Q1

Suppose that the following processes arrive for execution at the times indicated. Each process will run for the listed amount of time.

Process Arrival Time Service Time

P1 0 4

P2 1 7

P3 4 3

P4 6 8

1. Choose two out of the following three scheduling algorithms :

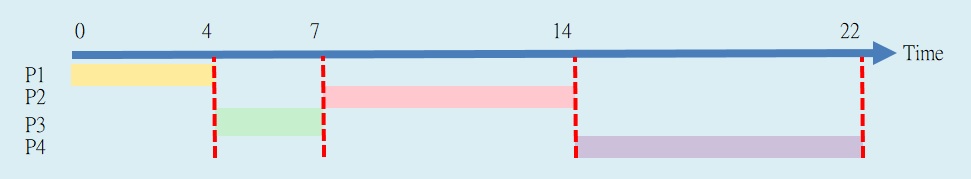
Round Robin (Q=1)

Shortest Remaining Time

Multilevel Feedback Scheduling (as covered in class : 3 queues; highest priority queue round robin with Q=1, next lower priority queue, RR with Q=2, last queue FCFS) Non-preemptive priority between queues.

1. For the chosen algorithm draw the Gantt diagram. Give their advantages/disadvantages.

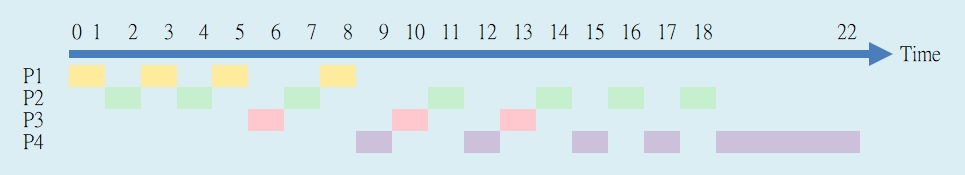
Shortest Remaining Time



Advantages : better average waiting time, among all non-preemptive

Disadvantages : not soltable for CPU – scheduling hard to know in advance CPU buest

Round Robin



Advantages : each one can be service for a Q time

Disadvantages : Very important jobs wait in line. Largest job takes enough time for

completion. Setting the quantum too short causes too many

context switches and lower the CPU efficiency. Setting the quantum

too long may cause poor response time and approximates FCFS.

Q2

After each step, give the value of the updated semaphore and the content of the updated semaphore queue.

**Counting** Semaphores : S1, S2, S3

**Binary** Semaphores : S4, S5, S6

Semaphores initial values :

S1=1, S2=0, S3=2, S4=0, S5=1, S6=1

1. P1 : P(S3)
2. P2 : P(S5)
3. P3 : P(S5)
4. P5 : P(S2)
5. P2 : V(S2)
6. P10 : V(S6)
7. P2 : P(S4)
8. P6 : P(S5)
9. P4 : V(S4)
10. P4 : P(S2)
11. S3=1
12. S5=0
13. S5=0 S5Q(P3)
14. S2=-1 S2Q(P5)
15. S2=0
16. S6=1
17. S4=0 S4Q(P2)
18. S5=0 S5Q(P3,P6)
19. S4=0
20. S2=-1 S2Q(P4)

**1. What is meant by a thread in runnable state?**

When a thread is started, it is made runnable. Resources are allocated to the thread.

**2. Which method invokes the run() method?**

The start method.

**3. What is the time unit of sleep method?**

The time unit of the sleep method is milliseconds.

**4. What are the advantages / reasons of using threads instead of processes?**

* Inter-thread communication (sharing data etc.) is significantly simpler to program than inter-process communication.
* Context switches between threads are faster than between processes. That is, it's quicker for the OS to stop one thread and start running another than do the same with two processes.

**5. Which are the two ways to implement threads?**

Sub-classing the thread class and overriding the run( ) method (implement Runnable or extends Thread).

**6. Which scheduling algorithm Java uses?**

Java uses a preemptive priority CPU scheduling algorithm. FIFO if 2 threads have the same priority.

**7. What is NORM\_PRIORITY (default priority) in Java?**

The range of Thread priority in [java](http://www.javaranch.com/) is 1-10; min is 1, max is 10. The default priority of any thread is 5.

**8. Explain Try Catch block inside sleep method.**

For the *wait( )* and *sleep( )* methods, if the thread that is interrupted is blocked, the method that blocked the thread throws an InterruptException object.

**9. Thread Methods**

* suspend( ) suspends execution of the currently running thread. (the method is deprecated, deadlock for monitors)
* join( ) waits for this thread to die.
* wait( ) on an object.
* sleep( ) puts the currently running thread to sleep for a specified amount of time (milliseconds)

For the wait( ) and sleep( ) methods, if the thread that is interrupted is blocked, the method that blocked the thread throws an InterruptException object.

**10. Advantages / Disadvantages****of SA’s**

*Non-preemptive Scheduling:* Processes can remain on CPU until they terminate

*Shortest Job First:*

* Advantages- Optimal average waiting time (among non-preemptive SA’s)
* Disadvantages- Not suitable for CPU scheduling (because you would need to know CPU burst)

*FCFS:*

* The ready queue contains the PCB of the process and it is treated like a FIFO queue.
* Advantages- Easy to implement because there are no computations
* Disadvantages- Large processes are created in front of the queue 🡪 creates poor average waiting/ turnaround time

*Preemptive Scheduling:* The schedule can forcibly remove a process from the CPU

*Round Robin:*

* Used for Time-sharing environments. Processes are dispatched in a FIFO sequence but each process can only run for a limited amount of time, known as a **time-slice** or **quantum**.
* Advantages- Considered fair- each process has a chance to use the CPU
* Disadvantages- Slightly larger than average CPU bursts of I/O bound process

*Shortest Remaining Time:*

* Process with the shortest CPU burst is scheduled. If a new process (P2) arrives in the ready queue with a shorter service time, P2 is scheduled and P1 is preempted.
* Advantages- Optimal average waiting time (among preemptive SA’s)
* Disadvantages- Hard to implement for CPU

Threads:

* Multithreading allows a process to have more than 1 thread.
* Heavy Weight process has 1 thread and 1 task.
* Light Weight process consists of a program counter, register set and a stack space.
* The most important resource is the address space.

Main Memory of threads contain:

* Thread Control Block
* Execution Stack
* Minimal Resources (Set of Registers)
* Threads are efficient because they work as a team and they are sharing the same resources.
* One to One, One to Many, Many to One, Many to Many

Advantages:

* Threads can share address space.
* Better resource utilization, threads share memory space with Process.
* Easier to control and manipulate by the O.S.
* Creating a thread is faster than creating another process.

Disadvantages:

* Operating System needs to support multithreaded environment. Synchronization is difficult and stricter because threads are sharing resources.
* Process Thread is the main thread and executes in main memory. If main thread terminates all threads terminate.

Thread Process States:

New Thread start()

Runnable yield()

Blocked wait(), sleep()

Terminate exit(), abnormal exit by user, completion

If thread is in wait(); notify() or notifyall() must be called.

isAlive() – boolean function to determine if thread is alive.