Assignment 4

**Task 1:**

Assumption:

In the event when a context switch and a new process is added to the Runnable queue the preference is given to the new process.

Complete the simulation for

Round Robin (RR)

average waiting time **8.88**

Full log at APPENDIX.1

Shortest remaining time first (SRTF) with given CPU burst time

average waiting time **4.50**

APPENDIX.2

Shortest job first (SJF) with future prediction

average waiting time **7.25**

APPENDIX.3

**Task 2:**

Observations:

Process 0 mostly has a large burst time, except once

Process 1 has balance of large and small burst times

Process 2 bust time is ascending though the time line

Process 3 similar to process 2 it is in ascending order

|  |  |  |
| --- | --- | --- |
| Process | Number of Short bust | Number of long bust |
| 0 | 1 | 3 |
| 1 | 2 | 2 |
| 2 | 1 | 3 |
| 3 | 1 | 3 |

Short bust = burst time < 5

Long bust = burst time > 5

So in our example we will treat Process 1 as a O/I bound process and the others as CPU bound processes.

Round Robin (RR)

TODO

Shortest remaining time first (SRTF) with given CPU burst time

TODO output

Shortest job first (SJF) with future prediction

TODO output

2. Which of the evaluated schemes in this assignment (Task 1.1-1.3, 2.1) generates the optimal schedule (gives minimum average waiting time) for a system with:

a) All short processes

Inputs

0 0 2

1 1 1

2 2 2

3 5 1

3 30 2

1 31 1

2 32 1

0 38 2

2 60 1

0 62 2

1 65 1

3 66 1

1 90 2

0 95 2

2 98 1

3 99 2

b) Very short and very long processes interleave each other with unpredictable pattern.

0 0 30

1 1 1

2 2 30

3 5 1

3 30 1

1 31 1

2 32 30

0 38 30

2 60 30

0 62 30

1 65 1

3 66 1

1 90 1

0 95 30

2 98 30

3 99 1

Process 1 and 3 are short process

Process 0 and 2 are long process

My scheduling algorithm i decide to use was Foreground-background

<https://en.wikipedia.org/wiki/Foreground-background>

3. State a different scheduling mechanism that has not been mentioned in the lecture notes and explain the intuition behind the using this scheme.

1. Assume your system has N CPU cores, and each process only requires burst time on 1 core. Will it make the scheduler more complicated? Suggest how to extend the current scheduler to multiprocessor system.

**APPENDIX**

1

(0, 0)

(2, 1)

(4, 2)

(6, 0)

(8, 1)

(10, 3)

(12, 0)

(14, 1)

(16, 0)

(18, 1)

(20, 0)

(30, 3)

(32, 1)

(34, 2)

(36, 3)

(38, 2)

(40, 0)

(42, 3)

(44, 2)

(46, 0)

(48, 0)

(50, 0)

(60, 2)

(62, 0)

(64, 2)

(66, 1)

(68, 3)

(70, 2)

(72, 1)

(74, 3)

(76, 2)

(78, 3)

(80, 3)

(90, 1)

(92, 1)

(94, 1)

(96, 0)

(98, 1)

(100, 2)

(102, 0)

(104, 3)

(106, 1)

(108, 2)

(110, 0)

(112, 3)

(114, 2)

(116, 0)

(118, 3)

(120, 2)

(122, 0)

(124, 3)

(126, 2)

average waiting time 8.88

2

(0, 0)

(2, 2)

(4, 0)

(5, 3)

(7, 0)

(13, 1)

(30, 3)

(31, 1)

(33, 3)

(37, 2)

(43, 0)

(60, 2)

(62, 0)

(64, 2)

(65, 1)

(68, 2)

(72, 3)

(90, 1)

(100, 3)

(108, 2)

(117, 0)

average waiting time 4.50

3

(0, 0)

(9, 1)

(17, 2)

(19, 3)

(30, 3)

(35, 2)

(41, 1)

(43, 0)

(60, 2)

(67, 3)

(75, 1)

(78, 0)

(90, 1)

(100, 2)

(109, 3)

(117, 0)

average waiting time 7.25