



CS5250 Assignment 4

A0174443E XU Yudong

Task1 1

Round Robing

```
(0, 0)
(2, 1)
(4, 2)
(6, 3)
(8, 0)
(10, 1)
(12, 0)
(14, 1)
(16, 0)
(18, 1)
(20, 0)
(30, 3)
(32, 1)
(34, 2)
(36, 3)
(38, 2)
(40, 0)
(42, 3)
(43, 2)
(45, 0)
(60, 2)
(62, 0)
(64, 2)
(66, 1)
(68, 3)
(70, 2)
(72, 1)
(73, 3)
(75, 2)
(76, 3)
(90, 1)
(96, 0)
(98, 2)
(100, 3)
(102, 1)
(104, 0)
(106, 2)
(108, 3)
(110, 1)
(112, 0)
(114, 2)
(116, 3)
(118, 0)
(120, 2)
(122, 3)
(124, 0)
```

```
(126, 2)
average waiting time 8.56
```

SRTF

```
(0, 0)
(2, 2) 2
(4, 0) 2
(5, 3) 2
(7, 0) 4
(13, 1) 16
(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
```

SJF with Prediction

```
(0, 0)
(9, 1)
(17, 3)
(19, 2)
(30, 3)
(35, 2)
(41, 1)
(43, 0)
(60, 2)
(67, 1)
(70, 3)
(78, 0)
(90, 1)
(100, 0)
(110, 2)
(119, 3)
average waiting time 7.12
```

Task 2

1. Test for the given input

Short Job First with Prediction give least average waiting time.

For RR, when the time quantum is greater or equal to 10, the average waiting time is exactly the same as FCFS for the provided input, which also provide the best average waiting time for RR (6.44)

For SJF, it give the best average waiting time when alpha is greater or equal to 0.5. The minimum average waiting time for SJF is 7.12.

Below is the test result for RR:

```
(1, 8.94)
(2, 8.56)
(3, 9.12)
(4, 9.31)
(5, 8.94)
(6, 9.5)
(7, 9.62)
(8, 9.69)
(9, 8.88)
(10, 6.44)
(11, 6.44)
(12, 6.44)
(13, 6.44)
(14, 6.44)
(15, 6.44)
(16, 6.44)
(17, 6.44)
(18, 6.44)
(19, 6.44)
```

Below is the test result for SJF:

```
(0.05, 7.25)
(0.1, 7.25)
(0.15, 7.25)
(0.2, 7.25)
(0.25, 7.25)
(0.3, 7.44)
(0.35, 7.44)
(0.4, 7.44)
(0.45, 7.44)
(0.5, 7.12)
(0.55, 7.12)
(0.6, 7.12)
(0.65, 7.12)
(0.7, 7.12)
```

```
(0.75, 7.12)
(0.8, 7.12)
(0.85, 7.12)
(0.9, 7.12)
(0.95, 7.12)
```

2. What is the optimal scheduling scheme.

a) All short processes

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

Answer: As SRTF in preemptive case and SJF in non-preemptive case are optimal respectively. We only discuss the rest scheduling.

In both case, if the repetitive processes are predictable, i.e. burst time for the same process are almost the same with very small difference. SJF with prediction approximate the SJF, thus provide the best result. If the repetitive processes varies in burst time, SJF's performance is non stable and input dependent.

For all short process, FCFS or RR with quantum larger than the longest process provide the best result.

For short and long process interleave each other, FCFS does not provide good result as short process after the long one suffer the waiting time. In this case, RR provide good average waiting time.

3. 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 multi-processor system.

Answer: There are two ways of scheduling: 1. Each CPU core fetches process to run, 2. A core assign process to each core include itself. In case 1, it is complicated, race condition need to be handled to avoid two cores get the same process. And most of the operations in simulator need to be atomic. In my implementation, the minheap need to be global and all operation is atomic. In addition, everytime we get the min burst time process from the list, we need to remove it first. In case 2, it is simpler. Instead of assigning 1 process at the same time, we assign N processes at a time.

For implementation wise consideration, we no longer maintain the current time variable. Instead, we need to maintain each core's completion time. multi_core_simulation.py provides some sample codes to schedule in a multiple core environment.

[Git hub link](#)