

Quiz 4, Q&A

Q What is semaphore? Describe the down operation.

A. Semaphores are variables that are used to signal the status of shared resources to processes.

the **down** operation on a semaphore

- checks to see if the value is greater than 0
- if so, it decrements the value and continues
- if the value is 0, the process is put to *sleep* without the completing the **down** for the moment
- all is done as a single, indivisible atomic action
- it is guaranteed that once a semaphore operation has started, no other process can access the semaphore until the operation has completed

Q. What is the Monitor? Adv. & Disadv.

A. Monitors are a high level language construct for dealing with synchronization.

Adv.

- Monitors are easier to program than semaphores
- programmer only has to say what to protect, compiler actually does the protection (compiler will use semaphores to do protection)
- Natively supported by a number of programming languages: Java

- Disadv.

-

Main problem: provides less control

Q Explain the metrics in the scheduling.

Metrics;

-

Execution time: T_s

-

Waiting time: time a thread waits for execution: T_w

-

Turnaround time: time a thread spends in the system (waiting plus execution time): $T_s + T_w = T_r$

-

Normalized turnaround time: T_r / T_s

Q Describe FCFS & SJF scheduling algorithms, give example.

1.

FCFS is the simplest scheduling policy

2.

Arriving jobs are inserted into the tail of the ready queue and the process to be executed next is removed from the head (front) of the queue

3.

FCFS performs better for long jobs

1.

SJF policy selects the job with the shortest (expected) processing time first

2.

Shorter jobs are always executed before long jobs

3.

One major difficulty with SJF is the need to know or estimate the processing time of each job

4.

Also, long running jobs may starve for the CPU when there is a steady supply of short jobs

