Spring 2019 COMP 3511 Homework Assignment #2

Handout Date: March 8, 2019 Due Date: March 22, 2019

Name: ______________ ID: ___________________E-Mail: _____________

Please read the following instructions carefully before answering the questions:

- You should finish the homework assignment **individually**.
- There are total of **4** questions.
- When you write your answers, please try to be precise and concise.
- Fill in your name, student ID, email at the top of each page.
- Please fill in your answers in the space provided, or you can type your answers in the MS Word file.

- **<u>Homework Submission</u>**: the homework is submitted to **assignment #2** on **CASS**

1. [20 points] Multiple choices

   1) Which of the following components of program state are shared across threads in a multithreaded process?
   ①Register values ②Heap memory ③Global variables ④Stack memory

   A) ①②
   B) ②③
   C) ①④
   D) ③④
   Answer: __

   2) Which one of the following is not shared by threads?
   A) program counter
   B) stack
   C) both program counter and stack
   D) none of the mentioned
   Answer: __

   3) Which module gives control of the CPU to the process selected by the short-term scheduler?
   A) dispatcher
   B) interrupt
   C) scheduler
   D) none of the mentioned
   Answer: __

4) Cancelling a thread asynchronously ___________.

A) spoils the process execution
B) may not free each resource
C) frees all the resources properly
D) allows the target thread to periodically check if it should be cancelled
Answer: __

5) According to Amdahl's Law, what is the speedup gain for an application that is 40% parallel and we run it on a machine with 4 processing cores?

A) 0.7
B) 1.82
C) 1.43
D) 0.55
Answer: __

6) Which of the following scheduling algorithms could result in starvation?
①First-come, first served ②Shortest job first (non-preemptive)
③Round Robin ④Priority

A) ①②
B) ②④
C) ①③
D) ③④
Answer: __

7) CPU scheduling is the basis of ___________
A) multiprocessor systems
B) multiprogramming operating systems
C) larger memory sized systems
D) none of the mentioned.
Answer: __

8) One of the disadvantages of the priority scheduling algorithm is that:
A) it schedules in a very complex manner
B) its scheduling takes up a lot of time
C) it can lead to some low priority process waiting indefinitely for the CPU
D) none of the mentioned
Answer: __

9) LWP is ____________.

A) short for lightweight processor
B) placed between user and kernel threads
C) placed between system and kernel threads
D) common in systems implementing one-to-one multithreading models
Answer: __

10) With round robin scheduling algorithm in a time shared system,
A) using very large time slices degenerates it to First Come First Served scheduling algorithm
B) using very small time slices degenerates it to First Come First Served scheduling algorithm
C) using extremely small time slices increases performance
D) using very small time slices degenerates it to Shortest Job First algorithm
Answer: __

2. [20 points] Multithread Process

1) What set of resources are needed or used when a thread within a process is created? How do they differ from those used when a process is created? (4 points)

2) Linux does not distinguish between processes and threads. Instead, it treats them in the same way by allowing a task to be more akin to a process or a thread depending on the set of flags passed to the clone() system call. Please compare and contrast clone() and fork() in Linux. (4 points)

3) Consider a multicore system and a multithreaded program using *many-to-many threading model*. Let the number of user-level threads in the program be much greater than the number of processing cores in the system. How many kernel threads should the OS allocate in order to increase the utilization of the multiprocessor system, and why ? (4 points)

Hint: Consider the number of kernel threads allocated to the program is less than, equal to or is greater than the number of processing cores.

4) Using Amdahl's Law, list the speedup gain of an application in a table or figure that has a 60 percent parallel component for systems with one, two, three, … and eight processing cores. (8 points)

3. [20 points] CPU scheduling

1) Distinguish between PCS and SCS scheduling. Under what threading model, PCS and SCS are the same? (5 pints)

2) Can FCFS be considered as one type of priority scheduling? Please be careful in justifying your answer. (4 points)

3) What is the difference between response time and turnaround time? How do they conflict with each other? (5 points)

4) CPU scheduling often needs to balance CPU-bound and I/O-bound programs. Suppose we design a simple CPU scheduling algorithm that favors those processes that have used the least CPU time in the recent past. Why will this algorithm favor I/O-bound programs and yet not permanently starve CPU-bound programs? Can this also improve I/O device utilization? (6 points)

4. [40 points] Scheduling algorithms

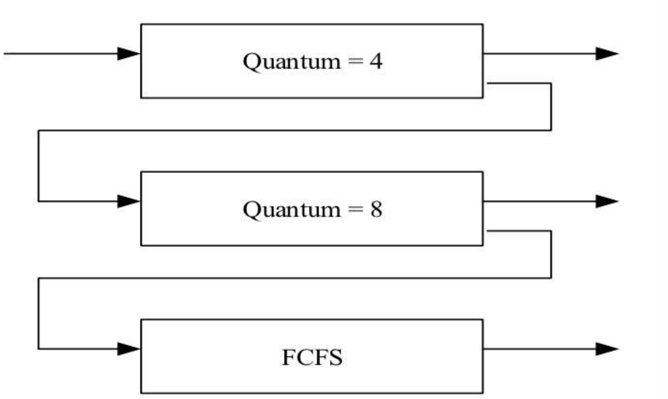
1) Consider two processes, process A is CPU-bound with CPU burst time one hour, and process B is I/O-bound repeating CPU burst time 1 ms and disk I/O 9 ms. Please illustrate using Gnatt charts the round-robin scheduling with a time quantum respectively setting to 100 ms and 1 ms, and SRTF scheduling. Please discuss which one performs the best (10 points)? Hint: consider disk utilization and number of context switch.

2) Given the arrival time and CPU-burst times of 6 processes shown in the following diagram:

Process	Arrival Time (ms)	Burst Time (ms)
P1	0	8
P2	3	16
P3	7	3
P4	15	12
P5	20	6
P6	5	2

Suppose the OS uses a 3-level feedback queue to schedule the above 6 processes. Round-Robin scheduling strategy is used for the queue with the highest priority and the queue with the second highest priority, but the time quantum used in these two queues is different. First-come-first-serve scheduling strategy is used for the queue with the lowest priority. The scheduling is **preemptive**.

(Note: In this scenario, the scheduling is preemptive, which means that the execution of the current job may be preempted by another job with **higher** priority. a newly arriving job can preempt a job currently running only if its priority is **higher** than this job)



Construct a Gantt chart depicting the scheduling for the set of processes specified in the above diagram using this 3-level feedback queue, and compute the average waiting time for all processes (10 points)

3) (20 points) Consider the following set of processes, with the length of the CPU burst time given in milliseconds:

Process	Arrival Time(ms)	Burst Time(ms)
P1	0	4
P2	2	3
P3	3	1
P4	6	2
P5	7	7
P6	11	4

a) (10 points) Draw four Gantt charts that illustrate the execution of these processes using the scheduling algorithms listed below:

(i) FCFS
(ii) SJF
(iii) Non-Preemptive priority (a smaller priority number implies a higher priority), with the priorities listed here:

Process	Priority
P1	1
P2	4
P3	2
P4	5
P5	1
P6	3

(iv) RR (quantum = 4)

b) (4 points) What is the turnaround time of each process for each of the scheduling algorithms in part a?

Turnaround time	P1	P2	P3	P4	P5	P6
FCFS						
SJF						
Non-Preemptive priority						
RR						

c) (4 points) What is the waiting time of each process for each of these scheduling algorithms in part a?

Waiting time	P1	P2	P3	P4	P5	P6
FCFS						
SJF						
Non-Preemptive priority						
RR						

d) (2 points) Which of the algorithms results in the minimum average waiting time (over all processes) mentioned above?