higher priority value means a higher priority.

	Stack	PID	Status	Priority	Next	CPU Burst
0	Ptr	4	Ready	1	4	33
1	Ptr	5	Blocked	10	1	134
2	Ptr	7	Running	7	0	58
3	Ptr	11	Ready	5	2	150
4	Ptr	13	Ready	8	3	145

a) Suppose that round-robin scheduling is being used. Which process would run after the current time slice expires?

b) Suppose that priority scheduling is being used and process 7 makes a blocking request. Which process would run next?

c) Suppose that process 7 makes a blocking request and round robin scheduling is being used. How would the table look like?

d) Suppose that while process 7 is running, process 5's blocking request is satisfied and it is no longer blocking. How would the process table look like (just before the next process context switch)?

e) Suppose a new process with PID 24 and priority 6 is created. How would the process table look like?

4.

Consider the implementation of all the short-term scheduling algorithms.

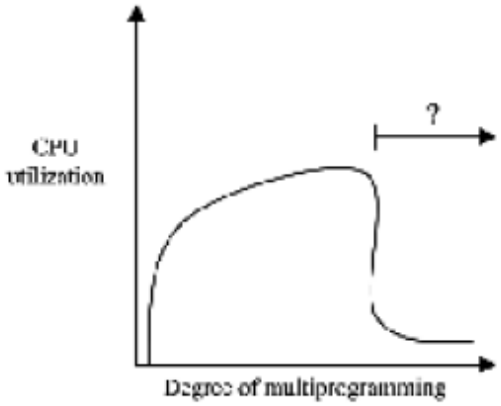
(i) Which algorithms are effectively impossible to implement? Explain why?

(ii) Which algorithms require a timer interrupt for the CPU?

(iii) Why RR is the only real scheduling algorithm that can be used in practice?

5.

Consider the graph given below that shows a plot of "degree of multiprogramming" on X-axis versus "CPU utilization" on Y-axis. What is the phenomenon observed by a sudden dip in the graph (as shown with the question mark) called? How can it be prevented?



What is meant by context switching? Why are context switches desirable?

6.

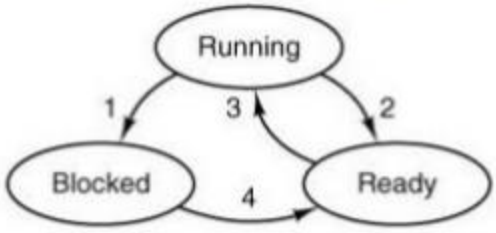
Consider a new scheduling algorithm that gives the highest priority to processes that have just entered the system, but is fair to all processes. The algorithm works like this: There are two queues, one for new processes and one for old processes. When a process enters the system, it is put at the end of the new queue. After 2 milliseconds on the new queue, whether a process has been scheduled or not, it is moved to the end of the old queue. When it is time to schedule a process, the system schedules the process at the head of one of the queues, alternating between the two queues. Each process runs to completion before the next process is scheduled. Assume that processes enter the system at random times and that most processes take much longer than 2 milliseconds to execute.

- Does this algorithm give the highest priority to new processes?
- Is this algorithm starvation free?
- Whether this algorithm is fair to all processes. By "fair" we mean every process has a wait time approximately equal to the average wait time, assuming all processes have close to the same execution time.

7.

How does the mixture of I/O bound processes with CPU bound processes maximize system utilization? Why is this more important in batch systems than in multiprocessing systems?

In the following graph you can see the three scheduling states a process can be in. Write an example event that triggers each transition.



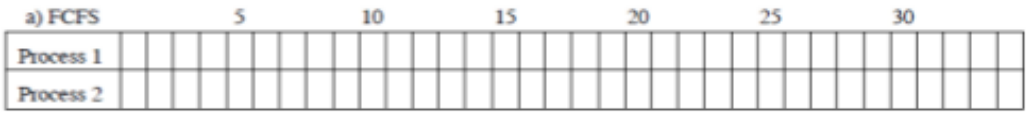
8.

c) Consider two processes, each with two CPU bursts with one I/O burst in between. Process 1 has a CPU burst of 9 units followed by an I/O burst of 7 units and a CPU burst of 6 unit.Process 2 has a CPU burst of 2 units followed by an I/O burst of 1 units and a CPU burst of 5 units.Suppose that Process 1 arrives in the ready queue just before Process 2 and just after Process 2 arrives the process that was in the CPU terminates. No other processes are in the system. For each of the scheduling algorithms below create Ganttcharts as given below. Fill each box with the state of the corresponding process. Use R for running, W for waiting/blocked, and D for ready. Calculate the waiting times and CPU utilization (as a fraction) for each process and fill in the table below.

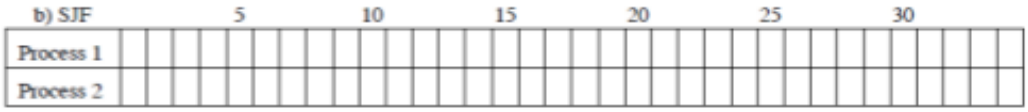
- First-Come/First-Served
- Shortest Job First (non-preemptive)
- Preemptive Shortest Job First
- Round Robin with a quantum of 3.

Gantt Charts:

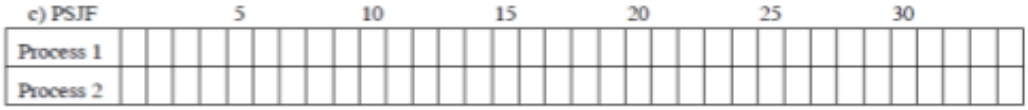
a) FCFS



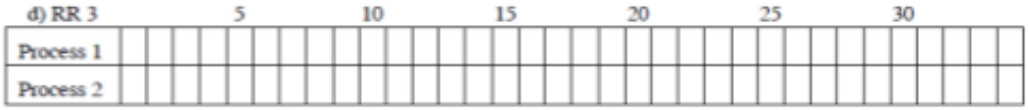
b) SJF



c) PSJF



d) RR 3



Algorithm	CPU Utilization	Waiting Time			Time Finished	
		Process 1	Process 2	Average	Process 1	Process 2
a) FCFS						
b) SJF						
c) PSJF						
d) RR 3						