Project Analysis

2019 Summer
CSCI 4210 Operating Systems
Yujue Wang (wangy66)
Ming Lu (lum8)

1. The SRT is the best algorithm with the shortest turnaround time and wait time.

For CPU-bound processes, SRT might be the best-suited algorithm. Because CPU-bound processes would spend most of the time doing CPU, the lower the wait time and turnaround time, the faster the process is.

Support:

Command line: 2 0.01 256 2 4 0.5 128

Algorithm FCFS

-- average wait time: 57.459 ms

-- average turnaround time: 164.486 ms

Algorithm SJF

-- average wait time: 57.459 ms

-- average turnaround time: 164.486 ms

Algorithm SRT

-- average wait time: 43.162 ms

-- average turnaround time: 151.054 ms

Algorithm RR

-- average wait time: 46.486 ms

-- average turnaround time: 154.378 ms

Command line: 64 0.001 4096 8 4 0.5 2048

Algorithm FCFS

-- average wait time: 3185.499 ms

-- average turnaround time: 4092.930 ms

Algorithm SJF

-- average wait time: 2889.805 ms

-- average turnaround time: 3797.237 ms

Algorithm SRT

-- average wait time: 2811.782 ms

-- average turnaround time: 3720.338 ms

Algorithm RR

-- average wait time: 3253.886 ms

-- average turnaround time: 4161.782 ms

For I/O-bound processes, SJF might be the best suited algorithm. Because in the ready queue, CPU-bound processes would take up longer CPU time each time when they get the CPU, and the I/O bound processes will always have less CPU time than CPU bound processes, the scheduling algorithms that favor those processes that have used the least processor time would be best suited for I/O-bound processes.

2. When there are more processes (more than 2) running in the system, changing from END to BEGINNING would decrease process wait time, and as a result, decrease turnaround time. Thus, adding the preempted process to the beginning of the ready queue would be a better approach here.

Support:

Command line: 2 0.01 256 16 4 0.5 128 END

-- average wait time: 646.331 ms

-- average turnaround time: 735.626 ms

Command line: 2 0.01 256 16 4 0.5 128 BEGINNING

-- average wait time: 631.324 ms

-- average turnaround time: 720.618 ms

Command line: 64 0.001 4096 8 4 0.5 2048 END

-- average wait time: 3253.886 ms

-- average turnaround time: 4161.782 ms

Command line: 64 0.001 4096 8 4 0.5 2048 BEGINNING

-- average wait time: 3187.630 ms

-- average turnaround time: 4095.526 ms

3. We recalculated the time by 4 different alpha values, 0.25, 0.5, 0.75, 0.9:

- SJF: The best result is alpha 0.25.

1. When alpha increases from 0.25 to 0.5, the wait time and turnaround time increase dramatically.
2. When alpha increases from 0.5 to 0.75, the wait time and turnaround time decrease dramatically.

3. When alpha increases from 0.5 to 0.75, the wait time and turnaround time increase slightly.
4. When alpha increases from 0.25 to 0.9, the total number of context switches is the same.

- Alpha 0.25

-- average wait time: 657.235 ms

-- average turnaround time: 745.538 ms

- Alpha 0.5

-- average wait time: 718.043 ms

-- average turnaround time: 806.346 ms

- Alpha 0.75

-- average wait time: 669.520 ms

-- average turnaround time: 757.823 ms

- Alpha 0.9

-- average wait time: 672.600 ms

-- average turnaround time: 760.903 ms

- SRT: The best result is alpha 0.25.

1. When alpha increases from 0.25 to 0.5, the wait time and turnaround time increase dramatically.
2. When alpha increases from 0.5 to 0.9, the wait time and turnaround time decrease gradually.

3. When alpha increases from 0.25 to 0.9, the total number of context switches increases steadily.
4. When alpha increases from 0.25 to 0.9, the total number of preemptions increases steadily.

- Alpha 0.25

-- average wait time: 658.765 ms

-- average turnaround time: 748.030 ms

- Alpha 0.5

-- average wait time: 742.488 ms

-- average turnaround time: 832.035 ms

- Alpha 0.75

-- average wait time: 695.521 ms

-- average turnaround time: 785.248 ms

- Alpha 0.9

-- average wait time: 685.909 ms

-- average turnaround time: 775.695 ms

4. According to the simulation of SRT and SJF at alpha 0.5, with increasing of the number of processes, the average wait time and turnaround time of SRT are gradually much lower than SJF. Although the context switches time increasing, SRT decreases the wait time and is more optimal for multiple processes.

5.

a. There is no logic in I/O burst. Since there is no algorithm applied to I/O burst for each process, there would be multiple processes doing I/O at the same time, which shouldn't be happening in the real-world situation. Thus, to better model real-world operating system, the certain algorithm should be taken into account in arranging I/O burst.
b. We modeled the system with arrivals occurring at a constant average rate. However, the real world system would be more random.
c. We avoided randomly generated values that are far down the "long tail" of the distribution. The "long tail" processes would exist in the real world.
d. We only operate a single CPU, there will be more than one CPU in the real-world operation.

6.

- Description: This algorithm is based on SJF, but an extra wait queue and a waiting time upper limit time are added. Once a process finishes the burst, the program will check both the next shortest time job and the longest waiting process. If the wait time of a process is over the upper limit (waiting for a long time), the program will give the priority to this process rather than the shortest time job.
- Advantage: Such an algorithm will avoid a process wait for a long time.
- Disadvantage: Correspondingly, the turnaround time will increase, because during the running of a longer process other processes will wait for much more time, like the comparison of FCFS and SJF.