UNIVERSITY OF BIRMINGHAM

M.Sc. Computer Science
Computer Systems

Additional Questions on Week 6 Materials – Part II

Question #1: What is the function of the ready queue?

The ready queue is a queue of processes in the READY state of the five state process model. A process will enter the READY queue when it may be executed without waiting for a resource. The queue exists to establish a fair and efficient order for processes to be executed.

Question #2: What is the difference between preemptive scheduling and non-preemptive scheduling? What is the issue with the latter?

Preemptive scheduling is based on timer interrupts, where a running thread may be interrupted by the OS and switched to the ready state at will (usually if something more important comes through) or when it has exceeded its timing allocation. Non-preemptive scheduling means once a thread is in the running state, it continues until it completes, or at least gives up the CPU voluntarily. Threads that do this are likely to monopolize the CPU.

Question #3: 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 non-preemptive 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?

10.53

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

9.53

c) The SJF algorithm is supposed to improve performance, but notice that we chose to run process P1 at time 0 because we did not know that two shorter processes would arrive soon. Compute what the 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 P1 and P2 are waiting during this idle time, so their waiting time may increase. This algorithm could be known as future-knowledge scheduling.

6.86

Question #4: The traditional UNIX, the scheduler enforces an inverse relationship between priority numbers and priorities: the higher the number, the lower the priority. The scheduler recalculates process priorities once per second using the following function:

$$\text{Priority} = (\text{recent CPU usage} / 2) + \text{base}$$

where base = 60 and recent CPU usage refers to a value indicating how often a process has used the CPU since priorities were last recalculated. Assume that recent CPU usage for process P_1 is 40, for process P_2 is 18, and for process P_3 is 10. What will be the new priorities for these three processes when priorities are recalculated? Based on this information, does the traditional UNIX scheduler raise or lower the relative priority of a CPU -bound process?

The priorities assigned to the processes are 80, 69, and 65 respectively. The scheduler lowers the relative priority of CPU -bound processes.

Question #5: Describe round robin scheduling. What is the parameter associated with the scheduler? What is the issue in choosing the parameter?

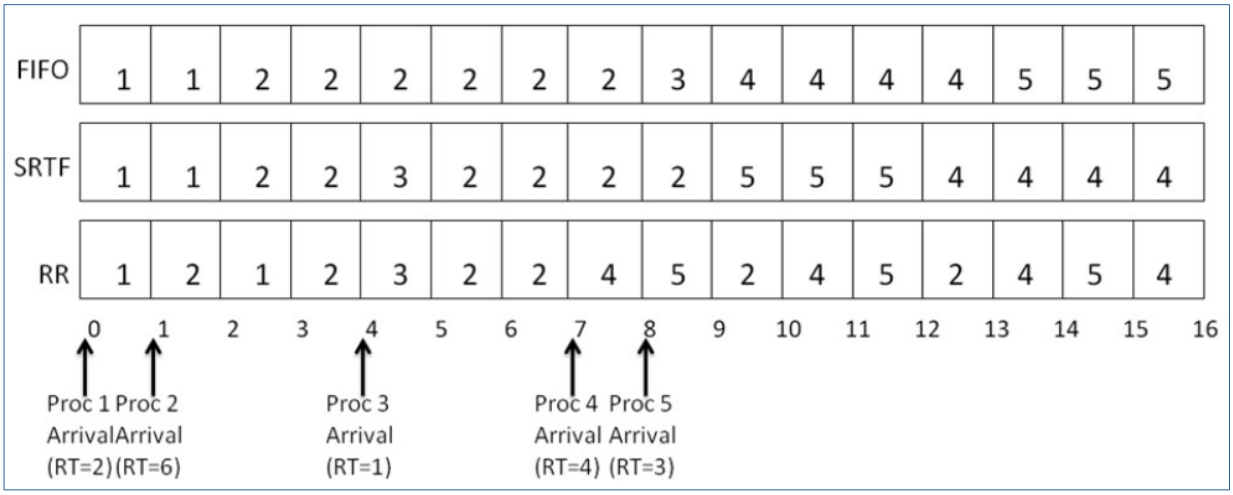
Round-robin scheduling works by giving each process a “timeslice” to run in, implemented by a ready queue and a regular timer interrupt. When a timeslice expires, the next process preempts the current process and runs for its timeslice, putting the preempted process at the end of the queue. The parameter concerned with it is the timeslice, which has to be exactly the right size. If it was too short, there is a large overhead every time it expires and the context switches, if it was too long, then we might end up with an unresponsive system.

Question #5: Consider the following table of processes and their associated arrival and running times.

Process ID	Arrival Time (AT)	Running Time (RT)
1	0	2
2	1	6
3	4	1
4	7	4
5	8	3

(a) Show the scheduling order for these processes under 3 different policies: First Come First Serve (FCFS), Shortest-Remaining-Time-First (SRTF), Round-Robin (RR) with a quantum size = 1, by filling in the Gantt chart with ID of the process currently running in each time quantum. Assume that context switch overhead is 0 and that new RR processes are added to the head of the queue and new FCFS processes are added to the tail of the queue.

Solution:



(b) Compute the response time for each process in each schedule above & fill in the following table:

Scheduler	Process 1	Process 2	Process 3	Process 4	Process 5
FIFO					
SRTF					
RR					

Solution:

Scheduler	Process 1	Process 2	Process 3	Process 4	Process 5
FIFO	0 – 0 = 0	2 – 1 = 1	8 – 4 = 4	9 – 7 = 2	13 – 8 = 5
SRTF	0 – 0 = 0	2 – 1 = 1	4 – 4 = 0	12 – 7 = 5	9 – 8 = 1
RR	0 – 0 = 0	1 – 1 = 0	4 – 4 = 0	7 – 7 = 0	8 – 8 = 0