VNU–HCMUS

FACULTY OF INFORMATION TECHNOLOGY

Chapter 3

CPU Scheduling

Instructor: **Vu Thi My Hang, Ph.D.**

TA/Lab Instructor: **Le Quoc Hoa, M.Sc.**

Plan

- Basic concepts
- Scheduling algorithms

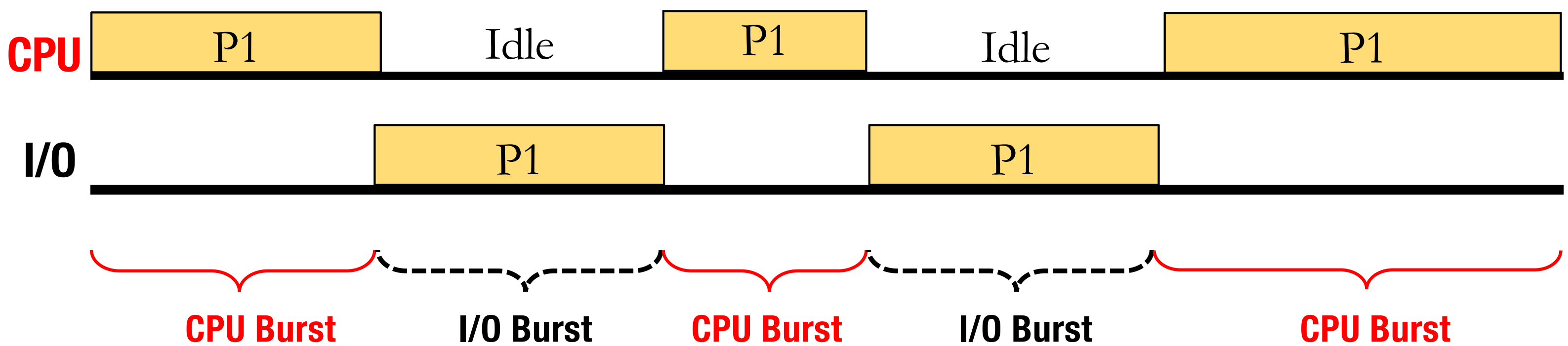
Plan

- **Basic concepts**
- Scheduling algorithms

Basic concepts

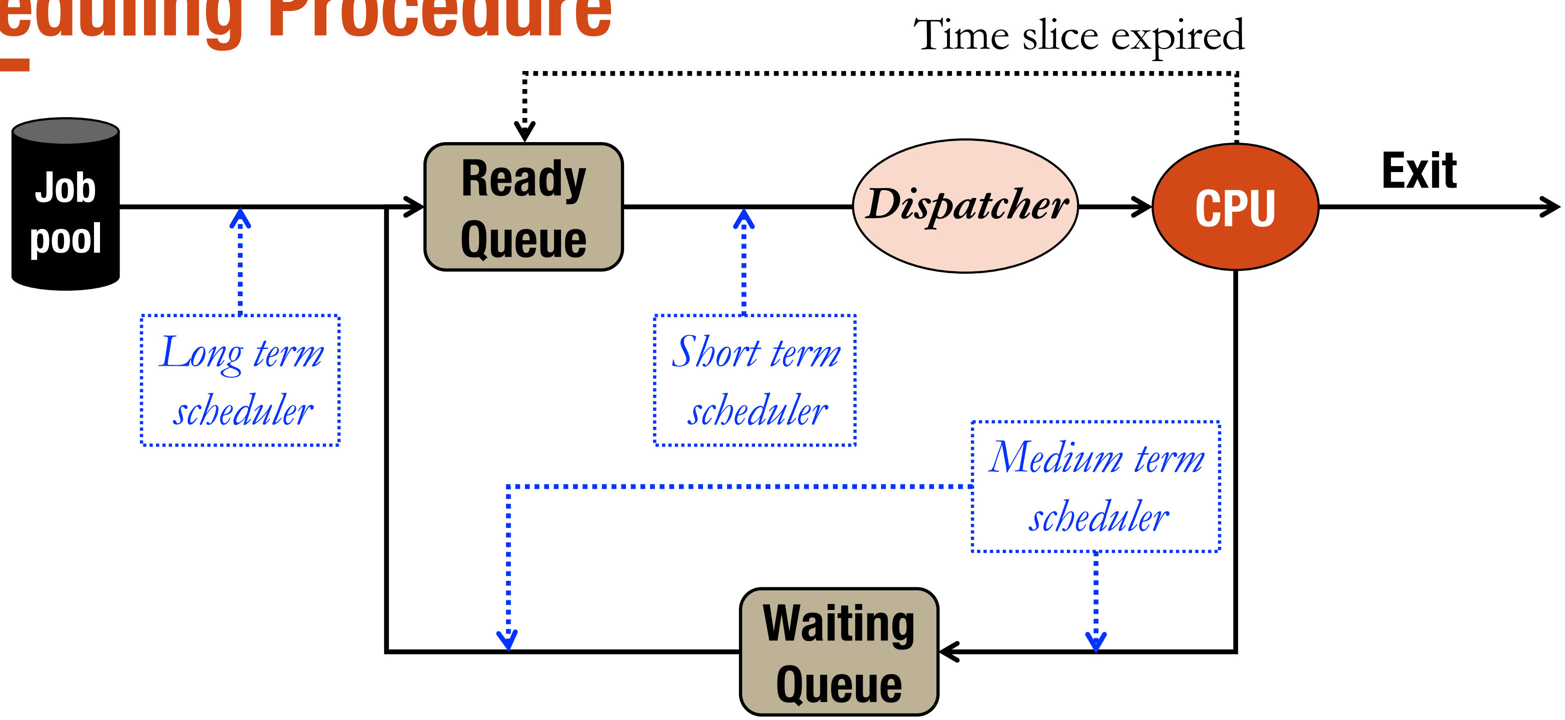
CPU – I/O Burst Cycle (Chu kỳ CPU và Chu kỳ Nhập/Xuất)

- A process execution alternates between:
  - ✓ CPU Burst performing calculations
  - ✓ I/O Burst performing I/O operations



Basic concepts

Scheduling Procedure

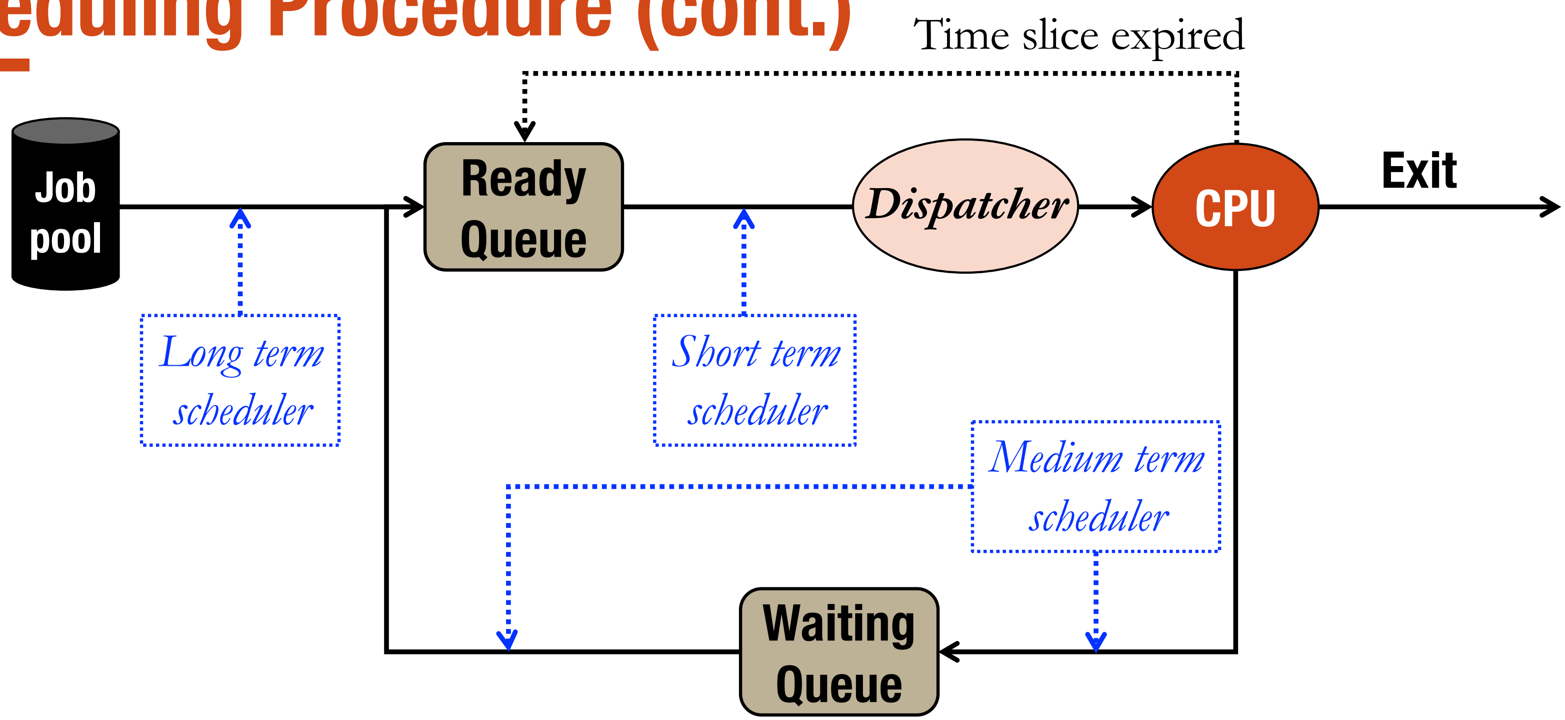


Scheduler (Bộ định thời)

Scheduler is an OS component involving scheduling activities, which consists of:

✓ *Long term scheduler* (Job scheduler) selects jobs from the job pool and loads them into the memory (ready queue) for execution

Scheduling Procedure (cont.)



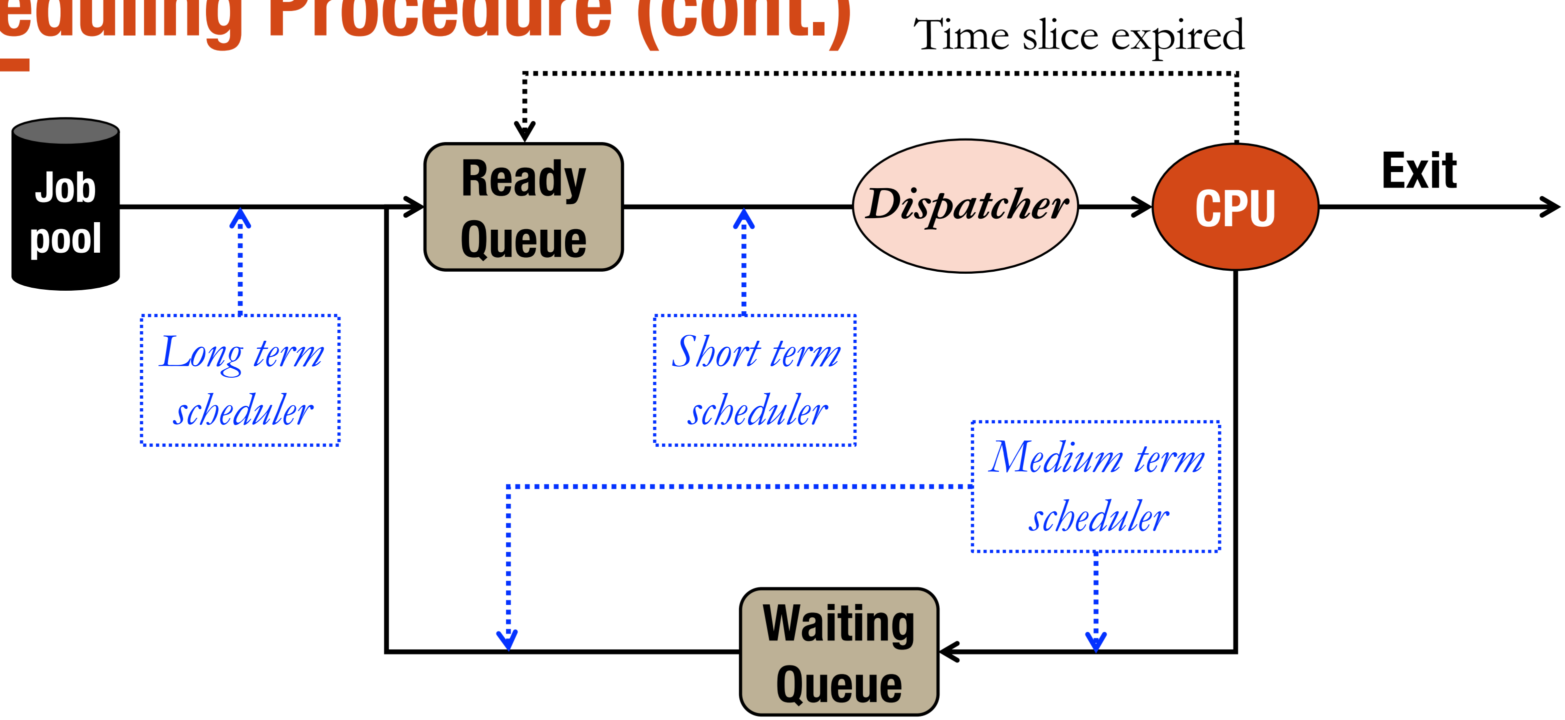
Scheduler

Scheduler is an OS component involving scheduling activities, which consists of:

✓ *Short term scheduler* (CPU scheduler) selects a process to be executed among ready processes

Basic concepts

Scheduling Procedure (cont.)

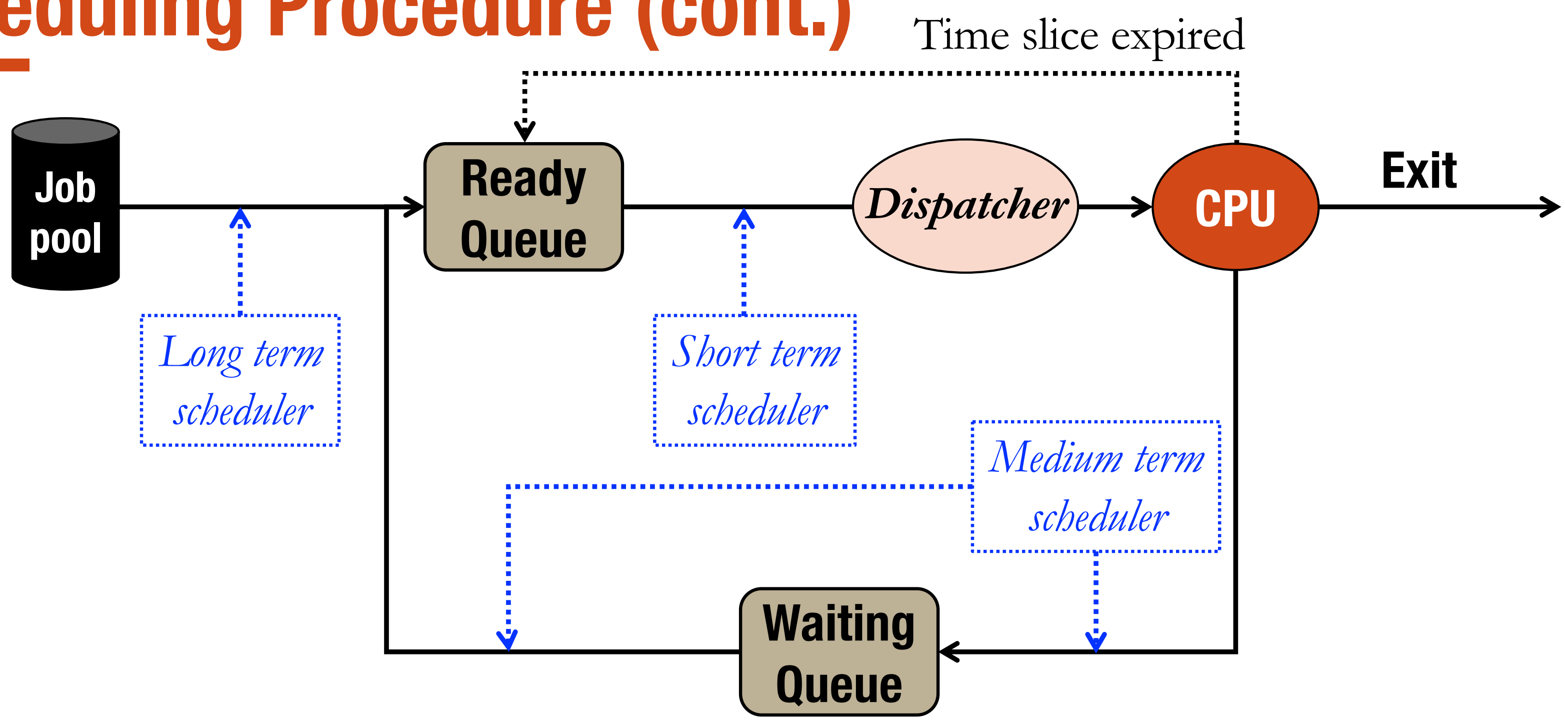


Scheduler

Scheduler is an OS component involving scheduling activities, which consists of:

✓ *Medium-term scheduler* handles swapped in and swapped out processes

Scheduling Procedure (cont.)



Dispatcher (Bộ điều phối)

Allocates CPU to the process selected by Short term scheduler

Basic concepts

Scheduling criteria

- Fairness and efficiency (Công bằng và Hiệu quả)
- Maximize CPU utilization (Tối đa hoá sử dụng CPU)
- Maximize throughput (Cực đại hoá thông lượng)
  - ✓ **Throughput** is the number of processes completed per unit time
- Minimize turnaround time, waiting time, and response time
  - ✓ **Turnaround time** is the interval from the time of process submission to the time of process completion (Thời gian lưu lại hệ thống)
  - ✓ **Waiting time** is the interval in the ready queue (Thời gian đợi)
  - ✓ **Response time** is the interval from the time of process submission to the time of producing the first response (Thời gian hồi đáp/đáp ứng)

Basic concepts

When to schedule ...

- New process created
  - ✓ Execute parent or child process?
- Process terminates
- Process blocked for an I/O operation or an event
- Interrupt occurring
  - ✓ Completion of I/O operation
  - ✓ End of time slice

Which is the next process to execute?

Scheduling Types

(Điều phối độc quyền == Không thương lượng)

- In **Nonpreemptive scheduling**, the running process keeps the CPU until it **terminates** or switches to the **waiting state** (by an I/O completion or an event)

(Điều phối không độc quyền == Có thương lượng/hợp tác)

- In **Preemptive scheduling** (i.e., cooperative scheduling), the running process may be suspended by the OS because of a **higher priority process** arriving or its **time slice expired**

Plan

- Basic concepts
- **Scheduling algorithms**

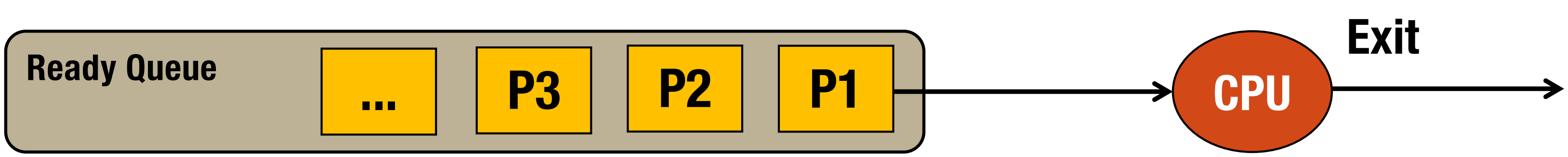
First-Come, First-Served (FCFS) Scheduling

- Nonpreemptive scheduling
- Pick the first process standing in ready queue for execution

☺ Simplicity

☹ Low performance

☹ Do not consider process priority



First-Come, First-Served (FCFS) Scheduling (cont.)

Process	Arrival Time	CPU Burst	TT	WT
P1	0	24	24	0
P2	1	5	28	23
P3	2	3	30	27
			AVG_{TT} = 27.33	AVG_{WT} = 16.67

Turnaround Time (TT) = (Completion Time – Arrival Time)

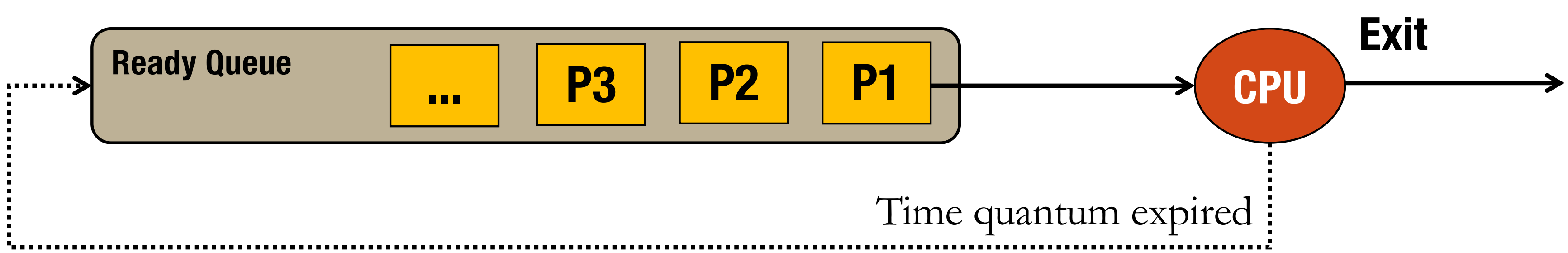
Waiting Time (WT) = (TT – CPU Burst)

E.g., TT(P3) = 32 – 2 = 30, WT(P3) = 30 – 3 = 27

(Điều phối xoay vòng)

Round-Robin (RR) Scheduling

- Preemptive scheduling
  - ✓ Especially used in timesharing systems
- Pick the first process standing in ready queue for execution within a given **time quantum q** (time slice)
  - ✓ After the time quantum, if the active process has not finished yet, it will move to the end of the ready queue for the next execution



Round-Robin (RR) Scheduling (cont.)

☺ Fairness

☺ Reduce waiting time of short CPU-burst processes

☹ Do not consider process priority

☹ How to define time quantum?

- ✓ Too large time quantum → Similar to FIFO Algorithm
- ✓ Too low time quantum → Too many context switches performed

Round-Robin (RR) Scheduling (cont.)

Process	Arrival Time	CPU Burst	TT	WT
P1	0	24	32-0 = 32	32-24 = 8
P2	1	5	16-1 = 15	15-5 = 10
P3	2	3	11-2 = 9	9-3 = 6
			AVG_{TT} = 18.66	**AVG_{WT} = 8**

Another method for calculating WT

WT(P1) = (11 – 4) + (16 – 15) = 7 + 1 = 8

WT(P2) = (4 – 1) + (15 – 8) = 3 + 7 = 10; WT(P3) = (8 – 2) = 6

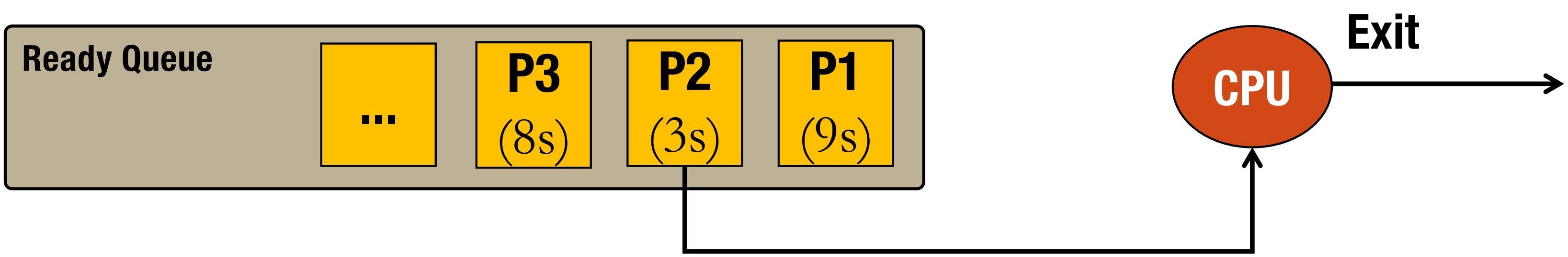
q = 4

P1	P2	P3	P1	P2	P1

0 4 8 11 15 16 32

Shortest-Job-First (SJF) Scheduling

- Nonpreemptive scheduling
  - ✓ Preemptive version: Shortest-Remaining-Time-Next (SRTN) Scheduling
- Pick the process having the smallest CPU-bursts time



- What if P4(2s) is put into the ready queue while P2 is running?
  - ✓ SJF: P2 continues its execution then CPU switches to P4
  - ✓ SRTN: P4 will be preempted immediately

Scheduling algorithms

Shortest-Job-First (SJF) Scheduling (cont.)

☺ Simplicity

☺ Reduce turnaround time and waiting time of "short" processes

☺ Increase throughput (i.e., more processes completed)

☹ Need to estimate CPU-burst of all processes

☹ Problem of **starvation**

- ✓ Processes having higher CPU-burst can be blocked indefinitely if smaller CPU-burst processes keep executing

Shortest-Remaining-Time-Next (SRTN) Scheduling

Process	Arrival Time	CPU Burst	TT	WT
P1	0	24	32-0 = 32	32-24 = 8
P2	1	5	9-1 = 8	8-5 = 3
P3	2	3	5-2 = 3	3-3 = 0
			AVG_{TT} = 14.33	**AVG_{WT} = 3.66**

Priority Scheduling (Điều phối với độ ưu tiên)

- Two versions: preemptive and nonpreemptive scheduling
  - ✓ Especially used in timesharing systems
- Each process is associated with a priority
- Pick the process having the highest priority in the ready queue
- What if a new process having higher priority than the currently executing process arrives?
  - ✓ Preemptive version: the running process will be stopped immediately
  - ✓ Nonpreemptive version: the running process finishes its task then the highest priority process will be allocated the CPU

Scheduling algorithms

Priority Scheduling (cont.)

☺ Take consideration of process priority

☹ Problem of **starvation**

- ✓ Processes having lower priority can be blocked indefinitely if higher priority processes keep executing
- ✓ Solutions for the starvation problem:
  - **Aging** (i.e., increase the priority of processes waiting for a long time in the ready queue)

Scheduling algorithms

Priority Scheduling (cont.)

Process	Arrival Time	CPU Burst	Priority	TT	WT
P1	0	24	3	32-0 = 32	32-24 = 8
P2	1	5	2	9-1 = 8	8-5 = 3
P3	2	3	1	5-2 = 3	3-3 = 0
	P3 > P2 > P1			AVG_{TT} = 14.33	AVG_{WT} = 3.66

Preemptive version

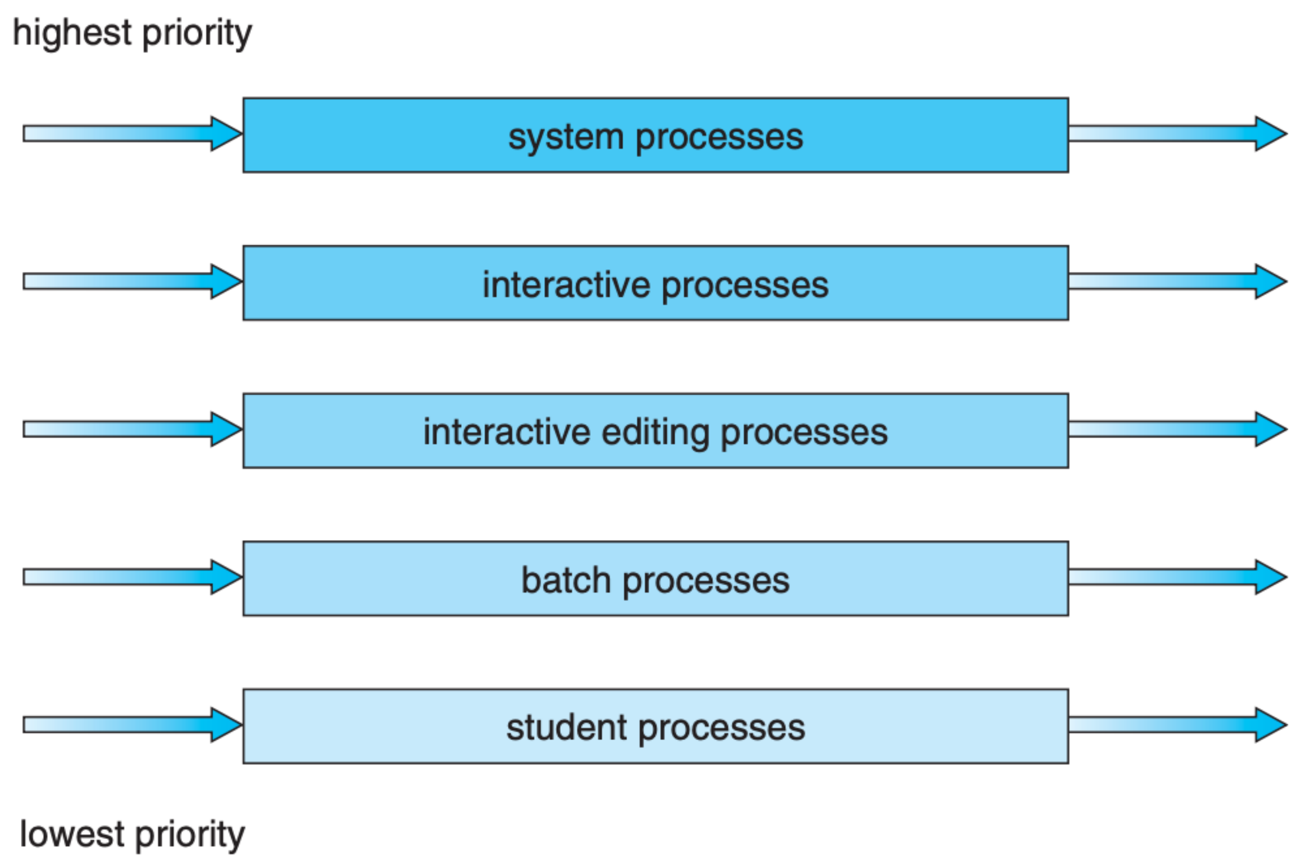
 (Điều phối với hàng đợi đa cấp)

Multilevel Priority Queue Scheduling

- Separate the ready queue into multiple priority queues
- Each priority queue is associated with a priority level
  - ✓ Each queue has its scheduling algorithms
- Each process is assigned to one queue
- Pick processes presenting in higher priority queue first
  - ✓ Process selection is based on the scheduling algorithm applied to this queue

Scheduling algorithms

Multilevel Queue Scheduling (cont.)



Scheduling algorithms

And other ...

- Guaranteed scheduling
- Lottery scheduling
- Fair-Share scheduling