Assignment 4:

Calculation of CPU Scheduling Algorithms

Duc M. Le

Thompson Rivers University

COMP 3411 – Operating System

Kevin O’Neil

October 1, 2021

# Question 1

First Come First Serve

|  |  |  |  |
| --- | --- | --- | --- |
| Process | Completion Time | Turnaround Time | Waiting Time |
| P1 | 16 | 16 | 0 |
| P2 | 26 | 26 | 16 |
| P3 | 31 | 28 | 23 |
| P4 | 39 | 34 | 26 |
| P5 | 51 | 43 | 31 |

Priority Non-Preemptive

|  |  |  |  |
| --- | --- | --- | --- |
| Process | Completion Time | Turnaround Time | Waiting Time |
| P1 | 34 | 34 | 18 |
| P2 | 10 | 10 | 0 |
| P3 | 39 | 36 | 29 |
| P4 | 18 | 13 | 5 |
| P5 | 51 | 43 | 31 |

Priority Preemptive

|  |  |  |  |
| --- | --- | --- | --- |
| Process | Completion Time | Turnaround Time | Waiting Time |
| P1 | 34 | 34 | 18 |
| P2 | 10 | 10 | 0 |
| P3 | 39 | 36 | 29 |
| P4 | 18 | 13 | 5 |
| P5 | 51 | 43 | 31 |

Round Robin

|  |  |  |  |
| --- | --- | --- | --- |
| Process | Completion Time | Turnaround Time | Waiting Time |
| P1 | 51 | 51 | 35 |
| P2 | 35 | 35 | 25 |
| P3 | 15 | 12 | 7 |
| P4 | 38 | 33 | 25 |
| P5 | 50 | 42 | 30 |

# Question 2

First Come First Serve

|  |  |  |  |
| --- | --- | --- | --- |
| Process | Completion Time | Turnaround Time | Waiting Time |
| P1 | 8 | 8 | 0 |
| P2 | 12 | 12 | 8 |
| P3 | 13 | 9 | 8 |

Priority Non-Preemptive

|  |  |  |  |
| --- | --- | --- | --- |
| Process | Completion Time | Turnaround Time | Waiting Time |
| P1 | 12 | 12 | 4 |
| P2 | 4 | 4 | 0 |
| P3 | 13 | 9 | 8 |

Priority Preemptive

|  |  |  |  |
| --- | --- | --- | --- |
| Process | Completion Time | Turnaround Time | Waiting Time |
| P1 | 12 | 12 | 4 |
| P2 | 4 | 4 | 0 |
| P3 | 13 | 9 | 8 |

Round Robin

|  |  |  |  |
| --- | --- | --- | --- |
| Process | Completion Time | Turnaround Time | Waiting Time |
| P1 | 13 | 13 | 5 |
| P2 | 9 | 9 | 5 |
| P3 | 10 | 6 | 5 |

# Question 3

Shortest Remaining Time

|  |  |  |  |
| --- | --- | --- | --- |
| Process | Completion Time | Turnaround Time | Waiting Time |
| P1 | 19 | 19 | 11 |
| P2 | 4 | 4 | 0 |
| P3 | 5 | 1 | 0 |
| P4 | 12 | 6 | 0 |

# Question 4

The priority scheduling algorithm suffers from starvation (aka infinite blocking) as it can leave some low-priority processes waiting indefinitely. In a heavily loaded computer system, a steady stream of higher-priority processes can prevent a low-priority process from ever getting the CPU.

A solution to the problem of indefinite blockage of low-priority processes is aging, which involves gradually increasing the priority of processes that wait in the system for a long time.