



Module 13

Operating Systems Two



Module Thirteen

- Operating Systems - Part Two
- In this presentation, we are going to talk about :
- Job Scheduling



Overview

- Previously we talked about:
- File Processing

Now: Job Scheduling

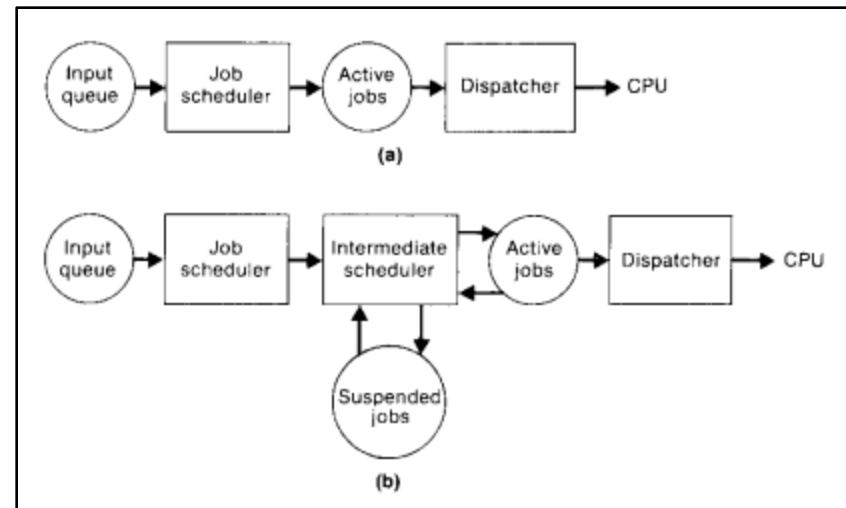


Job Scheduling

- How to get the Work done efficiently
- Goals:
 - Quick Response Time
 - Fast Turnaround
 - Processor Efficiency - Throughput
- Input Queue
 - Active Jobs
 - Suspended - rolled out - jobs

Types of Scheduling

- Long-term
 - performed when new process is created
- Medium-term
 - swapping
- Short-term
 - which ready process to execute next
 - DISPATCHING





Long-Term Scheduling

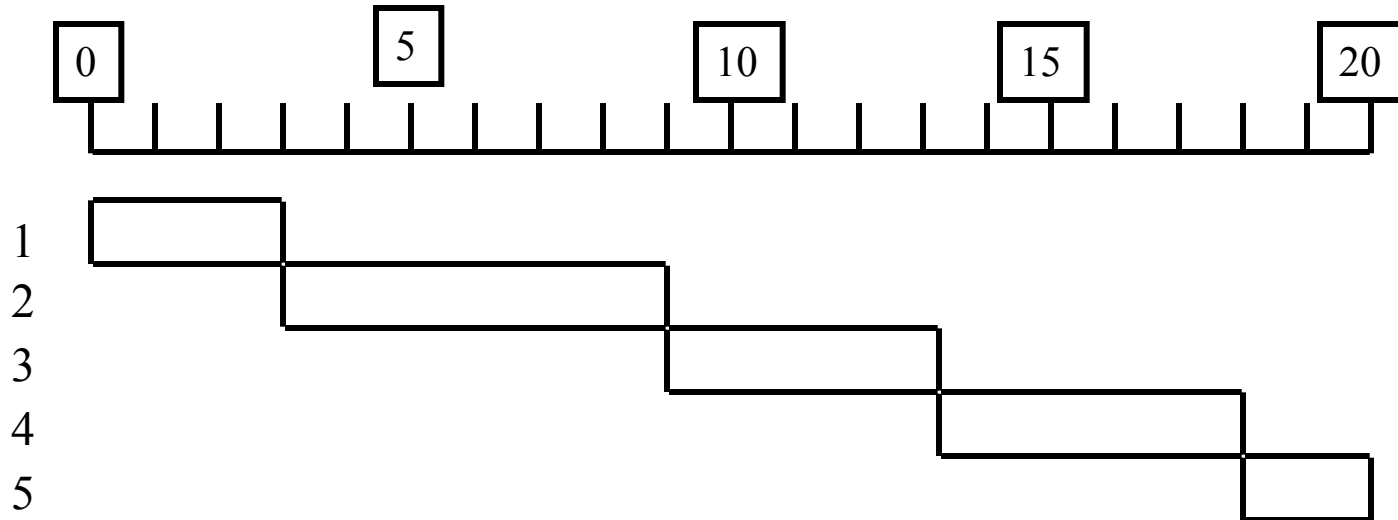
- Determines which programs are admitted to the system for processing.
- Controls the degree of multiprogramming.
- More processes, smaller percentage of time each process is executed.



Priorities

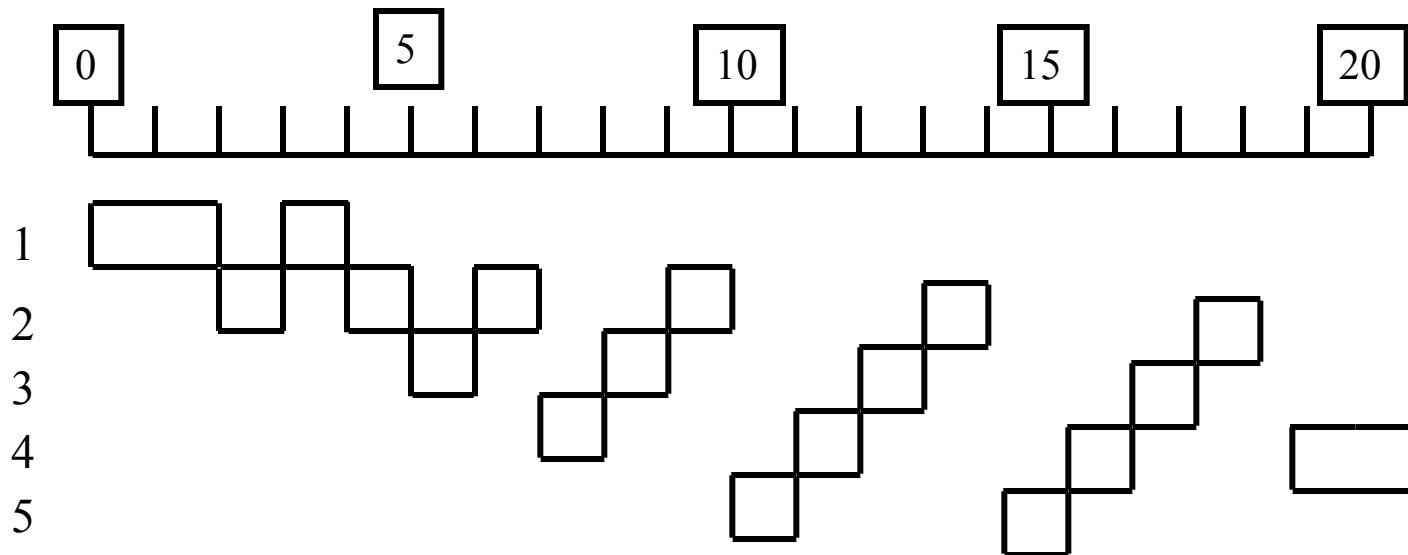
- Scheduler will always choose a process of higher priority over one of lower priority
- Have multiple ready queues to represent each level of priority
- Lower-priority may suffer starvation
 - allow a process to change its priority based on its age or execution history

First-Come-First-Served (FCFS)



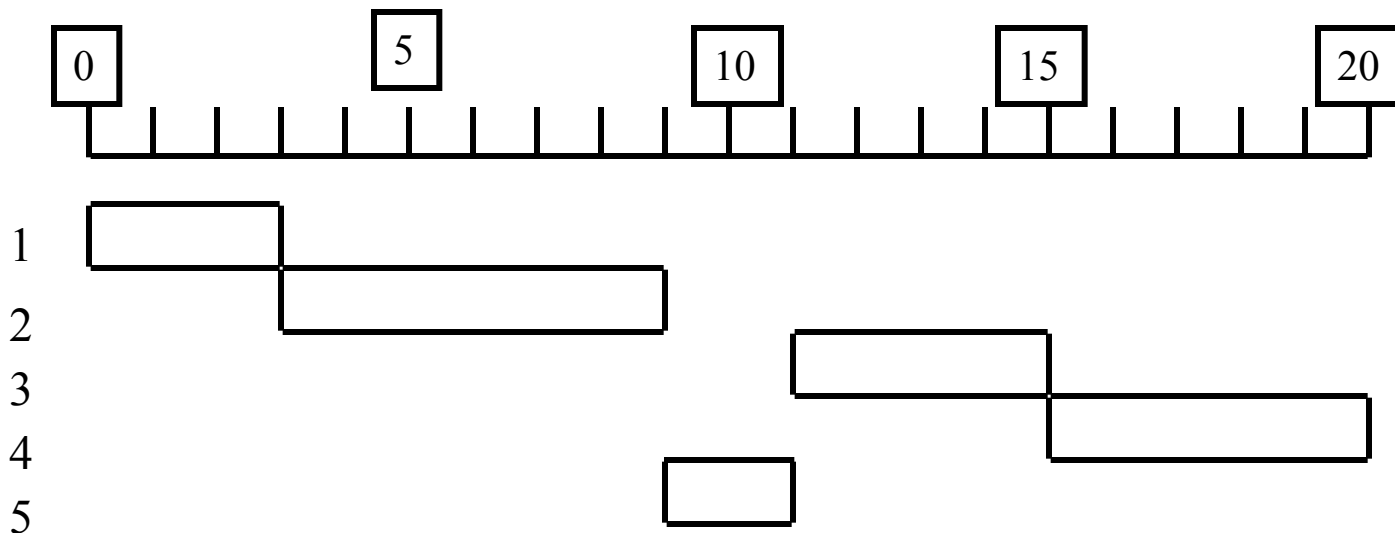
- Each process joins the Ready queue
- When the current process ceases to execute, the oldest process in the Ready queue is selected
- A short process may have to wait a very long time before it can execute.
- Favors CPU-bound processes; I/O processes have to wait until CPU-bound process completes.

Round-Robin



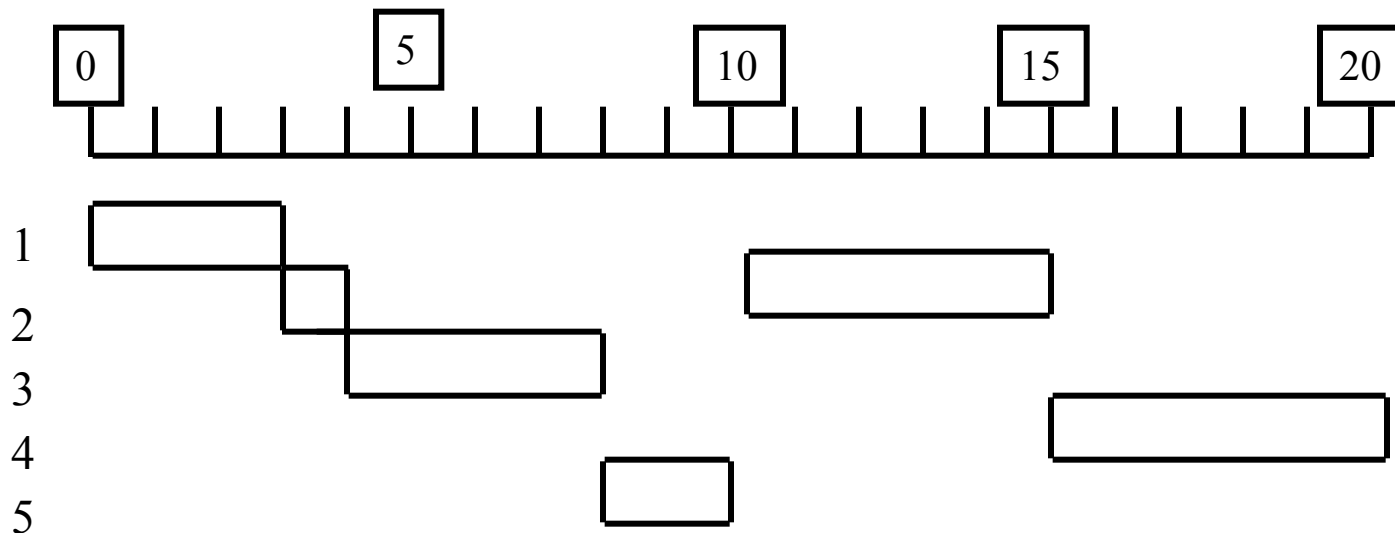
- Uses preemption based on a clock. (Timer Interrupt)
- An amount of time is determined that allows each process to use the processor for that length of time.

Shortest Process Next



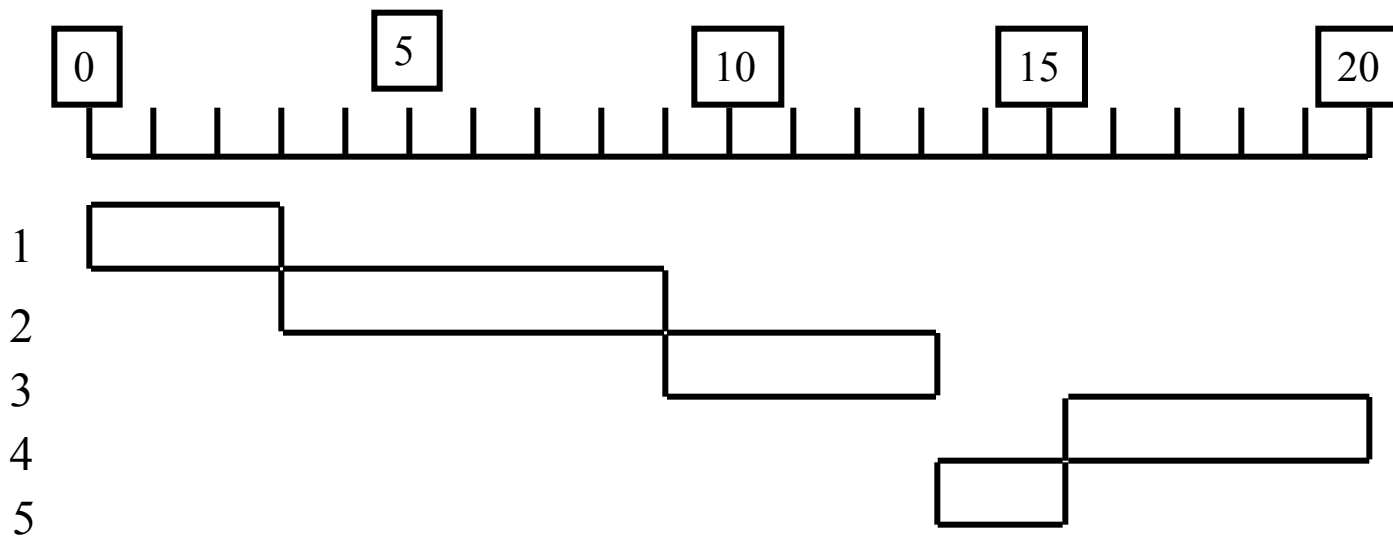
- Process with shortest expected processing time is selected next
- Short process jumps ahead of longer processes
- Predictability of longer processes is reduced.
- If estimated time for process not correct, the operating system may abort it.
- Possibility of starvation for longer processes.

Shortest Remaining Time



- Preemptive version of shortest process next policy.
- Must estimate processing time.

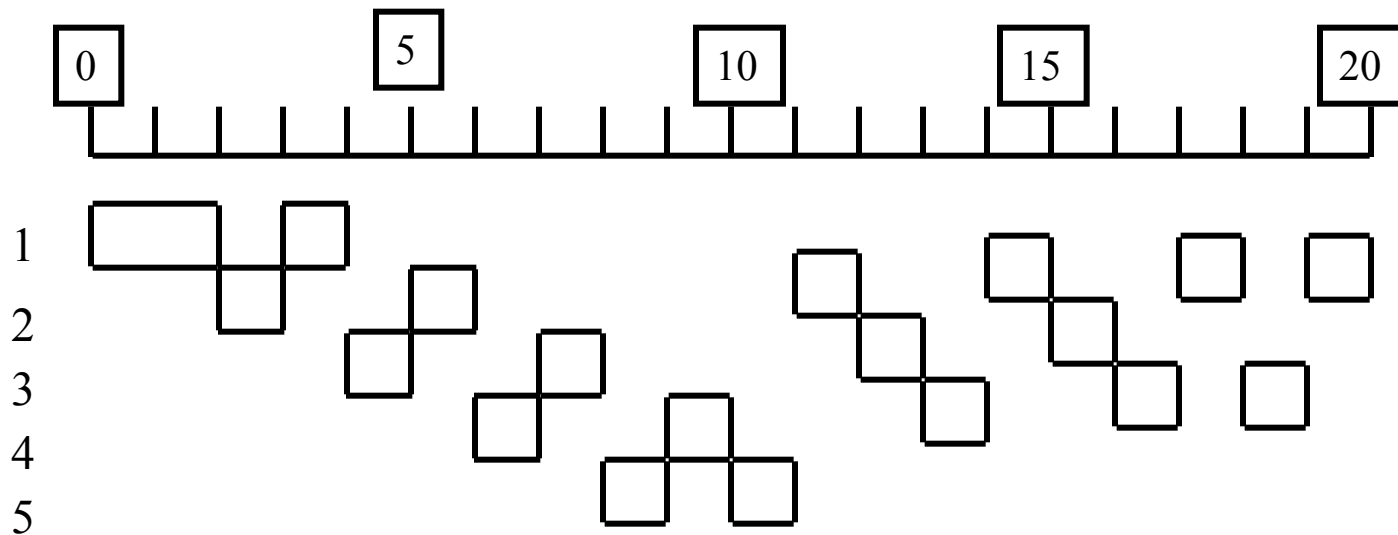
Highest Response Ratio Next



- Choose next process with the lowest ratio:

$$\frac{\text{time spent waiting} + \text{expected service time}}{\text{expected service time}}$$

Feedback



- Penalize jobs that have been running longer.
- Don't know remaining time process needs to execute.



Fair Scheduling

- User's application runs as a collection of processes (threads).
- User is concerned about the performance of the application.
- Need to make scheduling decisions based on groups of processes.
- Fairness - give equal and fair access to all processes.
- Differential responsiveness - discriminate between different classes of jobs.
- Efficiency - maximize throughput, minimize response time, and accommodate as many uses as possible



Summary

- Operating Systems Part Two
Job Scheduling

Next: Resource Allocation