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

process?

- Tiến trình có thể ở bất kỳ một trong ba trạng thái sau

đây:

+ Running ( Đang sử dụng CPU tại thời điểm đó và đang

chạy).

+ Ready ( Tạm thời dừng lại lại để cho phép một tiến

trình khác chạy).

+ Blocked ( Không chạy cho đến khi có một sự kiện nào

đó xảy ra )

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

Định nghĩa:

+ Tiến trình: có nghĩa là một chương trình đang được thực

hiện.

+ Luồng: có nghĩa là một phân đoạn của một tiến trình.

- bộ nhớ :

+ Tiến trình: Nặng

+ Luồng: nhẹ hơn tiến trình nhiều

- Thời gian dừng:

+ Tiến trình: mất nhiều thời gian hơn để chấm dứt.

+ Luồng: mất ít thời gian hơn để chấm dứt.

- Thời gian tạo:

+ Tiến trình: Cần nhiều thời gian hơn

+ Luồng: mất ít thời gian hơn

- Tài nguyên:

+ Tiến trình: tiêu thụ nhiều tài nguyên hơn.

+ Luồng: tiêu thụ ít tài nguyên hơn.

- Chia sẽ:

+ Tiến trình: Nó không chia sẻ dữ liệu

+ Luồng: chia sẻ dữ liệu với nhau.

- Memory:

+ Tiến trình: chủ yếu bị cô lập

+ Luồng: chia sẻ bộ nhớ

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

For minimizing the average response time, the processes have to be executed according to the Shortest Job First.  
0<𝐗<=3:𝐗,3,5,6,9

3<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.

1. Round robin (RR=4).
2. 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** |

a) RR with quantum = 1

**Gnatt chart** is:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| A | B | C | D | E | A | B | D | E | A | B | D | E | A | B | E | A | E | A | E | E |  |

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 ) |

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 ) |

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 ) |

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.

|  |  |
| --- | --- |
| **Preemptive Scheduling** | **Non-Preemptive Scheduling** |
| The resources are assigned to a process for a long time period. | Once resources are assigned to a process, they are held until it completes its burst period or changes to the waiting state. |
| Its process may be paused in the middle of the execution. | When the processor starts the process execution, it must complete it before executing the other process, and it may not be interrupted in the middle. |
| When a high-priority process continuously comes in the ready queue, a low-priority process can starve. | When a high burst time process uses a CPU, another process with a shorter burst time can starve. |
| It is flexible. | It is rigid. |
| It is cost associated. | It does not cost associated. |
| It has overheads associated with process scheduling. | It doesn't have overhead. |
| It affects the design of the operating system kernel. | It doesn't affect the design of the OS kernel. |
| Its CPU utilization is very high. | Its CPU utilization is very low. |
| Examples: Round Robin and Shortest Remaining Time First | FCFS and SJF are examples of non-preemptive scheduling. |