Spring 2015 COMP 3511 Homework Assignment #2 Solution

Handout Date: Mar 10, 2015 Due Date: Mar 24, 2015

Name: _____ ID: _____E-Mail: _____

Please read the following instructions carefully before answering the questions:

- You should finish the homework assignment **individually**.
- There are a 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.

- **Homework Collection: the hardcopy** is required and the homework is collected in collection box #16. The collection boxes locate outside Room 4210, near lift 21 (there are labels attached on the boxes).

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: B

   2) A program initializes a global variable X to zero, then spawns three threads, each of which reads X, increments it, and stores the result back. What is the value of X when all threads terminate?

   A) X = 1
   B) X = 3
   C) X = 1 or X = 3
   D) X = 1 or X = 2 or X = 3
   Answer: D

   3) Which of the following is the drawback of the One to One Model?

   A) Increased concurrency provided by this model
   B) Decreased concurrency provided by this model
   C) Creating so many threads at once can crash the system

D) Creating a user thread requires creating the corresponding kernel thread
Answer: D

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: B

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: C

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: B

7) In multilevel feedback scheduling algorithm, _____.

A) a process can be moved to a different classified ready queue.
B) classification of ready queue is permanent
C) processes are not classified into groups
D) None of the mentioned.
Answer: A

8) Dispatch latency is _____.

A) the speed of dispatching a process from running to the ready state
B) the time of dispatching a process from running to ready state and keeping the CPU idle
C) the time to stop one process and start running another one
D) None of these

Answer: C

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: B

10) Thread-local storage is data that _____.

A) is unique to each thread
B) has been modified by the thread, but not yet updated to the parent process
C) is generated by the thread independent of the thread's process
D) is not associated with any process
Answer: A

2. (15 points) Please answer the following question briefly.

1) (5 points) How is clone() used in Linux in comparing with fork()?

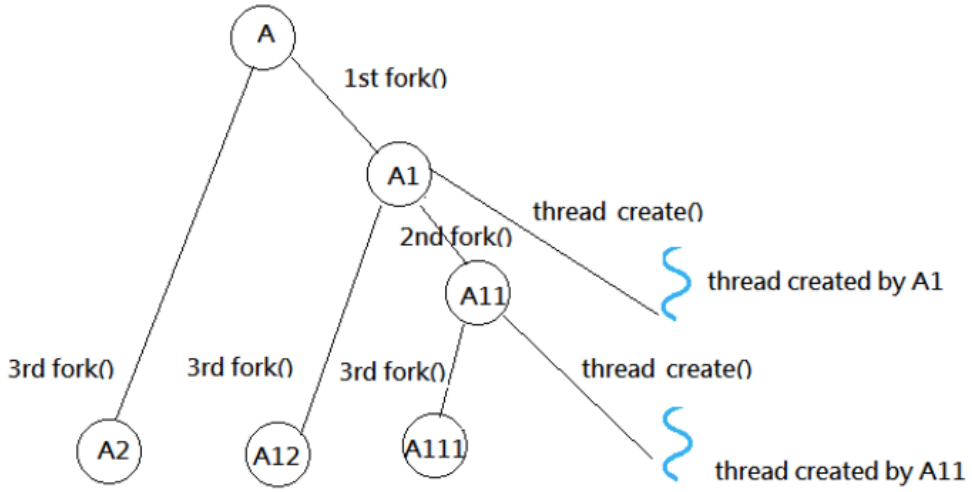
Answer: Linux uses both fork() and clone() to create a task. When clone() is used, it can use a set of parameters or flags that specifies how much sharing between the parent and child tasks.

2) (10 points) Consider the following code segment:

```
pid_t pid;
    pid = fork();
    if (pid == 0) { /* child process */
         fork();
         thread_create(...);
    }
fork();
```

How many unique processes are created? How many unique threads are created?

Answer: There are six processes and two threads. Let the original process to be A. After the first fork(), it becomes two process, A and A1 (both share the following codes). Suppose the process A runs first, it will do another fork(), generate another process A2. The child process A1 runs, it enters (pid==0) section of codes, it does the fork() that generates a new process called A11, and A11 shares all the following codes as the A1. So each does a thread_create(..), generating total 2 threads. Further, each does one last fork(), it will generate two more process A12 and A111. So there are total 6 processes.



1st fork() : the fork() before the "if" loop
2nd fork(): the fork() inside the "if" loop
3rd fork() : the fork() after the "if" loop

3. (20 points) Please answer the following question briefly.

1) (5 points) What are the two primary reasons for introducing threads?

Answer: (1) processes can involve performing multiple tasks; (2) creating several processes to handle each task is cumbersome, time- and resource-consuming. Creating multiple threads within a process eases this job by enabling natural sharing of codes, data or other resources.

2) (5 points) Please name three distinctive parameters of threads within a process? Please explain that a process does not have its own state if the process has multiple threads?

Answer: thread ID, program counters, registers, scheduling information, stack and etc. (1 points each). Since now a process has multiple threads, or multiple execution units, each of which can be in different states, the process state cannot be defined (3 points).

3) (3 points) How is a dispatcher related to and different from the short-term scheduler?

Answer: The short-term scheduler is responsible for selecting a process from a set of processes on the ready queue to run on the CPU next. The dispatcher is responsible for allocating the CPU to the process selected by the short-term scheduler, which involves context switching.

4) (5 points) What is the difference between response time and turnaround time?

Answer: Turnaround time measures the amount of time it takes to execute a process, which is computed between the time of its first arrival to the ready queue and the completion time of all its CPU bursts. Response time measures the time that elapses between a request and the first response produced, which is the time between arrival to the ready queue and the completion time of its current CPU burst.

5) (2 points) What is the main difference between a preemptive scheduling and a non-preemptive scheduling?

Answer: In the preemptive scheduling, a process will not be forced to give up CPU, while preemptive scheduling can.

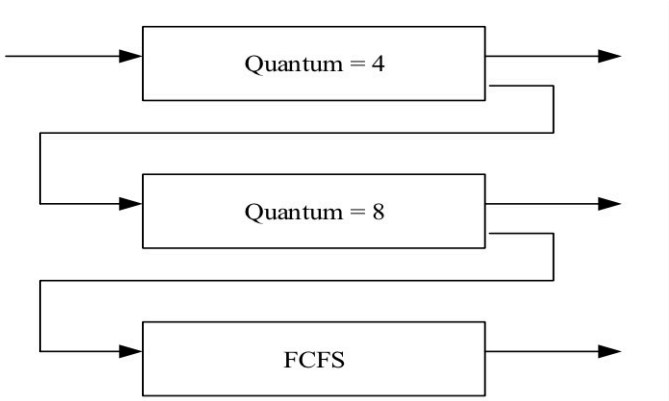
4. (45 points) CPU Scheduling

1) (15 points) Given the arrival time and CPU-burst of 5 processes shown in the following diagram:

Process	Arrival Time (ms)	Burst Time (ms)
P1	0	10
P2	2	15
P3	5	2
P4	12	14
P5	18	6

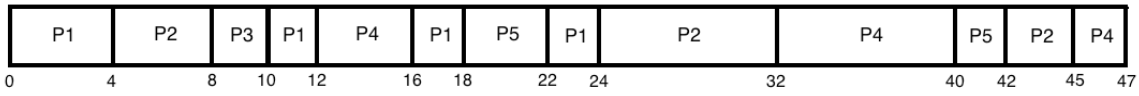
Suppose the OS uses a 3-level feedback queue to schedule the above 5 processes. Round-Robin scheduling strategy is used for the queue with the highest priority and the queue with the second highest priority, but the time quantum used in these two queues is different. First-come-first serve scheduling 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 arrived 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.

Answer:



2) (30 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	5
P3	3	2
P4	5	1
P5	7	8
P6	11	4

a) (16 points) Draw four Gantt charts that illustrate the execution of these processes using the scheduling algorithms listed below:

(i) FCFS

(ii) SJF (non-preemptive)

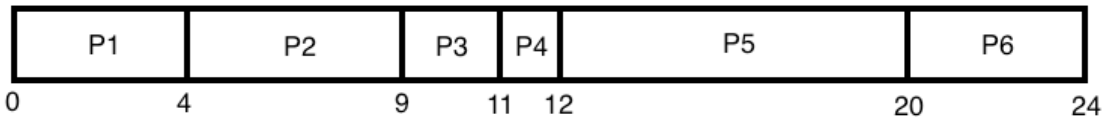
(iii) Preemptive priority (a smaller priority number implies a higher priority), with the priorities listed here:

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

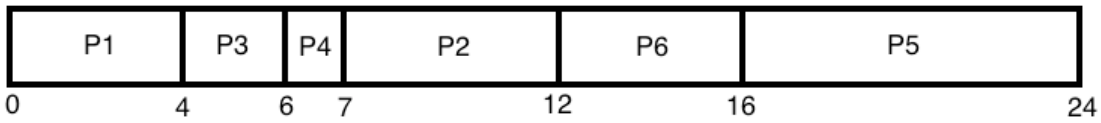
(iv) RR (quantum = 4)

Answer:

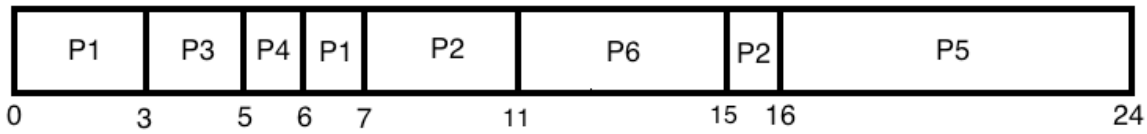
(i) FCFS



(ii) SJF



(iii) Preemptive priority



(iv) RR (quantum = 4)

b) (6 points) What is the turnaround time of each process for each of the scheduling algorithms in part a?

Answer:

Turnaround time	P1	P2	P3	P4	P5	P6
FCFS	4	7	8	7	13	13
SJF	4	10	3	2	17	5
Preemptive priority	7	14	2	1	17	4
RR	4	14	7	6	17	9

c) (6 points) What is the waiting time of each process for each of these scheduling algorithms in part a?

Answer:

Waiting time	P1	P2	P3	P4	P5	P6
FCFS	0	2	6	6	5	9
SJF	0	5	1	1	9	1
Preemptive priority	3	9	0	0	9	0
RR	0	9	5	5	9	5

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

Answer: SJF