**Q0**) What are the possible state transitions of a process?

=>> The process can be in any one of the following three possible states. 1) Running (actually using the CPU at that time and running). 2) Ready (runnable; temporarily stopped to allow another process run). 3) Blocked (unable to run until some external event happens).

**Q1**) What are the differences between a thread and a process?

=>> Process means a program is in execution, whereas thread means a segment of a process. A Process is not Lightweight, whereas Threads are Lightweight. A Process takes more time to terminate, and the thread takes less time to terminate. Process takes more time for creation, whereas Thread takes less time for creation.

**Q2**) What is a race condition?

=>> A race condition is an undesirable situation that occurs when a device or system attempts to perform two or more operations at the same time, but because of the nature of the device or system, the operations must be done in the proper sequence to be done correctly.

**Q3**) Five jobs are waiting to be run. Their expected run times are 9, 6, 3, 5, and *X*. In what order should they be run to minimize average response time? Given X = 10 and X = 1

=>>0<X<=3:X,3,5,6,93<X<=5:3,X,5,6,95<X<=6:3,5,X,6,96<X<=9:3,5,6,X,9X>9:3,5,6,9,X

**Q4**) Five batch jobs *A* through *E*, arrive at a computer center at almost the same time. They have estimated running times of 10, 6, 2, 4, and 8 minutes. Their (externally determined) priorities are 3, 5, 2, 1, and 4, respectively, with 5 being the highest priority. For each of the following scheduling algorithms, determine the mean process turnaround time.

(a) Round robin (RR=4).

(b) Priority scheduling.

(c) First-come, first-served (run in order 10, 6, 2, 4, 8).

(d) Shortest job first.

For (a), assume that the system is multiprogrammed, and that each job gets its fair share of the CPU. For (b) through (d) assume that only one job at a time runs, until it finishes. All jobs are completely CPU bound.

=>> The burst times and priorities of the processes are:

|  |  |  |
| --- | --- | --- |
| **Process** | **Burst Times** | **Priorities** |
| **A** | 6 | 3 |
| **B** | 4 | 5 |
| **C** | 1 | 2 |
| **D** | **3** | **1** |
| **E** | **7** | **4** |

1. RR with quantum = 1
2. So the average waiting time is given by:

|  |  |
| --- | --- |
| Twait | = ( (0+4+3+3+2+1) + (1+4+3+3) + (2) + (3+3+3) + (4+3+3+2+1+1+0) / 5 |
|  | = (13 + 11 + 2 + 9 + 14) /5 |
|  | = 49 / 5 |
|  | = **9.8** ( minutes ) |

1. b) Priority:  
   The processes are scheduled in the order: B E A C D  
   Similarly, the waiting time for A B C D E are: 11, 0, 17, 18, 4 respectively.  
   So the average waiting time is;

|  |  |
| --- | --- |
| Twait | = (11 + 0 + 17 + 18 + 4) /5 |
|  | = 50 / 5 |
|  | = **10** ( minutes ) |

1. c) FCFS:  
   The processes are scheduled in the order: A B C D E

|  |  |
| --- | --- |
| Twait | = (0 + 6 + (6+4) + (6+4+1) + (6+4+1+3)) /5 |
|  | = (0 + 6 + 10 + 11 + 14) /5 |
|  | = 41 / 5 |
|  | = **8.2** ( minutes ) |

1. d) SJF:  
   The processes are scheduled in the order: C D B A E

|  |  |
| --- | --- |
| Twait | = ( (1+4+3) + (1+3) + 0 + (1) + (1+4+3+6) )/ 5 = (8 + 4 + 0 + 1 + 14) /5 |
|  | = 27 / 5 |
|  | = **5.4** ( minutes ) |

**Q5)** What is the difference between preemption and non-preemption in the context of process scheduling.

=>> Key Differences Between Preemptive and Non-Preemptive Scheduling: In preemptive scheduling, the CPU is allocated to the processes for a limited time whereas, in Non-preemptive scheduling, the CPU is allocated to the process till it terminates or switches to the waiting state