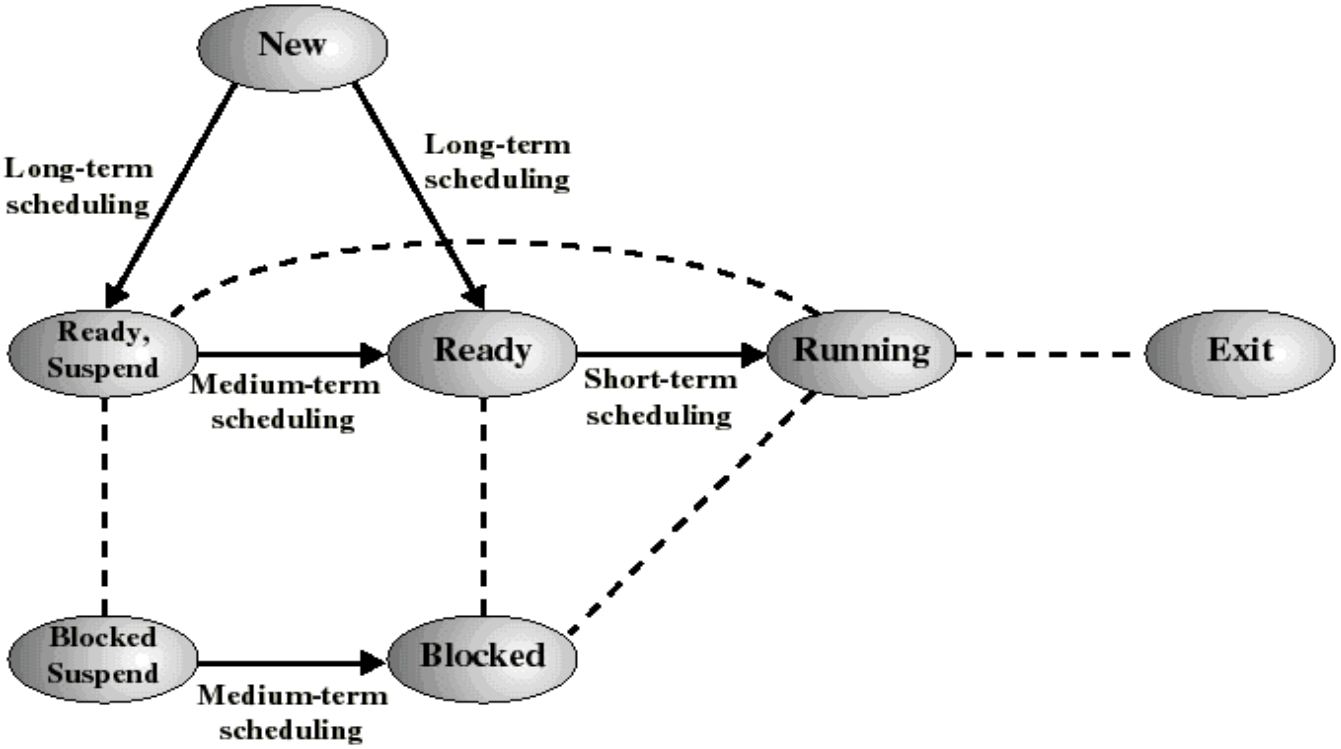
Process Scheduling

Unit 1.2-Chapter 2

CPU Scheduling

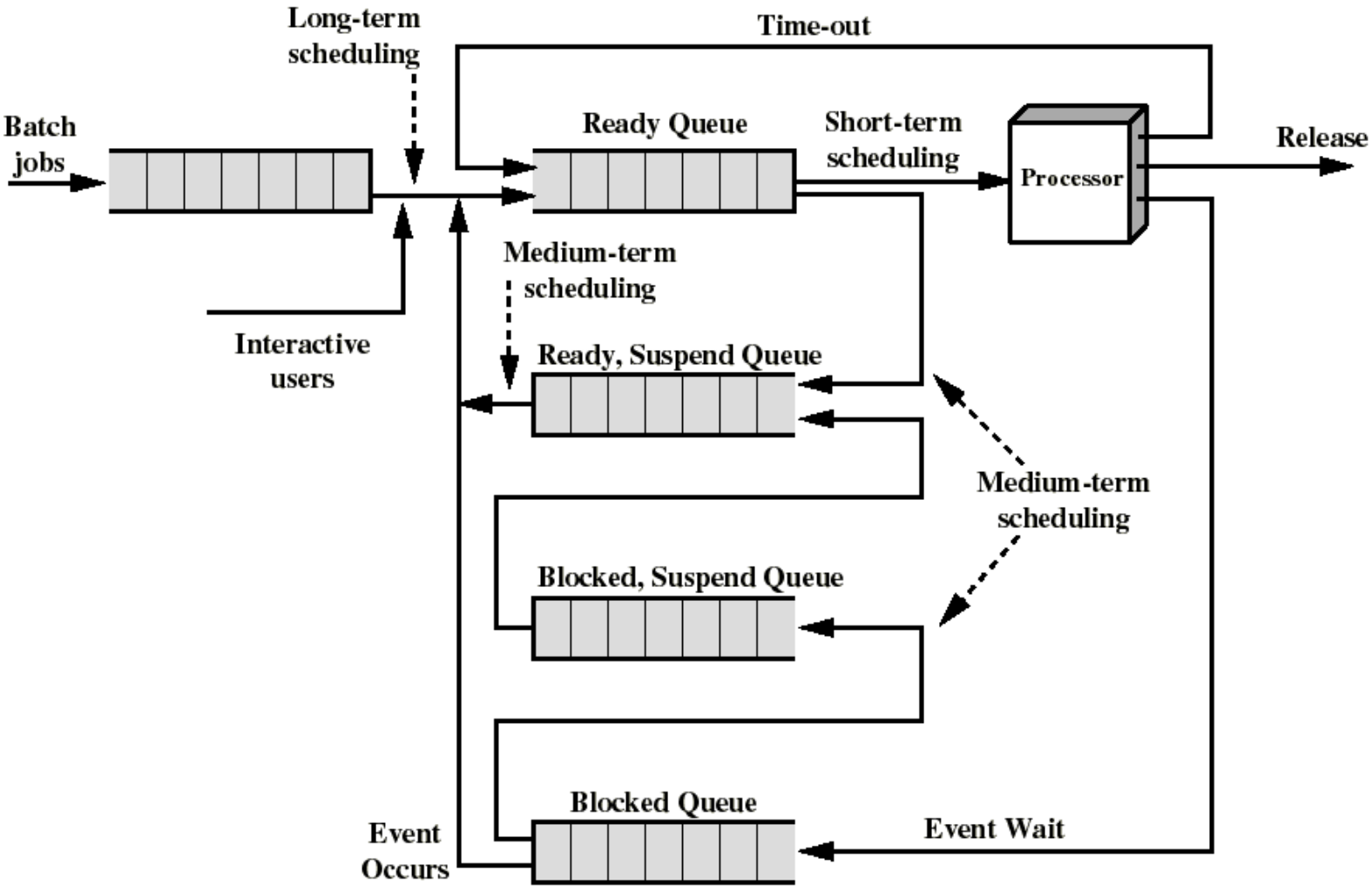
- Scheduling the processor among all ready processes
- The goal is to achieve:
  - High processor utilization
  - High throughput
    - number of processes completed per of unit time
  - Low response time
    - time elapsed from the submission of a request until the first response is produced

Classification of Scheduler



- Long-term: which process to admit?
- Medium-term: which process to swap in or out?
- Short-term: which ready process to execute next?

Queuing Diagram for Scheduling



Long-Term Scheduler

- Determines which programs are admitted to the system for processing
- Controls the degree of multiprogramming
- Attempts to keep a balanced mix of processor-bound and I/O-bound processes
  - CPU usage
  - System performance

Medium-Term Scheduler

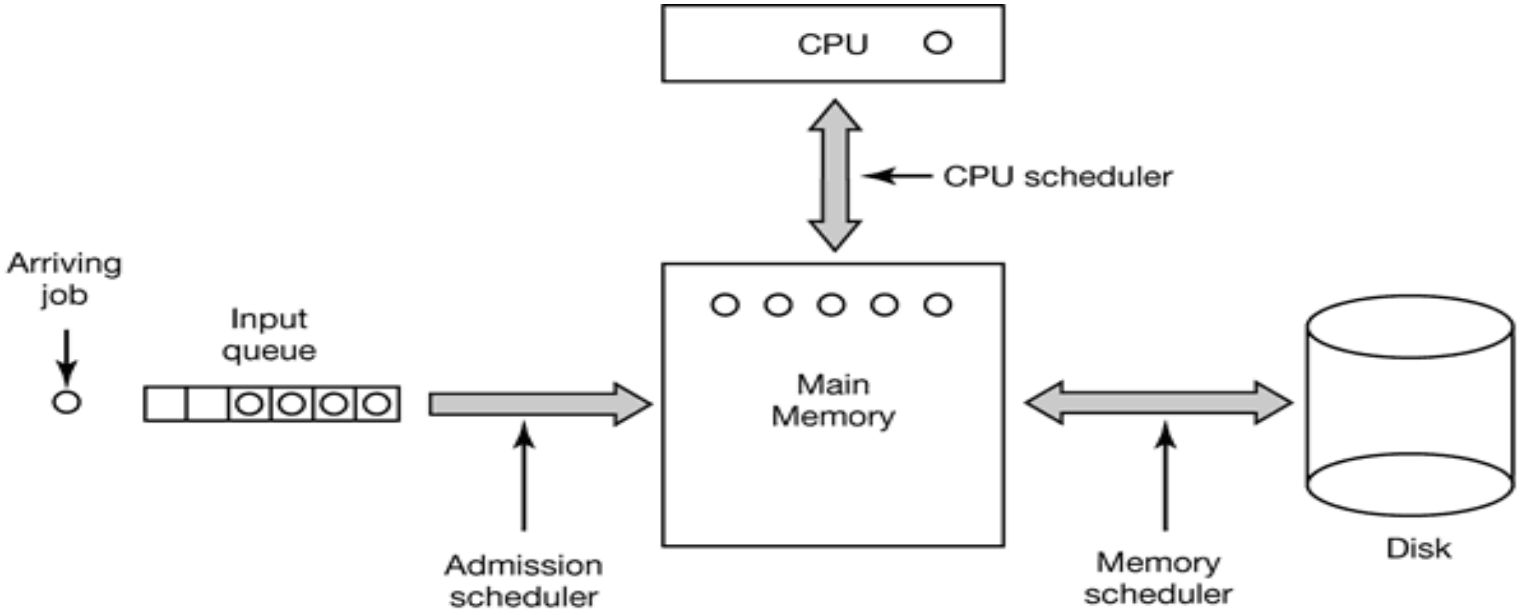
- Makes swapping decisions based on the current degree of multiprogramming
  - Controls which remains resident in memory and which jobs must be swapped out to reduce degree of multiprogramming

Short-Term Scheduler

- Selects from among ready processes in memory which one is to execute next
  - The selected process is allocated the CPU
- It is invoked on events that may lead to choose another process for execution:
  - Clock interrupts
  - I/O interrupts
  - Operating system calls and traps
  - Signals

Three level scheduling

- From a certain perspective, batch systems allow scheduling at three different levels, as illustrated in Fig
- As jobs arrive at the system, they are initially placed in an input queue stored on the disk.



Three level Scheduling

- The **admission scheduler (long term scheduler)** decides which jobs to admit to the system. The others are kept in the input queue until they are selected.
- A typical algorithm for admission control might be to look for a mix of compute-bound jobs and I/O-bound jobs. Alternatively, short jobs could be admitted quickly whereas longer jobs would have to wait.
- The admission scheduler is free to hold some jobs in the input queue and admit jobs that arrive later if it so chooses.

Three level scheduling(Cont)

- Once a job has been admitted to the system, a process can be created for it and it can contend for the CPU.
- However, it might well happen that the number of processes is so large that there is not enough room for all of them in memory. In that case, some of the processes have to be swapped out to disk.
- The second level of scheduling is deciding which processes should be kept in memory and which ones kept on disk.
- We will call this scheduler the **memory scheduler (short term schedular)**, since it determines which processes are kept in memory and which on the disk.

Three level of scheduling(cont.)

- The third level of scheduling is actually picking one of the ready processes in main memory to run next.
- Often this is called the **CPU scheduler** and is the one people usually mean when they talk about the "scheduler."
- Any suitable algorithm can be used here, either preemptive or non-preemptive.

Classification of Scheduling(CPU) Algorithms

- The selection function determines which ready process is selected next for execution
- The decision mode specifies the instants in time the selection function is exercised
  - **Nonpreemptive**
    - Once a process is in the running state, it will continue until it terminates or blocks for an I/O
  - **Preemptive**
    - Currently running process may be interrupted and moved to the Ready state by the OS
    - Prevents one process from monopolizing the processor

Scheduling Criteria

- **User-oriented criteria**
  - Response Time: Elapsed time between the submission of a request and the receipt of a response
  - Turnaround Time: Elapsed time between the submission of a process to its completion
- **System-oriented criteria**
  - Processor utilization
  - Throughput: number of process completed per unit time
  - fairness

Scheduling Algorithms

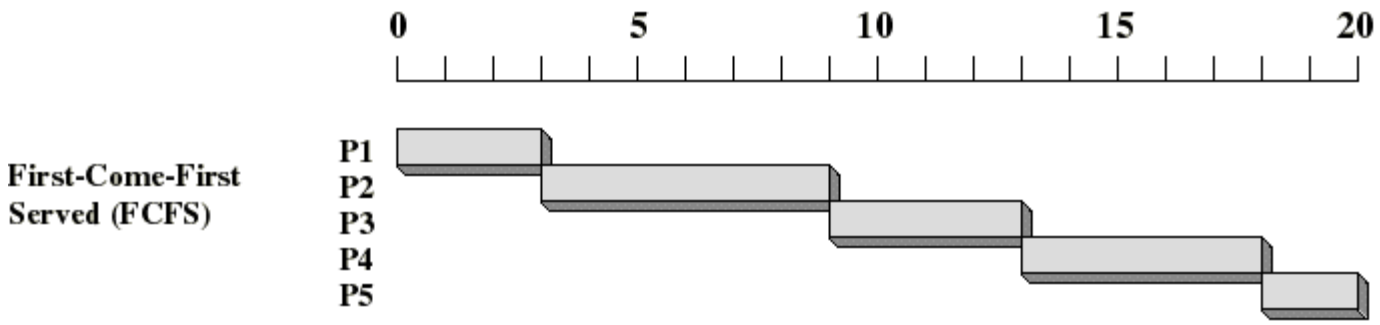
- First-Come, First-Served(FCFS)
- Shortest-Job-First(SJF) Scheduling
  - Also referred to as Shortest Process Next(SPN)
- Priority Scheduling
- **Round-Robin Scheduling**
- Multilevel Queue Scheduling

Process Mix Example

Process	Arrival Time	Service Time
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2

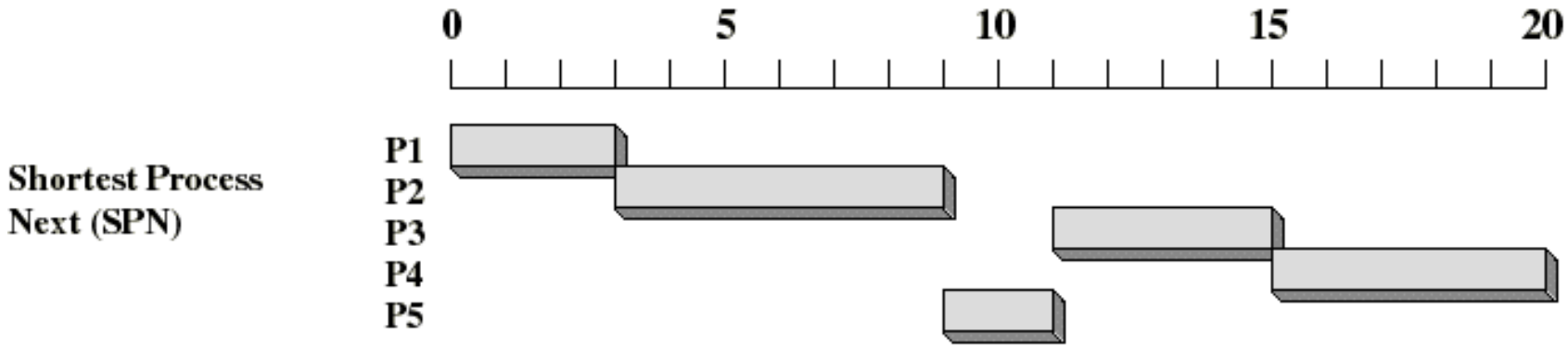
Service time = total processor time needed in one (CPU-I/O) cycle
Jobs with long service time are CPU-bound jobs and are referred to as "long jobs"

First Come First Served (FCFS)



- Selection function: the process that has been waiting the longest in the ready queue (hence, FCFS)
- Decision mode: **non-preemptive**
  - a process runs until it blocks for an I/O

Shortest Job First (Shortest Process Next)

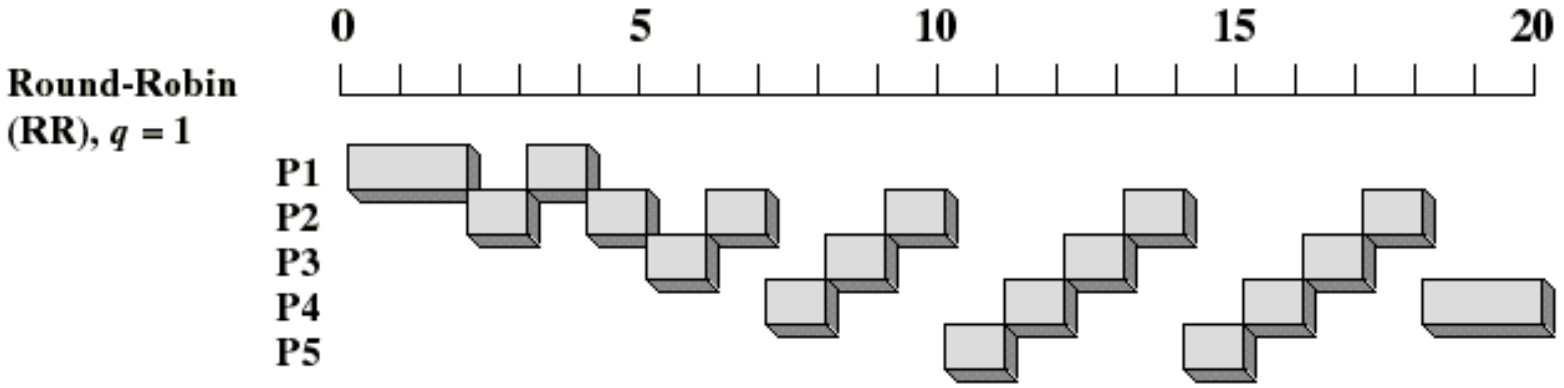


- Selection function: the process with the shortest expected CPU burst time
  - I/O-bound processes will be selected first
- Decision mode: **non-preemptive**
- The required processing time, i.e., the CPU burst time, must be estimated for each process

Priorities

- Implemented by having multiple ready queues to represent each level of priority
- Schedule the process of a higher priority over one of lower priority
- Lower-priority may suffer starvation
- To alleviate starvation allow dynamic priorities
  - The priority of a process changes based on its age or execution history

Round-Robin



- Selection function: same as FCFS
  - Decision mode: preemptive

- a process is allowed to run until the time slice period (quantum, typically from 10 to 100 ms) has expired
- a clock interrupt occurs and the running process is put on the ready queue

RR Time Quantum

- Quantum must be substantially larger than the time required to handle the clock interrupt and dispatching
- Quantum should be larger then the typical interaction
  - but not much larger, to avoid penalizing I/O bound processes

Round Robin: critique

- Still favors CPU-bound processes
  - An I/O bound process uses the CPU for a time less than the time quantum before it is blocked waiting for an I/O
  - A CPU-bound process runs for all its time slice and is put back into the ready queue
    - May unfairly get in front of blocked processes

Round Robin Variants

- Weighted RR
- RR with variable quanta