Operating System Midterm Exam #2

Teacher: Professor **Chu-Sing Yang**

Date: 2010/05/17 09:10 ~ 10:50 Name _____

Student ID _____

1. (25%)
   (a) What is the CPU scheduler.
   (b) Describe the four conditions that CPU scheduling decisions may take place.
   (c) What is a dispatcher.
   (d) Describe the five popular criteria using in CPU scheduling algorithm.

Answer: (a) Whenever the CPU becomes idle, the operating system must select one of the processes in the ready queue to be executed. The selection process is carried out by the short-term scheduler (or CPU scheduler). The scheduler selects a process from the processes in memory that are ready to execute and allocates the CPU to that process.

(b) CPU scheduling decisions may take place when a process:
(1) Switches from running to waiting state
- The result of an I/O request
- Wait for the termination of one of the child processes

(2) Switches from running to ready state
- An interrupt occurs

(3) Switches from waiting to ready
- Completion of I/O

(4) Terminates

(c) The dispatcher is the module that gives control of the CPU to the process selected by the short-term scheduler; this involves:
- switching context
- switching to user mode
- jumping to the proper location in the user program to restart that program

(d) CPU utilization, Throughput, Turnaround time, Waiting time, and Response time. (Please reference the chap 5 slide 5.13.)

2. (10%)

(1) In multi-processor system, operating system attempts to keep the workload balancing across all processors. Please describe two general approaches of load balance.

(2) Load balance often contradicts the benefits of processor affinity. Please explains why?

(1) Push migration

- A specific task periodically checks the load on each processor
- If it finds an imbalance
- Evenly distributes the load by moving (pushing) processes from overloaded to idle or less-busy processors

Pull migration

- Occurs when an idle processor pulls a waiting task from a busy processor

(2)The benefit of keeping a process running on the same processor is that the process can take advantage of its data being in cache. By either pulling or pushing a process from one processor to another, we invalidate the benefit.

3. (20%)

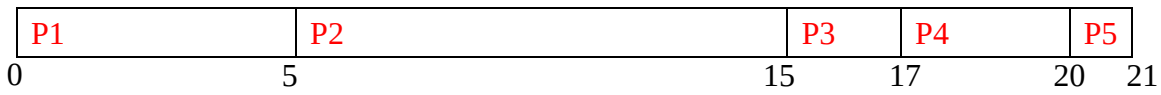
Consider the following set of processes, with the length of the CPU-burst time given in milliseconds:

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

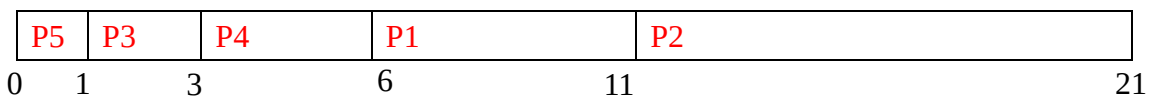
The processes are assumed to have arrived in the order *P*1, *P*2, *P*3, *P*4, *P*5, all at time 0.

a. Draw four Gantt charts illustrating the execution of these processes using FCFS, non-preemptive SJF, a non-preemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 2) scheduling.

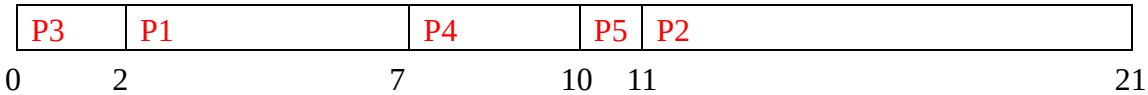
FCFS



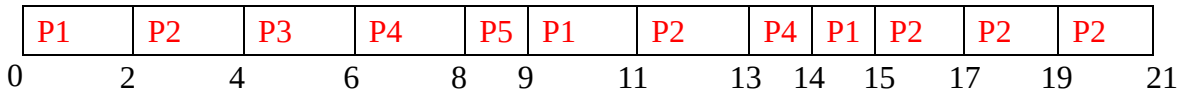
SJF



Priority



RR



4. Provide two kind of programming in which multithreading does not provide better performance than a single-threaded solution. (10%)

Answer: (1) Any kind of sequential program is not a good candidate to be threaded. An example of this is a program that calculates an individual tax return. (2) Another example is a "shell" program such as the C-shell or Korn shell. Such a program must closely monitor its own working space such as open files, environment variables, and current working directory.

5. Which of the following components of program state are shared across threads in a multithreaded process? (5%)

   a. Register values b. Heap memory c. Global variables d. Stack memory

Answer: The threads of a multithreaded process share heap memory and global variables. Each thread has its separate set of register values and a separate stack.

6. Describe the benefits of thread pools. (10%)

Answer:

(1) Servicing a request with an existing thread is usually faster than waiting to create a thread.

(2) A thread pool limits the number of threads that exist at any one point. This is particular important on systems that cannot support a large number of concurrent threads.

7. Please describe four benefits of using multithreading. (10%)

Answer:

Responsiveness, resource sharing, economy, and utilization of MP Architectures.

(Please reference the chap 4 slide 4.10.)

8. Please describe five challenges of multicore programming.

Answer:

Dividing activities, balance, data splitting, data dependency, testing and debugging.

(Please reference the chap 4 slide 4.13.)