Tutorial 4 Answer Sheet

CS3103 Operating Systems

Student Name: LIU Hengche

Student No. : 5 7 8 5 4 3 2 9

Day: ☐Tuesday ☑Wednesday ☐Friday

Time: ☐ 10:00 - 10:50 ☐ 11:00 - 11:50 ☐ 16:00 - 16:50 ☑ 18:00 - 18:50

CPU Scheduling

Submission:

- Deadline: Sunday, March 10, 2024, 23:59 pm HKT.
- Answers are allowed in text only. Any form of image/snapshot is not allowed.
- Submit this answer sheet via Canvas->Assignments->Tutorials->Tutorial 4.

Questions

Question 1: Compute the response time and turnaround time when running three jobs of length 200 with the SJF (`./scheduler.py -p SJF -l 200,200,200`) and FIFO (`./scheduler.py -p FIFO -l 200,200,200`) schedulers.

Answer:

SJF: 1). Response time: 0,200,400; 2). Turnaround time: 200,400,600;

Final statistics:

Job 0 -- Response: 0.00 Turnaround 200.00 Wait 0.00

Job 1 -- Response: 200.00 Turnaround 400.00 Wait 200.00

Job 2 -- Response: 400.00 Turnaround 600.00 Wait 400.00

Average -- Response: 200.00 Turnaround 400.00 Wait 200.00

FIFO: 1). Response time: 0,200,400; 2). Turnaround time: 200,400,600;

Final statistics:

Job 0 -- Response: 0.00 Turnaround 200.00 Wait 0.00

Job 1 -- Response: 200.00 Turnaround 400.00 Wait 200.00

Job 2 -- Response: 400.00 Turnaround 600.00 Wait 400.00

Average -- Response: 200.00 Turnaround 400.00 Wait 200.00

Question 2: Now do the same but with jobs of different lengths: 100, 200, and 300. The commands are (`./scheduler.py -p SJF -l 100,200,300`) and (`./scheduler.py -p FIFO -l 100,200,300`). What if you change the order of the job length? Try different orders to find the difference.

Answer:

SJF: 1). Response time: 0,100,300; 2). Turnaround time: 100,300,600;

Final statistics:

Job 0 -- Response: 0.00 Turnaround 100.00 Wait 0.00

Job 1 -- Response: 100.00 Turnaround 300.00 Wait 100.00

Job 2 -- Response: 300.00 Turnaround 600.00 Wait 300.00

Average -- Response: 133.33 Turnaround 333.33 Wait 133.33

FIFO: 1). Response time: 0,100,300; 2). Turnaround time: 100,300,600;

Final statistics:

Job 0 -- Response: 0.00 Turnaround 100.00 Wait 0.00

Job 1 -- Response: 100.00 Turnaround 300.00 Wait 100.00

Job 2 -- Response: 300.00 Turnaround 600.00 Wait 300.00

Average -- Response: 133.33 Turnaround 333.33 Wait 133.33

Since SJF would execute the jobs according to their runtime while FIFO would execute them according to their sequence, then the execution trace of the two scheduling methods are the same if the jobs are in the sequence of job runtime (i.e. 100<200<300). When we change the order of the job, then the difference could be seen. For example, if the order is 200,100,300, then FIFO order would be 200,100,300, the same as the job order, but SJF would be 100,200,300 as the order of job runtime. Below is the output of 200,100,300:

SJF: 1). Response time: 0,100,300; 2). Turnaround time: 100,300,600;

Final statistics:

Job 1 -- Response: 0.00 Turnaround 100.00 Wait 0.00

Job 0 -- Response: 100.00 Turnaround 300.00 Wait 100.00

Job 2 -- Response: 300.00 Turnaround 600.00 Wait 300.00

Average -- Response: 133.33 Turnaround 333.33 Wait 133.33

FIFO: 1). Response time: 0,200,300; 2). Turnaround time: 200,300,600;

Final statistics:

Job 0 -- Response: 0.00 Turnaround 200.00 Wait 0.00

Job 1 -- Response: 200.00 Turnaround 300.00 Wait 200.00

Job 2 -- Response: 300.00 Turnaround 600.00 Wait 300.00

Average -- Response: 166.67 Turnaround 366.67 Wait 166.67

Question 3: Compute the response time and turnaround time when running three jobs of length 3 with the RR scheduler and a time-slice of 1 (`./scheduler.py -p RR -l 3,3,3 -q 1`). Do the same but change the job lengths as 3,2,4.

Answer:

Job length of 3: 1). Response time: 0,1,2; 2). Turnaround time: 7,8,9;

Execution trace:

[time 0] Run job 0 for 1.00 secs

[time 1] Run job 1 for 1.00 secs

[time 2] Run job 2 for 1.00 secs

[time 3] Run job 0 for 1.00 secs

[time 4] Run job 1 for 1.00 secs

[time 5] Run job 2 for 1.00 secs

[time 6] Run job 0 for 1.00 secs (DONE at 7.00)

[time 7] Run job 1 for 1.00 secs (DONE at 8.00)

[time 8] Run job 2 for 1.00 secs (DONE at 9.00)

Final statistics:

Job 0 -- Response: 0.00 Turnaround 7.00 Wait 4.00

Job 1 -- Response: 1.00 Turnaround 8.00 Wait 5.00

Job 2 -- Response: 2.00 Turnaround 9.00 Wait 6.00

Average -- Response: 1.00 Turnaround 8.00 Wait 5.00

Job length of 3,2,4: 1). Response time: 0,1,2; 2). Turnaround time: 7,5,9;

Execution trace:

[time 0] Run job 0 for 1.00 secs

[time 1] Run job 1 for 1.00 secs

[time 2] Run job 2 for 1.00 secs

[time 3] Run job 0 for 1.00 secs

[time 4] Run job 1 for 1.00 secs (DONE at 5.00)

[time 5] Run job 2 for 1.00 secs

[time 6] Run job 0 for 1.00 secs (DONE at 7.00)

[time 7] Run job 2 for 1.00 secs

[time 8] Run job 2 for 1.00 secs (DONE at 9.00)

Final statistics:

Job 0 -- Response: 0.00 Turnaround 7.00 Wait 4.00

Job 1 -- Response: 1.00 Turnaround 5.00 Wait 3.00

Job 2 -- Response: 2.00 Turnaround 9.00 Wait 5.00

Average -- Response: 1.00 Turnaround 7.00 Wait 4.00

Question 4: For what types of workloads does SJF deliver the same turnaround times as FIFO?

Answer:

1. When all the jobs have the same same execution time, then SJF have no preference since the job runtime are all the same. This way SJF and FIFO would produce the same order of execution trace, resulting in identical turnaround times.

2. When the processes arrive in order of their burst times, from the shortest to the longest. This way SJF and FIFO would also schedule the processes in the same order, so that they have identical turnaround times.

Question 5: For what types of workloads and quantum lengths does SJF deliver the same response times as RR?

Answer:

When quantum length of RR is larger than or equal to the largest job length and the jobs come in increasing order, so that there would not be preemption in RR. This way SJF and RR would deliver the same response.

Question 6: What happens to response time with SJF as job lengths increase? Can you use the simulator to demonstrate the trend?

Answer:

The response time of SJF increases as the job lengths increase.

Python2 scheduler.py -p SJF -l 1,2,3 -c

Final statistics:

Job 0 -- Response: 0.00 Turnaround 1.00 Wait 0.00

Job 1 -- Response: 1.00 Turnaround 3.00 Wait 1.00

Job 2 -- Response: 3.00 Turnaround 6.00 Wait 3.00

Average -- Response: 1.33 Turnaround 3.33 Wait 1.33

Python2 scheduler.py -p SJF -l 2,4,6 -c

Final statistics:

Job 0 -- Response: 0.00 Turnaround 2.00 Wait 0.00

Job 1 -- Response: 2.00 Turnaround 6.00 Wait 2.00

Job 2 -- Response: 6.00 Turnaround 12.00 Wait 6.00

Average -- Response: 2.67 Turnaround 6.67 Wait 2.67

Python2 scheduler.py -p SJF -l 4,8,12 -c

Final statistics:

Job 0 -- Response: 0.00 Turnaround 4.00 Wait 0.00

Job 1 -- Response: 4.00 Turnaround 12.00 Wait 4.00

Job 2 -- Response: 12.00 Turnaround 24.00 Wait 12.00

Average -- Response: 5.33 Turnaround 13.33 Wait 5.33

We can see that if we double the job lengths, the average response time doubles. This suggests there's a linear relationship between job length and response time.

Question 7: What happens to response time with RR as quantum lengths increase? Can you write an equation that gives the worst-case response time, given N jobs?

Answer:

The response time increases as quantum lengths increase. This is because if the sequence in the job queue remains the same, suppose a job is the x-th to be executed, then the waiting time would be x*quantum, longer the quantum time, longer the response time.

The worst case response time is the N-th job. Then its response time is (N-1)*q (q is the quantum time)