CS 3305A: Operating Systems
Department of Computer Science
Western University
Assignment 2
Fall 2019
Due Date: October 30th 2019

Purpose:

The goals of this assignment are the following:

- Gain more experience with the C programming language from an OS's *process/thread* and *CPU scheduling* perspective.
- Get hands-on experience with the process/thread related *function calls* and CPU *scheduling algorithms.*

Part I: Process vs Thread (20 points)

You will be writing a C program to test the data sharing ability of a *thread* and *process*. Your C program will do the following:

1. Your *parent* program will have three variables: *int x,y,z;* which to be initialized as 10, 20, and 0, respectively.

2. *parent* creating *child*: *parent* will create a *child* by *fork()* and the child will perform *z = x+y* (i.e., add x and y and store the results in z). *parent* will wait for *child* to complete before parent proceeds. Upon completion of *child*, *parent* will print out the value of z. (8 points)

3. *parent* creating *thread*: After (2) above is completed, *parent* process will now create a *thread* by *pthread_create()* which will do the exact same task done by *child* above (i.e., z = x+y). *parent* will wait for its *thread* to complete before *parent* proceeds. Upon completion of the *thread*, *parent* will print out the value of z. (12 points)

Part II: Performance Evaluation of CPU Scheduling Algorithms (80 points)

You will be applying *CPU Scheduling Algorithms* in the C programming language. A sample input file is provided here (see below part II_d) which must be used to develop the *CPU Scheduling Algorithms*.

Part II_a: CPU Scheduling Environment Initialization (15 points)

Your C program will perform the following tasks based on the given input file *cpu_scheduling_input_file.txt*:

1. Create the number of ready queues as stated in the given input file
2. Assign time quantum (provided in the input file) for Round Robin (RR) algorithm
3. Create all the processes for each of the ready queues based on the input file specification (such as CPU burst time, arrival order etc.)

Part II_b: Scheduling Algorithm Execution (45 points)

Your C program will perform the following tasks in order based on the given input file:

1. Ready queues will be executed in the order of their queue number (q 1, q 2, etc.)
2. CPU scheduling algorithms FCFS, SJF, and RR will be applied on each ready queue

Part II_c: Results (20 points)

Once the execution of all the ready queues is complete, your C program should output results to the screen and into a text file “cpu_scheduling_output_file.txt”. For each queue, your output must include the following information:

1. Order of the processes selected by CPU in Ready Queue X
2. Individual waiting time for each process in Ready Queue X for FCFS and SJF
3. Average waiting time for Ready Queue X for FCFS and SJF
4. Turnaround time for each process in Ready Queue X for RR

A sample output file “cpu_scheduling_output_file.txt” is provided on the course website (under Assignment section) and your output file “cpu_scheduling_output_file.txt” must follow the same format.

Part II_d: Input File

```
q 1 tq 4 p1 30 p2 10 p3 24 p4 20 p5 17 p6 4 p7 7 p8 11 p9 8 p10 9 p11 5 p12 6 p13 3 p14 2 p15 1
q 2 tq 5 p1 1 p2 2 p3 2 p4 9 p5 8 p6 5 p7 12 p8 11 p9 15 p10 1 p11 4 p12 8 p13 22 p14 21 p15 30
q 3 tq 30 p1 30 p2 10 p3 24 p4 20 p5 17 p6 4 p7 7 p8 11 p9 8 p10 9 p11 5 p12 6 p13 3 p14 2 p15 1
```

Symbols used in the above input file:

----

q: Ready queue
tq: time quantum

Example:

```
q 1 tq 4 p1 10 p2 5 p3 7 p4 20 p5 17 p6 9 p7 3 p8 11 p9 15 p10 1
```

Ready Queue 1 has a total of ten processes namely p1, p2, p3, p4, p5, p6, p7, p8, p9, and p10. The sequence of these processes represents their arrival order. For example, p1 arrives first and p10 arrives last in this list of processes. In the "px y" format, y refers to the CPU burst time for px. For preemptive scheduling algorithms such as RR, a time quantum of 4 is assigned to each process.

Assignment related technical resources

Please visit the course website for submission instructions and other relevant materials. Also, consult TAs, and Instructor for any question you may have regarding this assignment.