**Tutorial 2: Processor Management**

Q1. (a) Describe the term “process” in the context of operating systems.

* A process is basically a program in execution. The execution of a process must be done in a specific order. A process is an entity that represents the fundamental unit of work that must be implemented in the system.
* Each process is represented by a PCB (Process control Block) repository for information.

(b) Elaborate how context switch enables multiple processes to share a single processor.

* Context switch is a kind of mechanism that will store and restore the context of a CPU in Process Control Block(PCB) so that a process execution can be resumed from the same point at a later time.

Q2. Differentiate between the followings.

1. I/O bound and CPU bound

|  |  |
| --- | --- |
| **I/O bound** | **CPU bound** |
| * I/O-bound job have many brief CPU cycle and long I/O cycles * Printing series of document | * CPU-bound jobs have long CPU cycles and shorter I/O cycles * Finding first 300 prime number |

1. Preemptive scheduling and non-preemptive scheduling

|  |  |
| --- | --- |
| **Preemptive scheduling** | **Non-preemptive scheduling** |
| * job can be interrupted by higher priority job | * cannot be interrupted by the other job |

Q3. (a) With the aid of a diagram, explain how a process may switch among the different states from its creation to termination.

|  |
| --- |
| * New : The process is in the stage of being created * Running : The CPU is working on this process's instructions. * Waiting : The process cannot run at the moment, because it is waiting for some resource to become available or for some event to occur. * Ready : The process has all the resources available that it needs to run, but the CPU is not currently working on this process's instructions. * Terminated : The process has completed |

1. Identify the event that will switch a process from running state to ready state and from running state to waiting state.

* Running --> Ready - Interruption occurs, which higher priority jobs comes in
* Running --> Waiting - I/O request occurs, For example, the process is changed to waiting state for keyboard input, writing to file or s

Q4. Identify the level (short-term, medium-term, long-term) of scheduler that would make a decision on each of the following situation. Justify your answer.

1. Which temporarily suspended process should be activated in order to balance the process mix?
   1. Medium term: determines when processes are to be suspended and resumed.
2. Which ready process should be assigned the CPU when it next becomes available?
   1. Short term: determines which of the ready processes can have CPU resources and for how long.
3. Which of a series of waiting batch jobs that have been spooled to disk should next be initiated?
   1. Long term: determines which programs are admitted to the system for execution and when and which one should be exited.

Q5. (a) Consider the following set of processes, with the length of the CPU burst given in milliseconds (ms). Ignore no context switch time.

|  |  |  |  |
| --- | --- | --- | --- |
| Process | Arrival Time | Execution Time | Priority |
| P1 | 0 | 3 | 1 |
| P2 | 2 | 6 | 2 |
| P3 | 3 | 2 | 3 |
| P4 | 6 | 4 | 3 |
| P5 | 9 | 3 | 4 |

Draw a **Gantt chart** illustrating the execution of the above processes, and calculate the **average turnaround time** and **average waiting time** based on each of the following algorithms:

TT = Finish – arrival

WT = TT – cpu cycle

1. Shortest Job First

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P1 | P3 | P2 | P5 | P4 |

0 3 5 11 14 18

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Process | Arrival Time | Execution Time | Priority | Finish time | Turnaround time | Waiting time |
| P1 | 0 | 3 | 1 | 3 | 3 | 0 |
| P2 | 2 | 6 | 2 | 11 | 9 | 3 |
| P3 | 3 | 2 | 3 | 5 | 2 | 0 |
| P4 | 6 | 4 | 3 | 18 | 12 | 8 |
| P5 | 9 | 3 | 4 | 14 | 5 | 2 |
|  |  |  |  | Avg | 6.2 | 2.6 |

1. Shortest Remaining Time First (SRTF)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P1 | P3 | P2 | P4 | P5 | P2 |

0 3 5 6 10 13 18

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Process | Arrival Time | Execution Time | Priority | Finish time | Turnaround time | Waiting time |
| P1 | 0 | 3 | 1 | 3 | 3 | 0 |
| P2 | 2 | 6 | 2 | 18 | 16 | 10 |
| P3 | 3 | 2 | 3 | 5 | 2 | 0 |
| P4 | 6 | 4 | 3 | 10 | 4 | 0 |
| P5 | 9 | 3 | 4 | 13 | 4 | 1 |
|  |  |  |  | Avg | 5.8 | 2.2 |

1. Preemptive Priority (Assume that a larger priority number implies a higher priority)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P2 | P4 | P5 | P4 | P2 | P1 |

0 2 3 5 6 9 12 13 17 18

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Process | Arrival Time | Execution Time | Priority | Finish time | Turnaround time | Waiting time |
| P1 | 0 | 3 | 1 | 18 | 18 | 15 |
| P2 | 2 | 6 | 2 | 17 | 15 | 9 |
| P3 | 3 | 2 | 3 | 5 | 2 | 0 |
| P4 | 6 | 4 | 3 | 13 | 7 | 3 |
| P5 | 9 | 3 | 4 | 12 | 3 | 0 |
|  |  |  |  | Avg | 9 | 5.4 |

1. Non Preemptive Priority (Assume that a smaller priority number implies a higher priority)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P4 | P5 |

0 3 9 11 15 18

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Process | Arrival Time | Execution Time | Priority | Finish time | Turnaround time | Waiting time |
| P1 | 0 | 3 | 1 | 3 | 3 | 0 |
| P2 | 2 | 6 | 2 | 9 | 7 | 1 |
| P3 | 3 | 2 | 3 | 11 | 8 | 6 |
| P4 | 6 | 4 | 3 | 15 | 9 | 5 |
| P5 | 9 | 3 | 4 | 18 | 9 | 6 |
|  |  |  |  | Avg | 7.2 | 3.6 |

1. Round Robin (Assume that time quantum = 3ms)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P4 | P2 | P5 | P4 |

0 3 6 8 11 14 17 18

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Process | Arrival Time | Execution Time | Priority | Finish time | Turnaround time | Waiting time |
| P1 | 0 | 3 | 1 | 3 | 3 | 0 |
| P2 | 2 | 6 | 2 | 14 | 12 | 6 |
| P3 | 3 | 2 | 3 | 8 | 5 | 3 |
| P4 | 6 | 4 | 3 | 18 | 12 | 8 |
| P5 | 9 | 3 | 4 | 17 | 8 | 5 |
|  |  |  |  | Avg | 8 | 4.4 |

1. State why strict non-preemptive scheduling is unlikely to be used in a computer center.
   1. It is because non-preemptive scheduling has higher waiting and response time that may cause the CPU utilization be less efficient. Besides, non-preemptive scheduling also will not switch process from ready state to running state or vice-versa based on the priority of process which mean non-preemptive scheduling will not prior handle higher priority job such as interrupt
2. Is round robin algorithm suitable for interactive systems? State **TWO (2)** reasons to support your answer.
   1. Yes suitable
   2. Fair resource allocation
   3. Similar respond time

(d) Complete the table below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Policy** | **Algorithm** | **Based on** | **Pros** | **Cons** |
| Non-preemptive | First Come First Serve (FCFS) | Arrival time | Batch system | Interactive system |
| Non-preemptive | Shortest Job First (SJN) | CPU cycle time | Job available at the same time | Interactive system |
| Non-preemptive / preemptive | Priority scheduling  (PS) | Priority level | Time critical processes | low priority job keeps waiting |
| preemptive | Shortest remaining time (SRT) | remaining execution time | Fast completion | Interactive system |
| preemptive | Round robin  (RR) | Time quantum | Fair resource allocation | More overhead |
| Wait Time = Finish Time – Arrival Time – Execution Time  Turnaround Time = Finish Time – Arrival Time | | | | |

**Self-Review**

Q1. Given the following metrics (Table 1) that show the arrival time, CPU burst time (in millisecond) and priority of a list of jobs.

**Table 1:** List of processes going to execute

|  |  |  |
| --- | --- | --- |
| **Process** | **Arrival Time** | **CPU Burst Time (ms)** |
| A | 0 | 10 |
| B | 3 | 5 |
| C | 5 | 3 |
| D | 7 | 8 |

Draw a Gantt chart to illustrate the execution of the above jobs, and calculate the average turnaround time and average waiting time based on each of the following algorithms:

1. Shortest Job Fist (SJF)

|  |  |  |  |
| --- | --- | --- | --- |
| A | C | B | D |

0 10 13 18 26

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Process** | **Arrival Time** | **CPU Burst Time (ms)** | **Finish Time** | **Turnaround time** | **Wait time** |
| A | 0 | 10 | 10 | 10 | 0 |
| B | 3 | 5 | 18 | 15 | 10 |
| C | 5 | 3 | 13 | 8 | 5 |
| D | 7 | 8 | 26 | 19 | 11 |
|  |  |  | AVG | 13 | 6.5 |

1. Round-Robin (RR) (Time Quantum = 4 ms)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | A | C | D | B | A | D |

0 4 8 12 15 19 20 22 26

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Process** | **Arrival Time** | **CPU Burst Time (ms)** | **Finish Time** | **Turnaround time** | **Wait time** |
| A | 0 | 10 | 22 | 22 | 12 |
| B | 3 | 5 | 20 | 17 | 12 |
| C | 5 | 3 | 15 | 10 | 7 |
| D | 7 | 8 | 26 | 19 | 11 |
|  |  |  | AVG | 17 | 10.5 |

Q2. Based on the solutions that you have given in ***Q1,*** do you agree that preemptive process scheduling has better performance than non-preemptive process scheduling? Justify your answer.

* In Q1 case, non-preemptive is better performance compare to preemptive process scheduling due to having a lower average wait time.

Q3. Consider the list of processes to be performed by processor shown in the **Table 2:**

**Table 2:** List of processes to be performed by processor

|  |  |  |
| --- | --- | --- |
| **Process** | **Arrival Time** | **CPU cycle** |
| A | 0 | 6 |
| B | 1 | 4 |
| C | 7 | 3 |
| D | 9 | 1 |

Draw a timeline for each of the following algorithms and calculate the *Average Turnaround Time* and *Average Waiting Time* for each of them:

1. Shortest Remaining Time First (SRTF)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | A | D | C |

0 1 5 10 11 14

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Process** | **Arrival Time** | **CPU Burst Time (ms)** | **Finish Time** | **Turnaround time** | **Wait time** |
| A | 0 | 6 | 10 | 10 | 4 |
| B | 1 | 4 | 5 | 4 | 0 |
| C | 7 | 3 | 14 | 7 | 4 |
| D | 9 | 1 | 11 | 2 | 1 |
|  |  |  | AVG | 5.75 | 2.25 |

1. Round Robin (Assume the time quantum is 3 time slices).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | B | A | B | C | D |

0 3 6 9 10 13 14

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Process** | **Arrival Time** | **CPU Burst Time (ms)** | **Finish Time** | **Turnaround time** | **Wait time** |
| A | 0 | 6 | 9 | 9 | 3 |
| B | 1 | 4 | 10 | 9 | 5 |
| C | 7 | 3 | 13 | 6 | 3 |
| D | 9 | 1 | 14 | 5 | 4 |
|  |  |  | AVG | 7.25 | 3.75 |