CPSC 449 – Programming Paradigms

Assignment #1

Due Date: Friday, September 28, 11:59 PM

Individual Work:

All assignments in this course are individual work. Students are advised to read the guidelines for avoiding plagiarism located on the course outline. Students are also advised that electronic tools may be used to detect plagiarism.

Late Penalty:

Late submissions will not be accepted.

Constraints:

You are *__NOT__* allowed to use built-in java data structure such as stack, queue. However, you may use ArrayList.

Submissions:

Submit separate java files and/or five (if you are doing additional challenge) using D2L. Submit a reflection report if you are working on additional challenge (see additional challenge for more details).

Problem Description:

CPU Scheduling Algorithm[1]:

In the part of the assignment, you will implement four different process scheduling algorithm used by operating system.

Calculate the following for each of the algorithm:

- Wait time for each process
- Average wait time of each process

1- First-come first-serve scheduling algorithm (FCFS)

FCFS is a very simple algorithm based on queue. Each job is scheduled as they arrived. Consider the example below:

[1] CPU Scheduling

Process	Burst Time (ms)
P1	24
P2	3
P3	3

In this case, the wait time for P1 is 0ms, P2 is 24ms, and P3 is 27ms while the average wait time would be 17.0ms.

2- Shortest-job-first scheduling algorithm (SJF)

The SJF algorithm picks the quickest little job first, performs the task, and then picks the next smallest fastest job to do.

Process	Burst Time (ms)
P1	6
P2	8
P3	7
P4	3

In the case above the average wait time is (0 + 3 + 9 + 16) / 4 = 7.0 ms,

3- Priority Scheduling

In the case of Priority scheduling, each job is assigned a priority and the job with the highest priority gets scheduled first.

Process	Burst Time (ms)	Priority
P1	10	3
P2	1	1
P3	2	4
P4	1	5
P5	5	2

For this case, the average wait time would be 8.2ms.

4- Round-Robin scheduling

Round robin scheduling is similar to FCFS scheduling, except that CPU bursts are assigned with a certain limits called a time quantum. Your CPU switches among the process.

- In the case where, process finishes its burst before the time quantum timer expires, then this process is swapped out of the CPU.
- If the timer goes off first, then the process is swapped out of the CPU and moved to the back end of the queue.

Consider the following example where we have three processes and time quantum is 4ms:

Process	Burst Time (ms)
P1	24
P2	3
P3	3

P1	P2	P3	P1	P1	P1	P1	P1
0	4	7	10	14	18	22	26	30

Wait time for P1 = 0 + (3 + 3) = 6
Wait time for P2 = 4
Wait time for P3 = 7

Average wait time = 17/3 = 5.66ms

Additional Challenge:

Implement the four scheduling algorithm in C language with the following constraints.

- Your program should ask first how many process user wants to schedule
- Assignment memory accordingly
- Take burst time and other input if needed
- Apply each of the scheduling algorithm on the processes
- Display the results

This is a course (cs11 by caltech) to start working with C, if you have never worked it before.

Reflection:

Write a reflection report comparing solving the same problem in both C and Java.

- How do you compare C with Java considering solving the same problem?
    - What were the challenges?

We suggest that you submit a reflection report even though you tried additional challenge and unable to finish.

Grading:

A+: All parts of the assignments are completed successfully, included the additional challenge.

A: All parts of the assignment, except for the additional challenge, are completed successfully.

B: Parts 1, 2 and 3 of the assignment are completed successfully.

C: Parts 1 and 2 of the assignment are completed successfully.

D: Part 1 of the assignment is completed successfully.

Credits:

The text is adapted from the University of Illinois at Chicago and that can be found here.

Reference:

Abraham Silberschatz, Greg Gagne, and Peter Baer Galvin, "Operating System Concepts, Eighth Edition ", Chapter 5