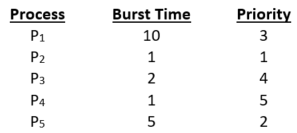
III\_SYT\_Reder\_Schröder

**Exercise: Process Scheduling**

1. Consider the following list of processes, with the length of CPU burst time given in milliseconds:   
   The processes are assumed to have arrived in the order P1, P2, P3, P4, P5 all at time 0.
2. Draw four Gantt charts that illustrate the execution of these processes using the following scheduling algorithms: FCFS, SJF, non-preemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 1).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **FCFS** | | | | | | | | | | | | | | | | | | |
| P1 | | | | | | | | | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  | P2 |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  | P3 | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  | P4 |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | P5 | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SJF** | | | | | | | | | | | | | | | | | | |
|  |  |  |  |  |  |  |  |  | P1 | | | | | | | | | |
| P2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | P3 | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | P4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | P5 | | | | |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **non-preemptive priority (low number = high priority)** | | | | | | | | | | | | | | | | | | |
|  |  |  |  |  |  | P1 | | | | | | | | | |  |  |  |
| P2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | P3 | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | P4 |
|  | P5 | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **RR (q = 1)** | | | | | | | | | | | | | | | | | | |
| P1 |  |  |  |  | P1 |  |  | P1 |  | P1 |  | P1 |  | P1 | P1 | P1 | P1 | P1 |
|  | P2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | P3 |  |  |  | P3 |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | P4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | P5 |  |  | P5 |  | P5 |  | P5 |  | P5 |  |  |  |  |  |

1. What is the turnaround time of each process for each of the scheduling algorithms in part a?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **b)** | **FCFS** | **SJF** | **Priority** | **RR** |
| P1 | 10 | 19 | 16 | 19 |
| P2 | 11 | 1 | 1 | 2 |
| P3 | 13 | 4 | 18 | 7 |
| P4 | 14 | 2 | 19 | 4 |
| P5 | 19 | 9 | 6 | 14 |

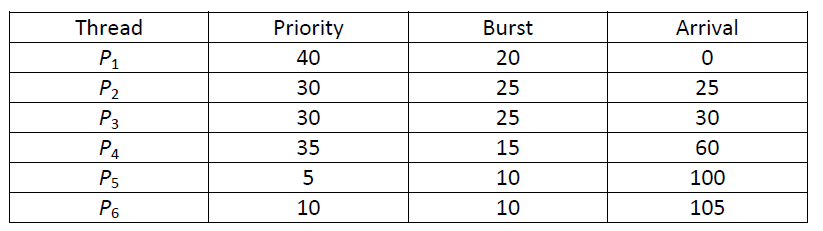
1. What is the waiting time of each process for each of these scheduling algorithms?

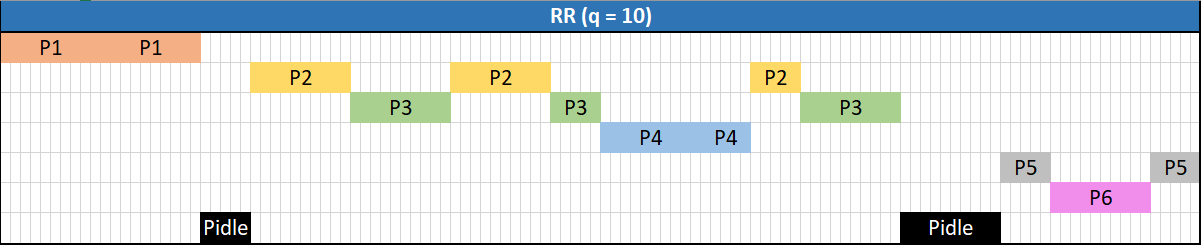
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **c)** | **FCFS** | **SJF** | **Priority** | **RR** |
| P1 | 0 | 9 | 6 | 9 |
| P2 | 10 | 0 | 0 | 1 |
| P3 | 11 | 2 | 16 | 5 |
| P4 | 13 | 1 | 18 | 3 |
| P5 | 14 | 4 | 1 | 9 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **d)** |  |  |  |  |
| **Summe:** | 48 | 16 | 41 | 27 |
| **Average:** | 9,6 | 3,2 | 8,2 | 5,4 |

1. Which of the algorithms results in the minimum average waiting time (over all processes)?

1. The following processes are being scheduled using a preemptive, round-robin scheduling algorithm, which in addition takes respect on process priorities. Each process is assigned a numerical priority, with a higher number indicating a higher relative priority. In addition to the processes listed below, the system also has an idle task (which consumes no CPU resources and is identified as Pidle). This task has priority 0 and is scheduled whenever the system has no other available processes to run. The length of a time quantum is 10 units. If a process is preempted by a higher-priority process, the preempted process is placed at the end of the queue.



1. Show the scheduling order of the processes using a Gantt chart.

1. What is the turnaround and the waiting time for each process?

|  |  |  |
| --- | --- | --- |
|  | **Turnaround** | **Waiting** |
| P1 | 20 | 0 |
| P2 | 55 | 30 |
| P3 | 60 | 35 |
| P4 | 15 | 0 |
| P5 | 20 | 10 |
| P6 | 10 | 0 |

1. What is the CPU utilization rate?  
   87,5%
2. Which of the following scheduling algorithms could result in starvation?
   1. First-come, first-served
   2. Shortest job first #
   3. Round robin
   4. Priority #

# Given a process. The previous four runs, from oldest to most recent, took 40, 20, 40, and 15 msec. What is the prediction of the next time if you take an initial estimate of 100 msec, with an ⍺ of 0.5?

|  |  |  |
| --- | --- | --- |
| **n** | **τ** | **t** |
| 0 | 100 | 40 |
| 1 | 70 | 20 |
| 2 | 45 | 40 |
| 3 | 42,5 | 15 |
| 4 | 28,75 |  |

# Consider the exponential average formula used to predict the length of the next CPU burst. What are the implications of assigning the following values to the parameters used by the algorithm?

* 1. α = 0 and τ0 = 100 milliseconds   
     The estimation of the algorithm will always be 100 msec.
  2. α = 0.99 and τ0 = 10 milliseconds   
     The estimation from the previous process will have almost no impact (1%) on the new estimated time.

1. Using the Windows scheduling algorithm, determine the numeric priority of each of the following threads
   1. A thread in the REALTIME\_PRIORITY\_CLASS with a relative priority of HIGHEST:  
      26
   2. A thread in the NORMAL\_PRIORITY\_CLASS with a relative priority of NORMAL:  
      8
   3. A thread in the HIGH\_PRIORITY\_CLASS wit a relative priority of ABOVE\_NORMAL:

14

1. Assuming that no threads belong to the REALTIME\_PRIORITY\_CLASS and that none may be assigned a TIME\_CRITICAL priority, what combination of priority class and priority corresponds to the highest possible relative priority in Windows scheduling?

15