ĐIỀU PHỐI TIẾN TRÌNH
CPU SCHEDULING

ThS. Nguyễn Thị Hải Bình

Khoa CNTT, ĐH Giao thông vận tải

Email: calmseahn@gmail.com

Website: calmseahn.weebly.com

CONTENTS

- CPU Scheduling
- Basic concepts
- Scheduling criteria
- Scheduling algorithms

CPU SCHEDULING

- In a single-processor system, only one process can be run at a time
- Simple computer system
  - When a process is *waiting*, CPU just sits *idle*
- Multiprogramming system
  - Several processes are keep in memory at one time
  - When a process have to *wait*, the OS *gives the CPU to another process*
- CPU scheduling is central to operating-system design

BASIC CONCEPTS

- CPU – I/O Burst Cycle
- I/O-bound and CPU-bound program
- CPU scheduler
- Non-preemptive and preemptive scheduling
- Dispatcher

CPU – I/O BURST CYCLE

- A *CPU burst* is when CPU performs useful computation
- An *I/O burst* consists of I/O operations being performed
- *CPU – I/O burst cycle*
  - Process execution consists of a series of *cycles* of CPU execution and I/O waits

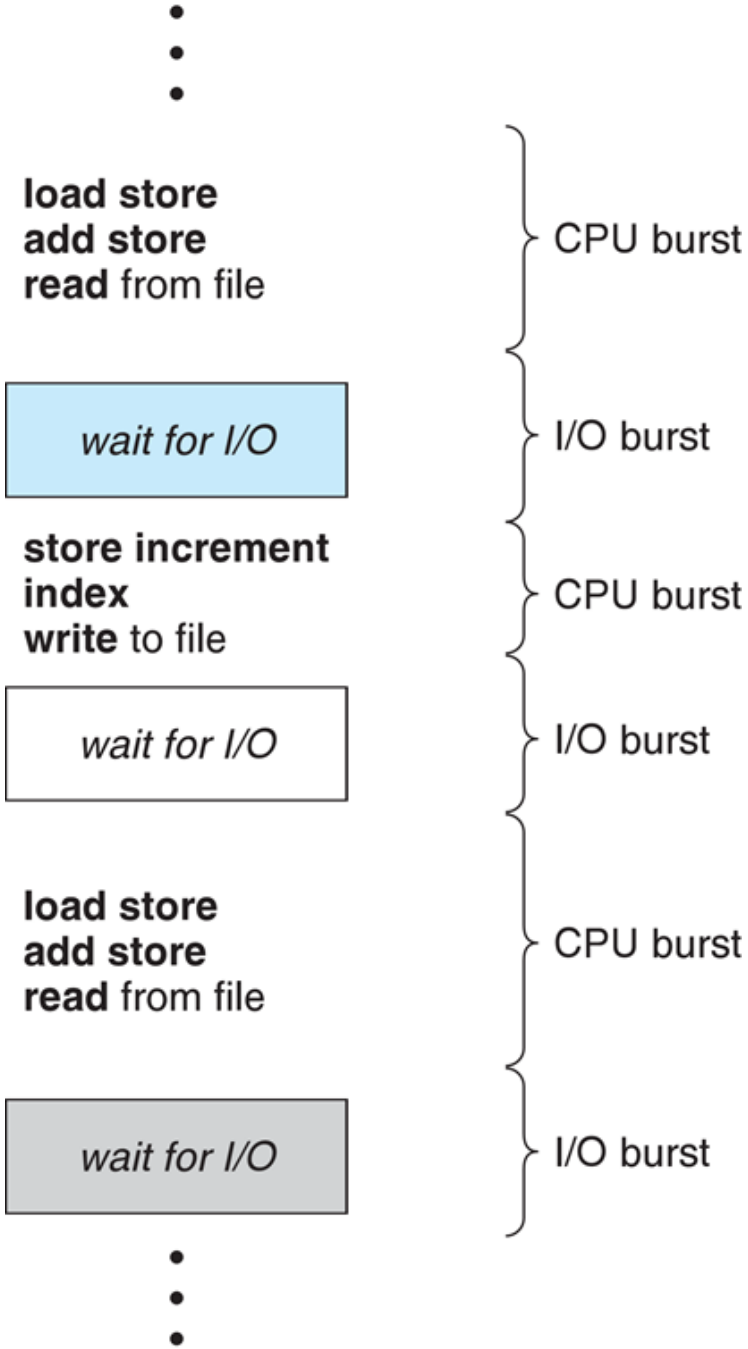


Figure 6.1 Alternating sequence of CPU and I/O bursts

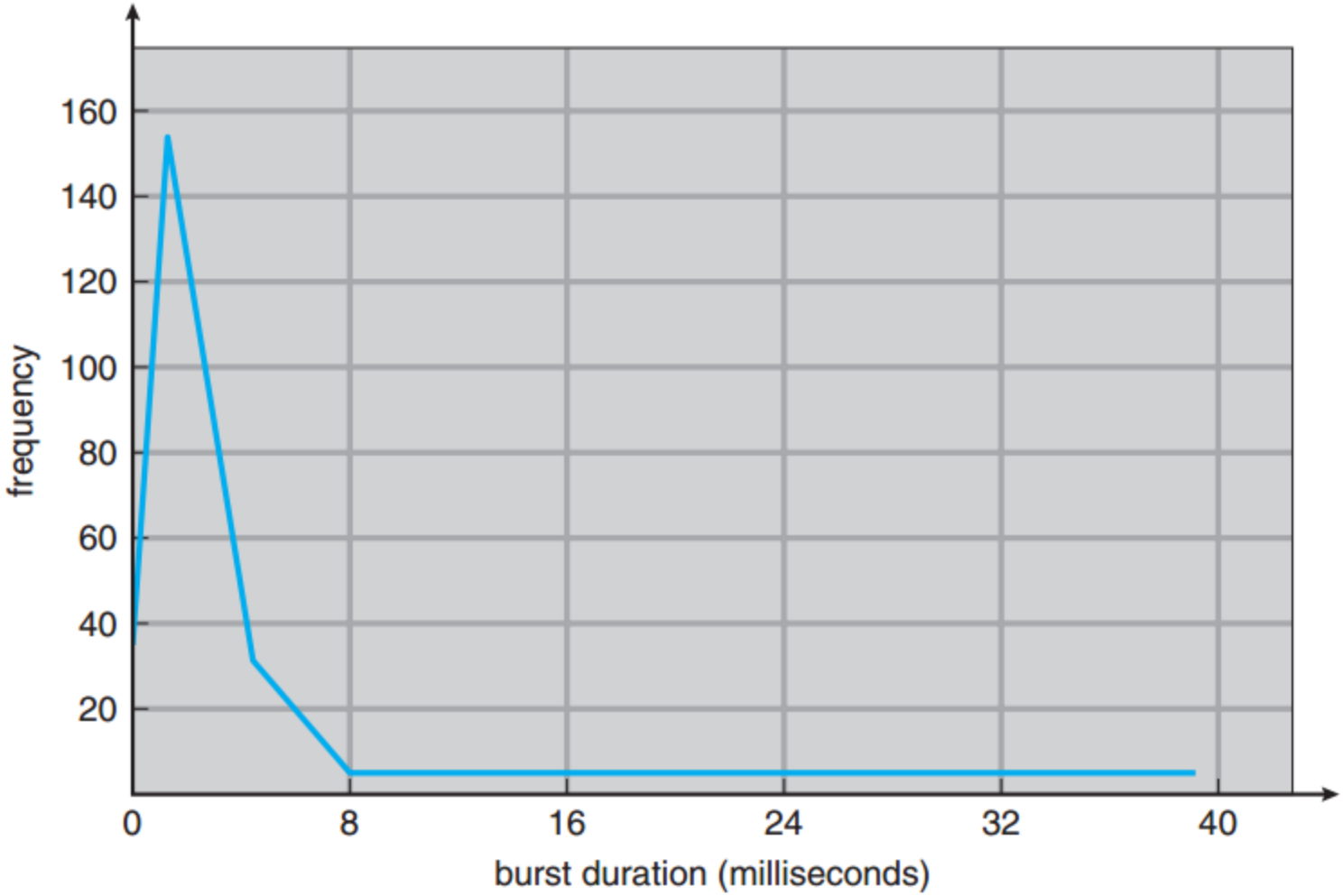


Figure 6.2 Histogram of CPU-burst durations.

I/O-BOUND AND CPU-BOUND PROGRAM

- I/O-bound program
  - Typically has many short CPU burst
- CPU-bound program
  - Might have a few long CPU burst

BỘ ĐIỀU PHỐI CPU

- Thuật ngữ:
  - CPU scheduler
  - Short-term scheduler (bộ điều phối ngắn hạn)
- Nhiệm vụ
  - *Selects* one of the processes in the *ready queue* and *allocates* the CPU to that process

BỘ ĐIỀU PHỐI CPU

- Bộ điều phối hoạt động khi
  1. Tiến trình chuyển từ trạng thái running sang trạng thái waiting
  2. Tiến trình chuyển từ trạng thái running sang trạng thái ready
  3. Tiến trình chuyển từ trạng thái waiting sang trạng thái ready
  4. Tiến trình kết thúc

NON-PREEMPTIVE SCHEDULING

- Thuật ngữ
  - Điều phối không tiếm quyền
  - Điều phối độc quyền
- Nguyên lí
  - Tiến trình giữ CPU cho tới khi tiến trình chuyển sang trạng thái chờ hoặc kết thúc → *Tiến trình tình nguyện từ bỏ CPU*
- Ví dụ: Windows 3.x, Macintosh OS

PREEMPTIVE SCHEDULING

- Thuật ngữ
  - Điều phối có tiếm quyền
  - Điều phối không độc quyền
- Nguyên lí
  - Khi tiến trình chuyển từ trạng thái *running hoặc waiting sang ready*, bộ điều phối sẽ quyết định *tiếp tục cấp phát CPU cho tiến trình* hay sẽ *chuyển CPU cho tiến trình khác*
- Ví dụ: Windows 95 ↑, Mac OS X
- Vấn đề:
  - Tiến trình 1 đang cập nhật dữ liệu và bị mất CPU
  - Tiến trình 2 đọc dữ liệu đang cập nhật

DISPATCHER

- Thuật ngữ
  - Bộ phân phối
  - Bộ điều vận
- Nhiệm vụ
  - Cấp quyền sử dụng CPU cho tiến trình đã được lựa chọn bởi bộ điều phối ngắn hạn (short-term scheduler)
- Chức năng
  - Chuyển đổi ngữ cảnh (context switch)
  - Chuyển về user mode
  - Thực thi tiến trình theo trạng thái đã lưu
- Dispatch latency (độ trễ điều phối)
  - Thời gian dừng một tiến trình và bắt đầu một tiến trình khác
- Yêu cầu: tốc độ nhanh

TIÊU CHUẨN ĐIỀU PHỐI (SCHEDULING CRITERIA)

- CPU Utilization (Khả năng tận dụng CPU)
  - Là một số nằm trong đoạn [0%, 100%]
  - Trên thực tế, CPU utilization thường nằm trong khoảng từ 40% (*hệ thống chịu tải thấp – lightly loaded system*) đến 90% (*hệ thống chịu tải cao – heavily loaded system*)
- Throughput (thông lượng)
  - Số lượng tiến trình được hoàn thành trong một đơn vị thời gian
- Turnaround time (thời gian hoàn thành)
  - Là khoảng thời gian từ khi tiến trình vào hệ thống tới khi ra khỏi hệ thống
  - Bao gồm: thời gian tải vào bộ nhớ, thời gian chờ trong ready queue, thời gian thực hiện (running), thời gian thực hiện vào/ra

TIÊU CHUẨN ĐIỀU PHỐI (SCHEDULING CRITERIA)

- Waiting time (Thời gian chờ)
  - Tổng thời gian tiến trình nằm trong hàng đợi ready
- Response time (Thời gian đáp ứng)
  - Là khoảng thời gian từ khi tiến trình nhận được một yêu cầu cho đến khi bắt đầu đáp ứng yêu cầu đó

CÁC THUẬT TOÁN ĐIỀU PHỐI SCHEDULING ALGORITHMS

- First-Come, Fisrt-Server (FCFS) scheduling
- Shortest-Job-First (SJF) scheduling
- Priority scheduling
- Round-Robin scheduling
- Multilevel queue scheduling
- Multilevel feedback queue scheduling

FIRST-COME FIRST-SERVER (FCFS)

- Tiến trình nào yêu cầu sử dụng CPU trước sẽ được cấp phát trước
- Điều phối không tiếm quyền (non-preemptive)
- Ưu điểm
  - Thuật toán đơn giản
- Nhược điểm
  - Hiệu quả của thuật toán phụ thuộc vào thứ tự của các tiến trình trong hàng đợi
  - Thông thường, thời gian chờ đợi trung bình cao

FCFS: VÍ DỤ

- Giả sử có 3 tiến trình P1, P2, P3

Process	Burst Time
P_1	24
P_2	3
P_3	3

- Giả sử 3 tiến trình nằm trong hàng đợi theo thứ tự P_1, P_2, P_3 → Biểu đồ Gantt (*Gantt chart*)

- Thời gian chờ trung bình là: $\frac{(0+24+27)}{3} = 17\ (ms)$

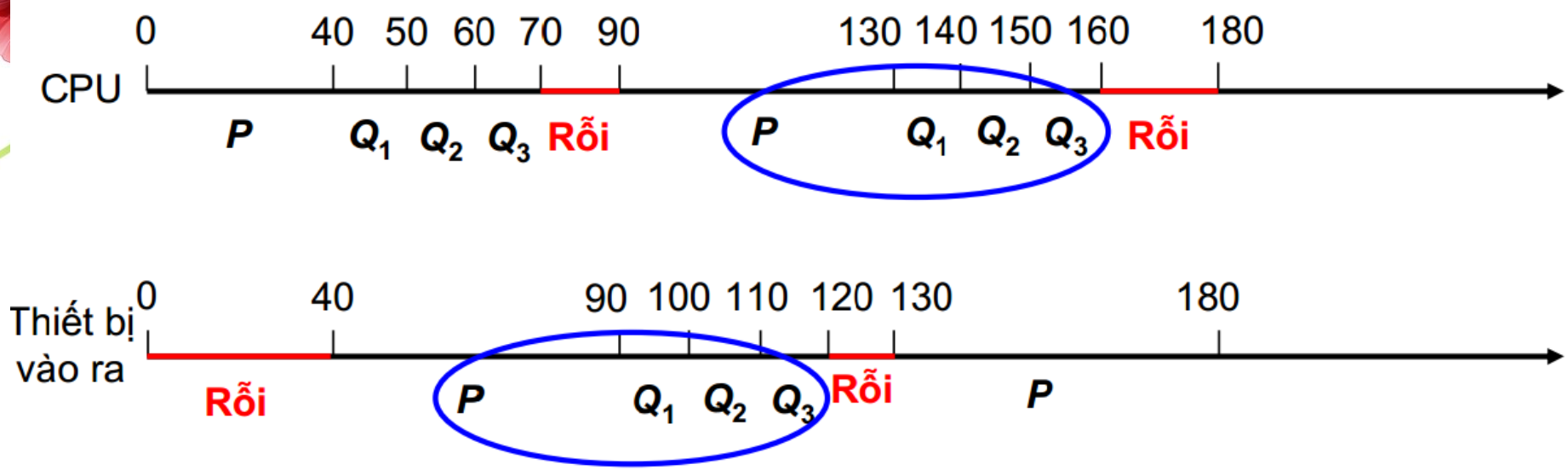
FCFS: VÍ DỤ

- Nếu 3 tiến trình nằm trong hàng đợi theo thứ tự P_2, P_3, P_0
- Ta có biểu đồ Gantt:

- Thời gian chờ trung bình là: 3ms

FCFS: CONVEY EFFECT

- Thuật ngữ: “Đoàn hộ tống”
- Xảy ra khi có một tiến trình *sử dụng nhiều thời gian CPU và vào ra* nằm ở đầu hàng đợi, và nhiều tiến trình *sử dụng ít thời gian CPU và vào ra* nằm ở phía sau
- CPU và thiết bị vào ra có nhiều thời gian rỗi
- Thời gian chờ trung bình của các tiến trình cao



- Tiến trình P có chu kỳ sử dụng CPU trong 40ms, và I/O trong 50ms
- Tiến trình Q_1, Q_2, Q_3 có chu kỳ sử dụng CPU trong 10ms và I/O trong 10ms
- Thứ tự hàng đợi: P, Q_1, Q_2, Q_3

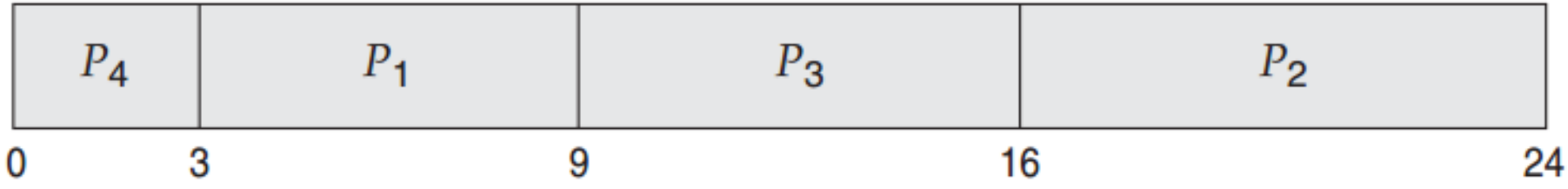
SHORTEST-JOB-FIRST (SJF)

- Mỗi tiến trình lưu trữ độ dài của phiên sử dụng CPU (CPU burst) tiếp theo
- Tiến trình nào yêu cầu ít thời gian CPU nhất sẽ được cấp phát CPU
- Nếu 2 tiến trình có cùng thời gian sử dụng CPU, FCFS sẽ được sử dụng
- SJF là tối ưu

SJF: VÍ DỤ

Process	Burst Time
P_1	6
P_2	8
P_3	7
P_4	3

Gantt chart



Thời gian chờ trung bình: (0 + 3 + 9 + 16)/4 = 7 ms

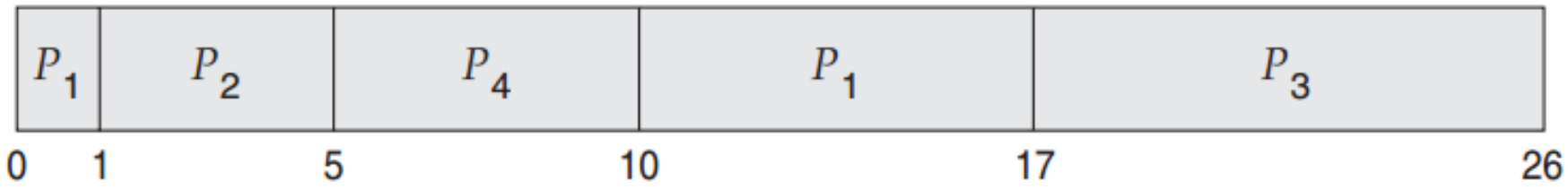
SJF: NON-PREEMPTIVE VS. PREEMPTIVE

- Non-preemptive
  - One CPU given to the process it cannot be preempted until completes its CPU burst
- Preemptive
  - If a new process arrives with CPU burst length less than remaining time of current executing process, preempt
  - Thuật ngữ:
    - Shortest-Remaining-Time-First (SRTF)
    - Shortest-Remaining-Next (SRN)

SRTF (PREEMPTIVE SJF): VÍ DỤ

Process	Arrival Time	Burst Time
P_1	0	8
P_2	1	4
P_3	2	9
P_4	3	5

Gantt chart



Thời gian chờ trung bình:

$$\frac{[(10-1)+(1-1)+(17-2)+(5-3)]}{4} = 6.5\ ms$$

NON-PREEMPTIVE SJF: VÍ DỤ

Process	Arrival Time	Burst Time
P_1	0	8
P_2	1	4
P_3	2	9
P_4	3	5

VÍ DỤ

Process	Arrival Time	Burst Time
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4

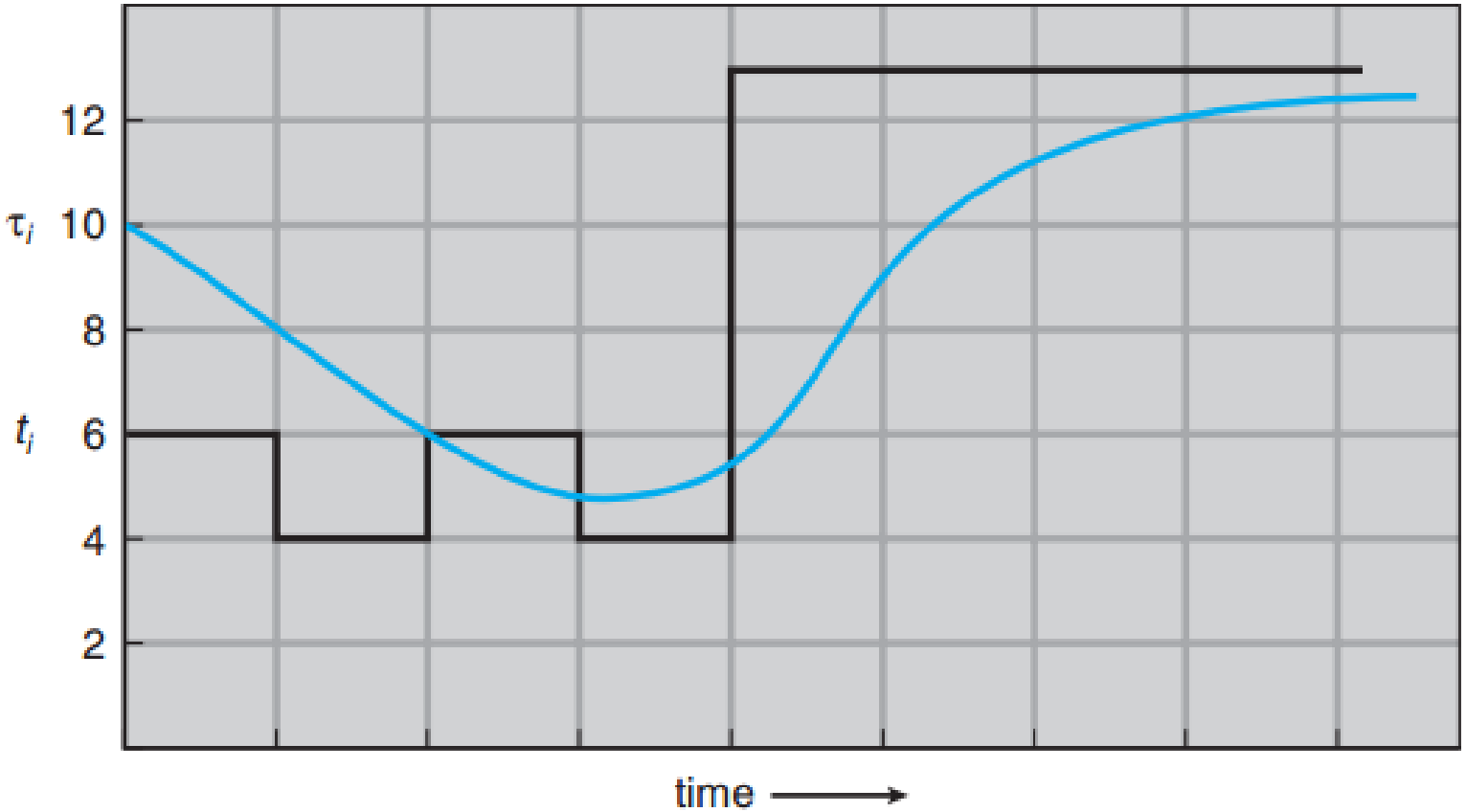
Vẽ Gantt chart và tính thời gian chờ trung bình đối với:

1. FCFS
2. Non-preemptive SJF
3. Preemptive SJF (SRTF)

SJF: DETERMINING LENGTH OF NEXT CPU BURST

- Vấn đề:
  - Không biết chính xác độ dài của phiên sử dụng CPU tiếp theo
  - Chỉ có thể dự đoán dựa theo độ dài của các phiên sử dụng CPU trước

$$\tau_{n+1} = \alpha\, t_n + (1 - \alpha)\tau_n$$



CPU burst (t_i)		6	4	6	4	13	13	13	...
"guess" (τ_i)	10	8	6	6	5	9	11	12	...

Figure 6.3 Prediction of the length of the next CPU burst.

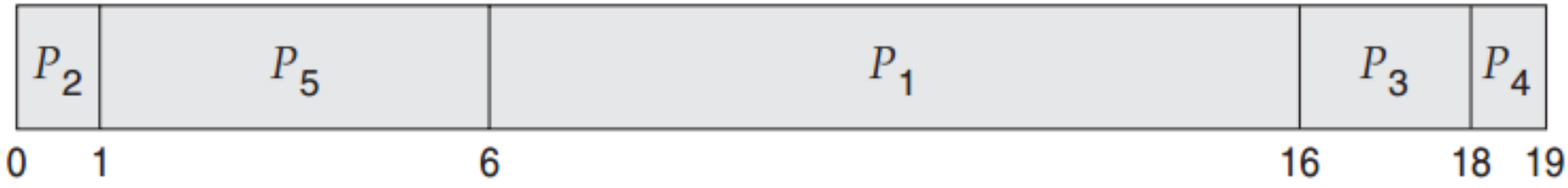
PRIORITY SCHEDULING

- Thuật ngữ: Điều phối có ưu tiên
- Nguyên tắc
  - Mỗi tiến trình được gắn với một tham số gọi là độ ưu tiên *p* (*priority*)
  - CPU được phân phối cho tiến trình có độ ưu tiên cao nhất
  - Nếu 2 tiến trình có cùng độ ưu tiên, FCFS được sử dụng
- *Chú ý*:
  - Độ ưu tiên là một số nguyên nằm trong khoảng nhất định, ví dụ từ 0 đến 7 hoặc từ 0 đến 4095
  - Giả sử, 0 tương ứng với độ ưu tiên cao nhất
  - SJF là một trường hợp đặc biệt của Priority Scheduling ($p = ?$)

PRIORITY SCHEDULING: VÍ DỤ

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	4
P_4	1	5
P_5	5	2

Gantt chart



Thời gian chờ trung bình:

$$\frac{6+0+16+18+1}{5} = \frac{41}{5} = 8.2\ ms$$

CÁCH XÁC ĐỊNH ĐỘ ƯU TIÊN

- Độ ưu tiên trong
  - Được tính dựa trên các một số đại lượng đo được của tiến trình, ví dụ như:
    - Giới hạn thời gian (*time limits*)
    - Yêu cầu về bộ nhớ (*memory requirements*)
    - Số file đang mở (*number of open files*)
    - Tỷ số giữa thời gian vào/ra trung bình và thời gian sử dụng CPU trung bình (*ratio of average I/O burst to average CPU burst*)
- Độ ưu tiên ngoài
  - Được thiết lập dựa trên một số tiêu chí ngoài HĐH như:
    - Tầm quan trọng của tiến trình (*importance of the process*)
    - Chi phí sử dụng máy tính (*amount of funds being paid for computer use*)
    - …

PREEMPTIVE VS. NON-PREEMPTIVE

- Preemptive priority scheduling
  - Nếu tiến trình mới xuất hiện có độ ưu tiên cao hơn tiến trình đang thực hiện (*currently running process*), CPU được chuyển cho tiến trình mới
- Non-preemptive priority scheduling
  - Nếu tiến trình mới xuất hiện có độ ưu tiên cao hơn tiến trình đang thực hiện, tiến trình mới được đưa vào đầu hàng đợi ready

STARVATION PROBLEM

- Known as Indefinite blocking
- Thuật ngữ
  - "Nạn đói"
  - Chờ không xác định
- Vấn đề
  - Tiến trình có độ ưu tiên thấp có thể nằm trong hàng đợi ready trong một khoảng thời gian dài nếu trong hàng đợi luôn có các tiến trình có độ ưu tiên cao hơn
- Ví dụ
  - Một tiến trình nằm trong hàng chờ từ năm 1967 được phát hiện khi tắt IBM 7094 (MIT) vào năm 1973

STARVATION PROBLEM

- Giải pháp
  - Tăng dần độ ưu tiên của tiến trình theo thời gian chờ trong hệ thống
- Ví dụ
  - Giải sử độ ưu tiên nằm trong khoảng từ 0 (high) đến 127 (low)
  - Cứ mỗi 15’ chờ, độ ưu tiên của tiến trình được tăng lên 1
  - Mất khoảng 32 giờ để một tiến trình có độ ưu tiên 127 đạt tới độ ưu tiên 0

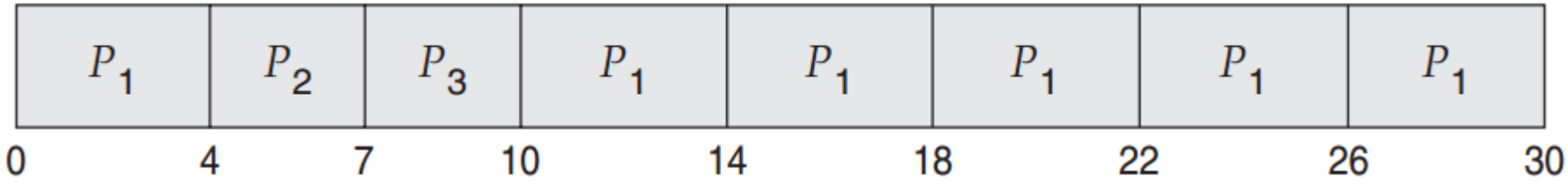
ROUND-ROBIN SCHEDULING

- Thuật ngữ: lập lịch quay vòng
- Được thiết kế cho hệ chia sẻ thời gian (time-sharing systems)
- Hoạt động theo chế độ preemptive
- Nguyên tắc
  - Tham số *lượng tử thời gian* (time quantum/ time slice) *t*
  - Mỗi tiến trình được sử dụng CPU trong một khoảng thời gian nhỏ hơn hoặc bằng *t*
  - Khi hết thời gian, tiến trình được đưa vào cuối hàng đợi ready
  - Nếu có *n* tiến trình, thời gian chờ đợi nhiều nhất là (*n*-1)*t*

ROUND-ROBIN SCHEDULING: VÍ DỤ

Process	Burst Time
P_1	24
P_2	3
P_3	3

Lượng tử thời gian t = 4 ms



Thời gian chờ trung bình:

$$\frac{(10-4)+4+7}{3} = \frac{17}{3} = 5.66\ ms$$

LƯỢNG TỬ THỜI GIAN

- Hiệu quả của thuật toán RR phụ thuộc vào kích thước của lượng tử thời gian
- *t* lớn : RR → FCFS
- *t* nhỏ: chuyển đổi ngữ cảnh (context switch) phải thực hiện nhiều lần
- Giá trị của lượng tử thời gian ảnh hưởng tới thời gian hoàn thành (turnaround time) của tiến trình
- Thông thường, *t* nằm trong khoảng từ 10 đến 100 ms
- *Nguyên tắc thiết lập giá trị cho lượng tử thời gian:*
  - *80 percent of the CPU bursts should be shorter than the time quantum*

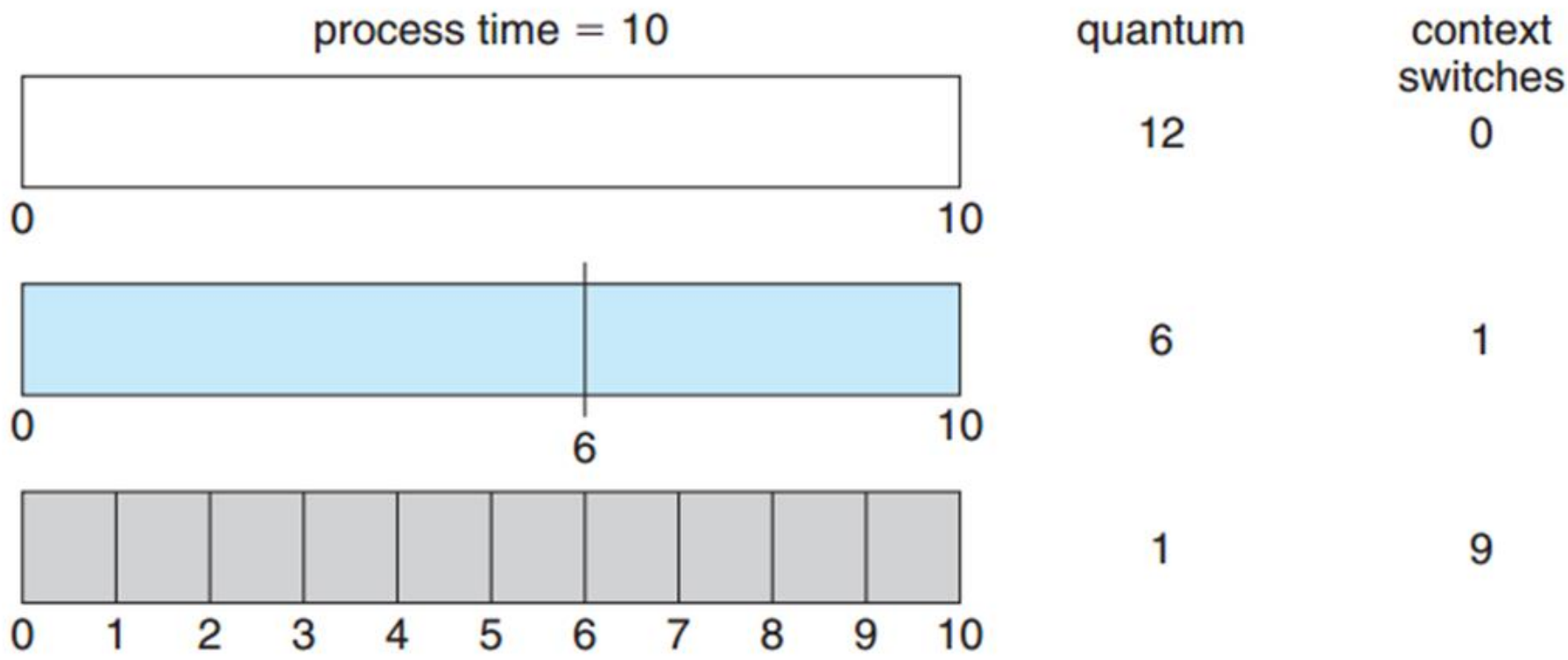
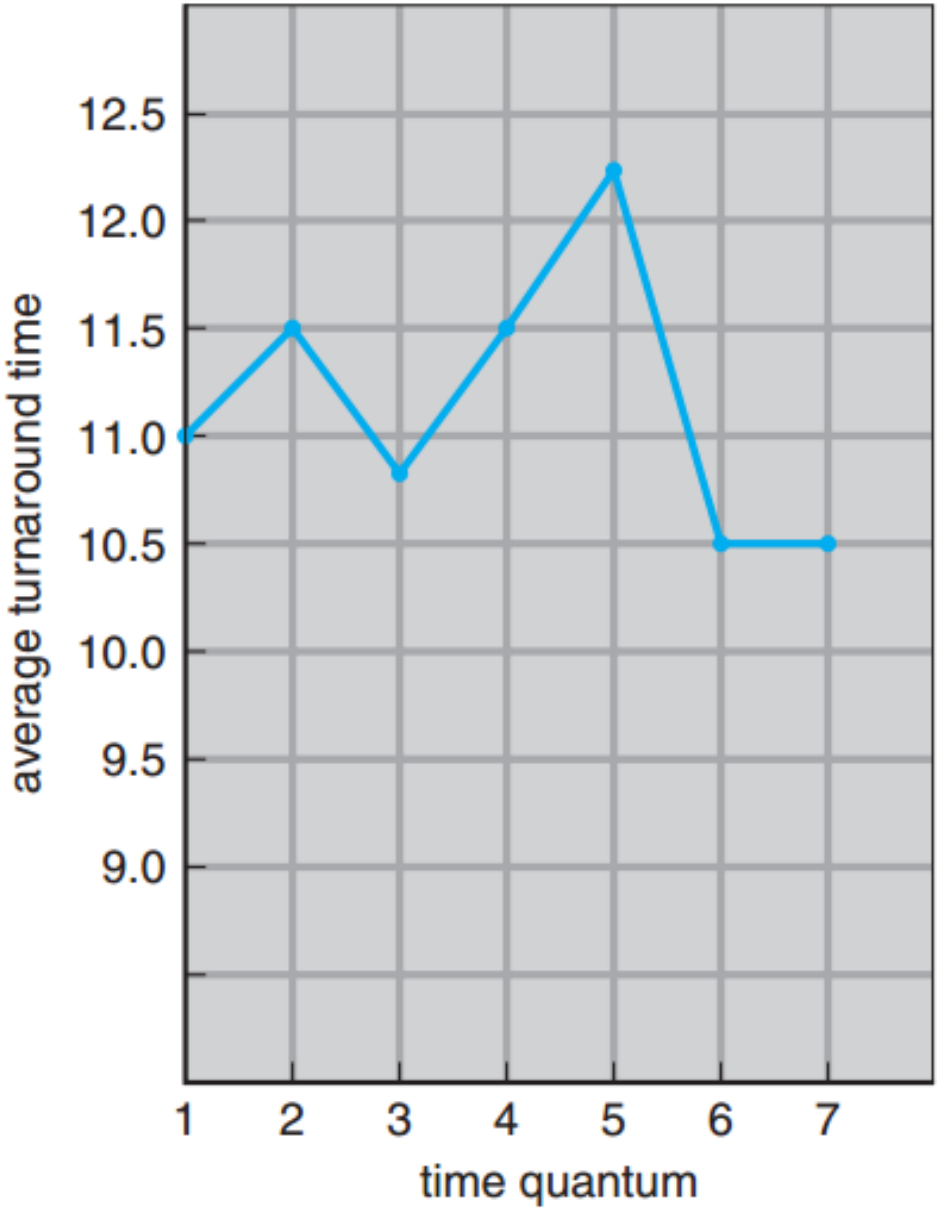


Figure 6.4 How a smaller time quantum increases context switches.



process	time
P_1	6
P_2	3
P_3	1
P_4	7

Figure 6.5 How turnaround time varies with the time quantum.

RR: VÍ DỤ

Process	Burst Time
P_1	53
P_2	17
P_3	68
P_4	24

$t = 20$

P_1	P_2	P_3	P_4	P_1	P_3	P_4	P_1	P_3	P_3

0 20 37 57 77 97 117 121 134 154 162

RR: VÍ DỤ

Process	Arrival Time	Burst Time
P_1	0	8
P_2	1	4
P_3	2	9
P_4	3	5

MULTILEVEL QUEUE

- Thuật ngữ: Điều phối với hàng chờ đa mức
- Hàng đợi ready được chia thành các hàng đợi
  - Foreground (interactive)
  - Background (batch)
- Mỗi hàng đợi sử dụng thuật toán điều phối riêng
  - Foreground – RR
  - Background – FCFS

MULTILEVEL QUEUE

- Điều phối giữa các hàng đợi
  - Fixed priority scheduling
    - Server all from foreground then from background
    - Possibility of starvation
    - Preemptive
  - Time slice
    - Each queue gets a certain amount of CPU time
    - i.e., 80% to foreground, 20% to background

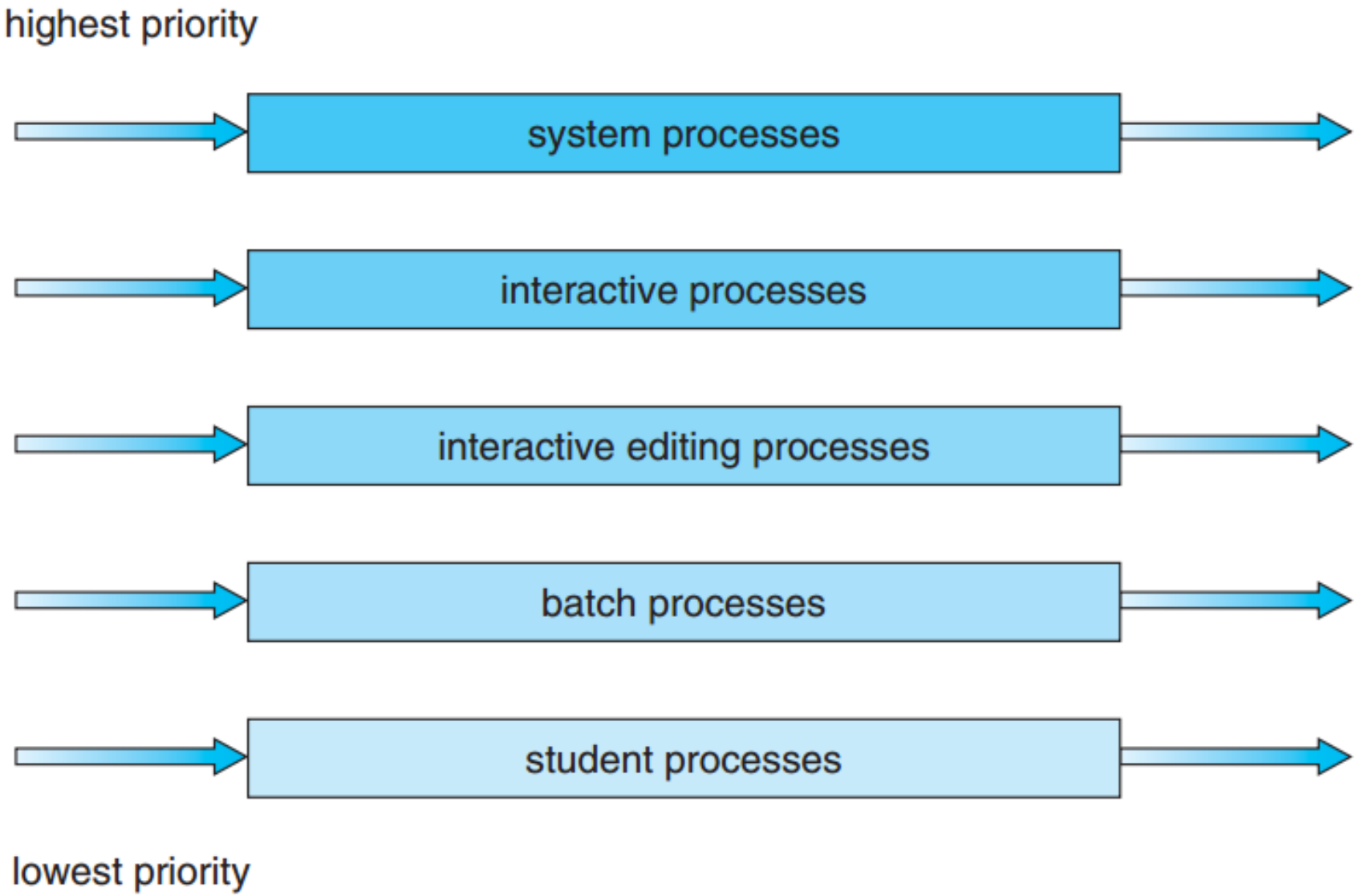


Figure 6.6 Multilevel queue scheduling.

MULTILEVEL FEEDBACK QUEUE SCHEDULING

- Thuật ngữ: Điều phối với hàng chờ đa mức có phản hồi
- Tiến trình có thể di chuyển giữa các hàng đợi
- Tiến trình được phân chia dựa theo đặc điểm sử dụng CPU
  - Tiến trình dùng quá nhiều thời gian CPU được chuyển xuống hàng đợi có độ ưu tiên thấp hơn
  - Tiến trình thiên về vào/ra (I/O-bound) và tiến trình tương tác (interactive process) nằm ở hàng đợi có độ ưu tiên cao
  - Tiến trình đợi quá lâu trong hàng đợi có độ ưu tiên thấp được chuyển lên hàng đợi có độ ưu tiên cao hơn

MULTILEVEL FEEDBACK QUEUE SCHEDULING

- Thuật toán MFQ cần có các tham số sau
  - Số lượng hàng đợi
  - Thuật toán điều phối cho mỗi hàng đợi
  - Phương pháp tăng/ giảm độ ưu tiên cho một tiến trình
  - Phương pháp xác định tiến trình mới sẽ được đưa vào hàng đợi nào

MULTILEVEL FEEDBACK QUEUE

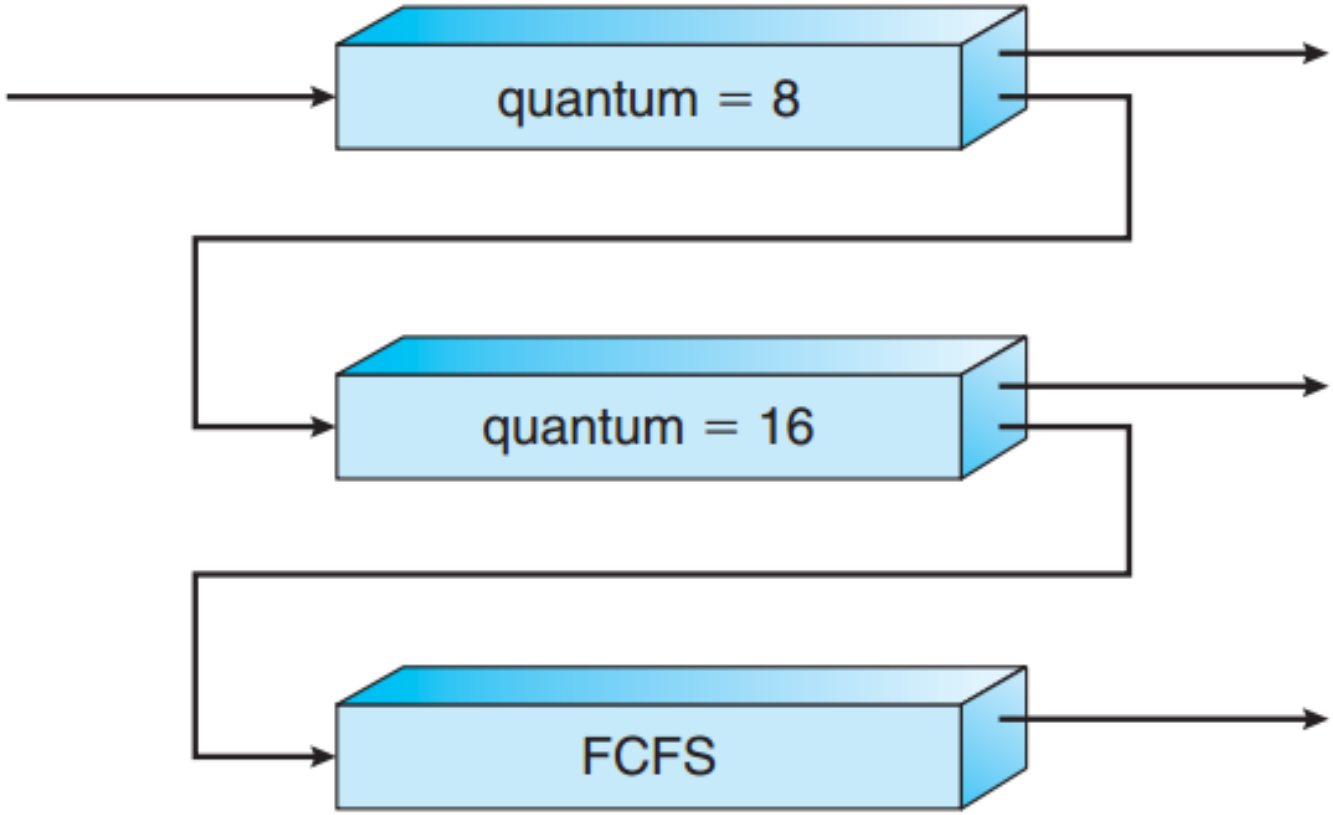


Figure 6.7 Multilevel feedback queues.

TỰ HỌC

- Luồng và điều phối luồng (Thread & Thread Scheduling) – chapter 4 & 6.4
- Điều phối đa xử lý (Multiple-Processor Scheduling) – 6.5
- Điều phối thời gian thực (Real-Time CPU Scheduling) – 6.6
- Đánh giá thuật toán điều phối (Algorithm Evaluation) – 6.8