



COMPUTER ENGINEERING



**UIT**  
TRƯỜNG ĐẠI HỌC  
CÔNG NGHỆ THÔNG TIN

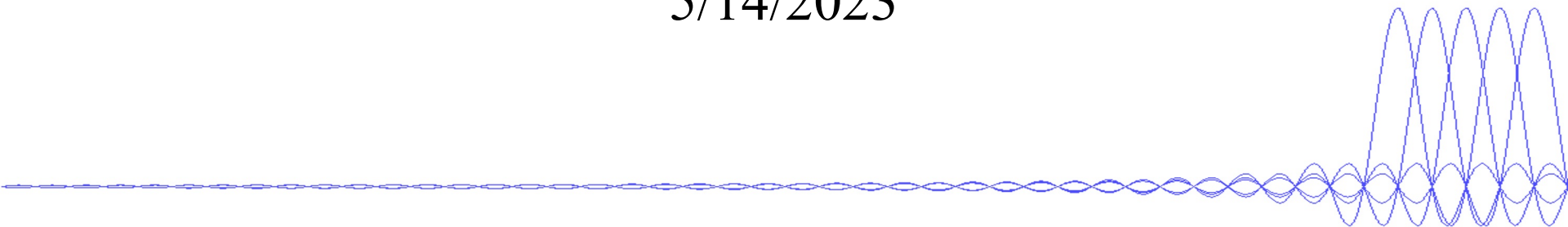
# HỆ ĐIỀU HÀNH

## Chương 4

# ĐỊNH THỜI CPU

### (Phần hai)

5/14/2023





# Câu hỏi ôn tập Chương 4 – Phần một

- Các khái niệm cơ bản về định thời
- Các bộ định thời
- Các tiêu chuẩn định thời CPU
- Các giải thuật định thời
  - First-Come, First-Served (FCFS)
  - Shortest Job First (SJF)
  - Shortest Remaining Time First (SRTF)
  - Priority Scheduling



# Nội dung Chương 4 – Phần hai

- Các giải thuật định thời
  - First-Come, First-Served (FCFS)
  - Shortest Job First (SJF)
  - Shortest Remaining Time First (SRTF)
  - Priority Scheduling
  - **Round-Robin (RR)**
  - **Highest Response Ratio Next (HRRN)**
  - **Multilevel Queue**
  - **Multilevel Feedback Queue**



# Giải thuật Round Robin (RR)

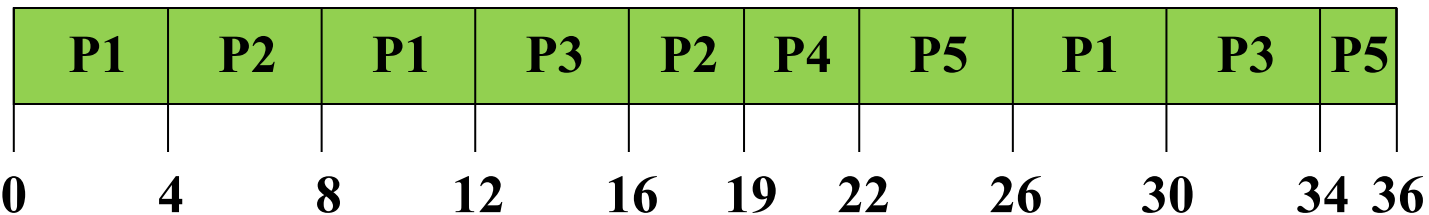
- Mỗi tiến trình nhận được một đơn vị thời gian CPU (time slice, quantum time) để thực thi. Thông thường khoảng thời gian này nhỏ, từ 10-100 ms.
- Sau khoảng thời gian đó, tiến trình bị đoạt quyền và trở về cuối *ready queue*.
- Gọi  $n$  là số lượng tiến trình trong *ready queue* và  $q$  là khoảng thời gian đơn vị mà CPU được cấp phát cho tiến trình (quantum time), khi đó, không có tiến trình nào phải chờ đợi quá  $(n - 1)q$  đơn vị thời gian.



# Ví dụ định thời sử dụng giải thuật RR

Process	Arrival Time	Burst Time
P1	0	12
P2	2	7
P3	5	8
P4	9	3
P5	12	6

## ■ Giải đồ Gantt (quantum time = 4)



## ■ Thời gian đáp ứng:

- $P1 = 0, P2 = 2, P3 = 7, P4 = 10, P5 = 10$
- Thời gian đáp ứng trung bình: 5.8



# Ví dụ định thời sử dụng giải thuật RR

## ■ Thời gian chờ:

- $P1 = 4 + 14, P2 = 2 + 8, P3 = 7 + 14, P4 = 10, P5 = 10 + 8$
- Thời gian chờ trung bình: 15.4

## ■ Thời gian hoàn thành:

- $P1 = 40, P2 = 17, P3 = 29, P4 = 13, P5 = 24$
- Thời gian hoàn thành trung bình: 22.6



# Nhận xét về giải thuật Round Robin

- Nếu  $q$  lớn: RR  $\Rightarrow$  FCFS
- Nếu  $q$  nhỏ: phải tốn chi phí chuyển ngữ cảnh giữa các tiến trình  $\Rightarrow q$  không nên quá nhỏ
- Ưu tiên tiến trình hướng CPU (CPU-bound process)
- RR sử dụng một giả thiết ngầm là tất cả các tiến trình đều có tầm quan trọng ngang nhau.
- Ưu điểm: Thời gian đáp ứng nhỏ.
- Hạn chế: Thời gian chờ đợi trung bình và thời gian hoàn thành trung bình của giải thuật RR thường khá lớn.

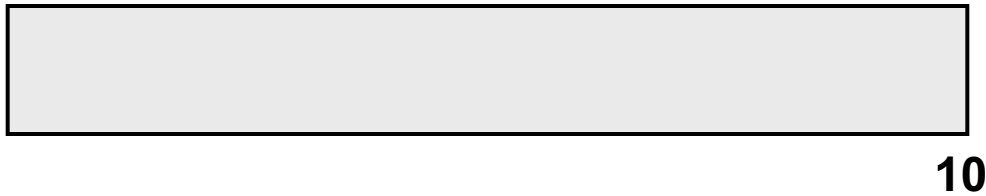


# Quantum time và chuyển ngữ cảnh trong RR

Process time = 10

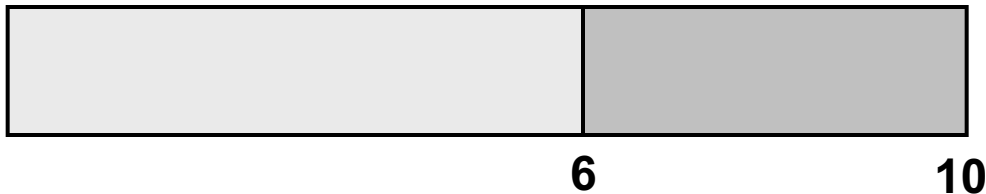
quantum

context  
switch



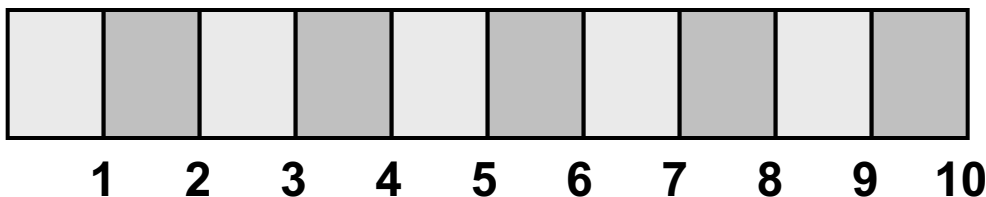
12

0



6

1



1

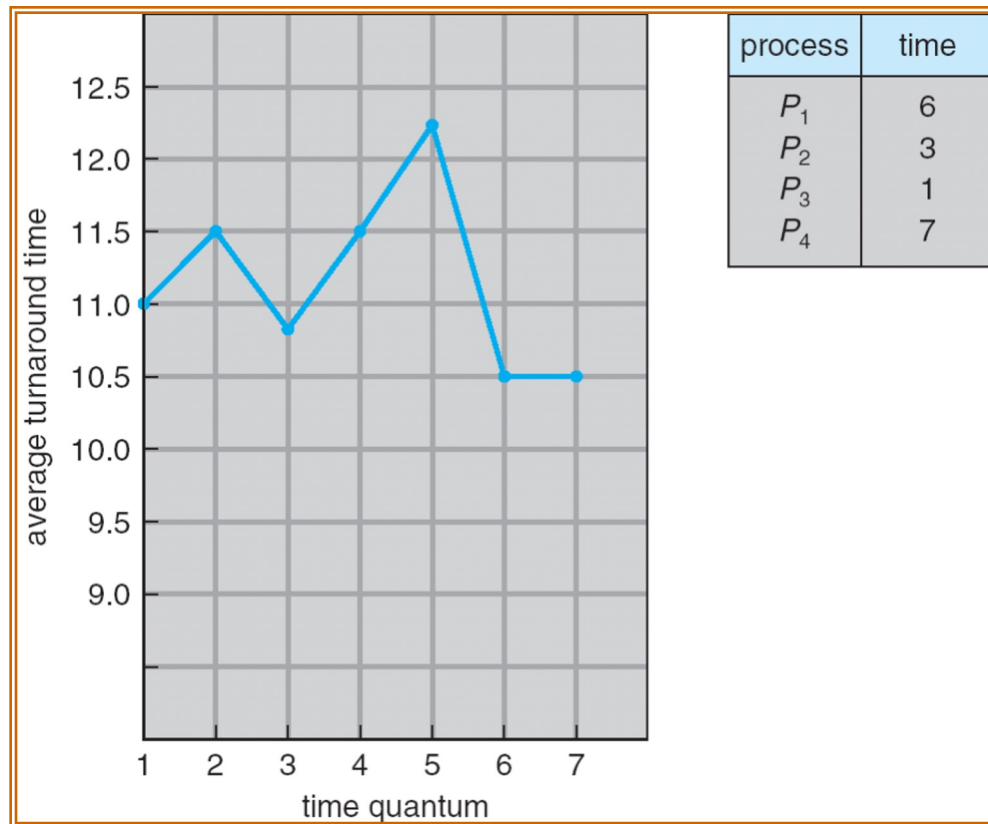
9





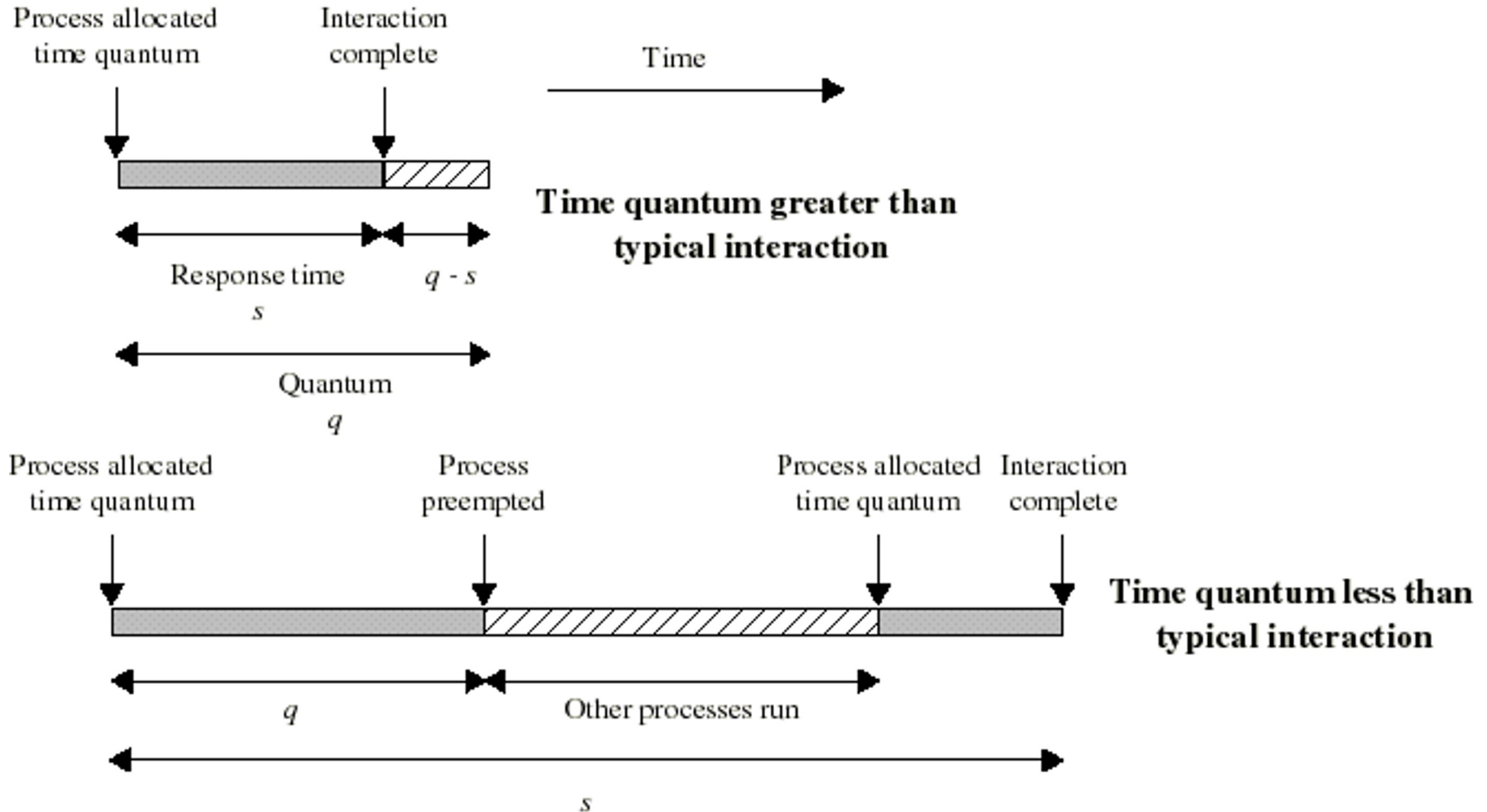
# Quantum time và Thời gian hoàn thành

- Thời gian hoàn thành trung bình không chắc sẽ được cải thiện khi quantum lớn





# Quantum time và Thời gian đáp ứng





# Quantum time và hiệu suất hệ thống

- Khi thực hiện chuyển ngữ cảnh, *kernel thread* sẽ sử dụng CPU, không phải *user thread*.
- **Phí tổn hệ thống (OS Overhead):** thời gian HĐH sử dụng CPU để thực hiện chuyển ngữ cảnh.
- Hiệu suất hệ thống: tùy thuộc vào kích thước của *quantum time*
  - Nếu quantum time ngắn: thời gian đáp ứng nhanh, nhưng phí tổn hệ thống lớn do số lần chuyển ngữ cảnh tăng
  - Nếu quantum time dài: hiệu quả sử dụng CPU tốt hơn, nhưng thời gian đáp ứng cũng lớn.
  - Nếu quantum time quá lớn, RR trở thành FCFS.



# Cách chọn quantum time

- Quantum time và thời gian cho chuyển ngữ cảnh:
  - Nếu quantum time là 20 ms và thời gian chuyển ngữ cảnh là 5 ms, như vậy OS overhead chiếm  $5/25 = 20\%$ .
  - Nếu quantum là 500 ms, thì phí tổn chỉ còn 1%. Nhưng nếu có nhiều người sử dụng trên hệ thống và thuộc loại interactive thì sẽ thấy đáp ứng rất chậm.
  - Tùy thuộc vào tập công việc mà lựa chọn quantum time.
  - Quantum time nên lớn trong tương quan so sánh với thời gian cho chuyển ngữ cảnh
    - Ví dụ với 4.3 BSD UNIX, quantum time là 1s



# Giải thuật Highest Response Ratio Next

- Chọn tiến trình kế tiếp có giá trị RR (Response ratio) lớn nhất
- Các process ngắn được ưu tiên hơn (vì *service time* nhỏ)

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

- Câu hỏi thảo luận:
  - So sánh với cơ chế Aging?
  - Ưu điểm và hạn chế của hướng tiếp cận này?



# Định thời sử dụng hàng đợi đa mức (Multilevel Queue Scheduling)

- *Ready queue* được chia thành nhiều hàng đợi riêng biệt theo một số tiêu chuẩn sau:
  - Đặc điểm và yêu cầu định thời của tiến trình
  - Phân loại tiến trình: Foreground (interactive) và background,...
- Tiến trình được gán cố định vào một hàng đợi, mỗi hàng đợi sử dụng giải thuật định thời riêng



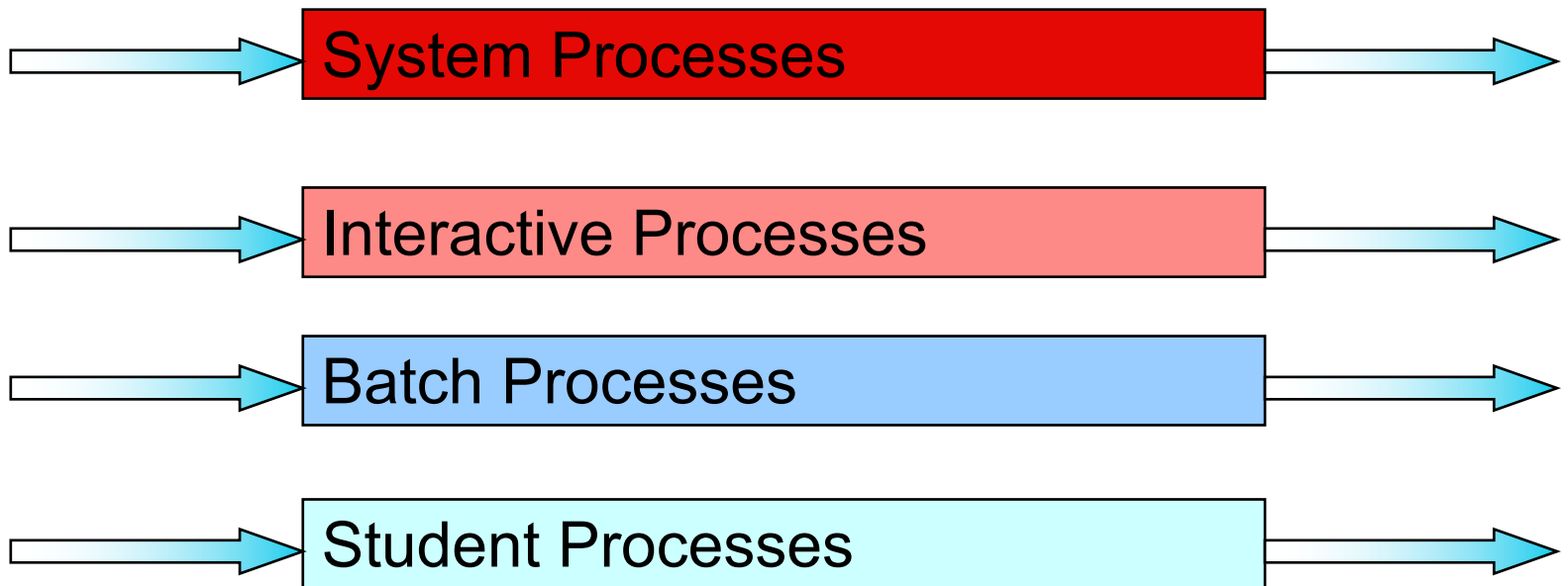
# Việc định thời giữa các hàng đợi

- Hệ điều hành cần phải định thời cho các hàng đợi:
  - Fixed priority scheduling: phục vụ từ hàng đợi có độ ưu tiên cao đến thấp □ có thể phát sinh vấn đề đói tài nguyên (starvation).
  - Time slice: mỗi hàng đợi được nhận một khoảng thời gian chiếm CPU và phân phối cho các tiến trình trong hàng đợi khoảng thời gian đó. Ví dụ: 80% cho hàng đợi foreground định thời bằng RR và 20% cho hàng đợi background định thời bằng giải thuật FCFS



# Ví dụ phân chia hạng đợi và tiến trình

Độ ưu tiên cao nhất



Độ ưu tiên thấp nhất





# Hệ chế của Multilevel Queue

- ❑ Tiến trình không thể chuyển từ hàng đợi này sang hàng đợi khác
  - ❑ Có thể làm giảm hiệu suất hệ thống trong trường hợp một hạng đợi có nhiều tiến trình trong khi các hạng đợi khác lại trống
- ⇒ Khắc phục bằng cơ chế feedback: cho phép tiến trình di chuyển một cách thích hợp giữa các hàng đợi khác nhau.
- ⇒ Giải thuật: **Multilevel Feedback Queue**



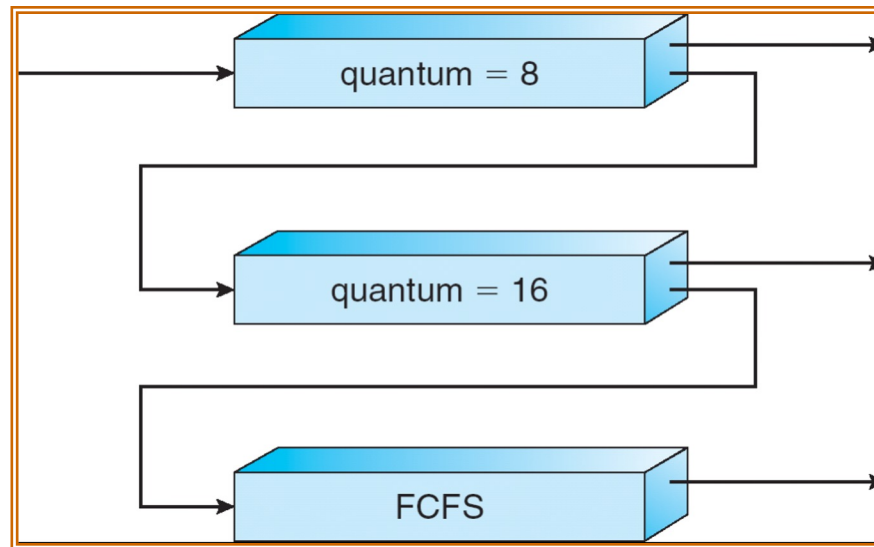
# Định thời với Multilevel Feedback Queue

- Phân loại tiến trình dựa trên các đặc tính về CPU-burst.
- Sử dụng chế độ trung dụng (preemptive).
- Sau một khoảng thời gian nào đó, các I/O-bound process và interactive process sẽ ở các hàng đợi có độ ưu tiên cao hơn còn CPU-bound process sẽ ở các hàng đợi có độ ưu tiên thấp hơn.
- Một tiến trình đã chờ quá lâu ở một hàng đợi có độ ưu tiên thấp có thể được chuyển đến hàng đợi có độ ưu tiên cao hơn (cơ chế aging)



# Ví dụ về Multilevel Feedback Queue

- Ví dụ: Có 3 hàng đợi
  - Q0, dùng RR với quantum 8 ms
  - Q1, dùng RR với quantum 16 ms
  - Q2, dùng FCFS





# Các vấn đề với Multilevel Feedback Queue

- Định thời dùng multilevel feedback queue đòi hỏi phải giải quyết các vấn đề sau
  - Số lượng hàng đợi bao nhiêu là thích hợp?
  - Dùng giải thuật định thời nào ở mỗi hàng đợi?
  - Làm sao để xác định thời điểm cần chuyển một process đến hàng đợi cao hơn hoặc thấp hơn?
  - Khi tiến trình yêu cầu được xử lý thì đưa vào hàng đợi nào là hợp lý nhất?



# So sánh các giải thuật

- Giải thuật định thời nào là tốt nhất?
- Câu trả lời phụ thuộc các yếu tố sau:
  - Tần suất tải việc (System workload)
  - Sự hỗ trợ của phần cứng đối với dispatcher
  - Sự tương quan về trọng số của các tiêu chuẩn định thời như response time, hiệu suất CPU, throughput,...
  - Phương pháp định lượng so sánh



# Tóm tắt lại nội dung buổi học

## ■ Các giải thuật định thời

- ❑ First-Come, First-Served (FCFS)
- ❑ Shortest Job First (SJF)
- ❑ Shortest Remaining Time First (SRTF)
- ❑ Priority Scheduling
- ❑ Round-Robin (RR)
- ❑ Highest Response Ratio Next (HRRN)
- ❑ Multilevel Queue
- ❑ Multilevel Feedback Queue



## Câu hỏi ôn tập chương 4

- Tại sao phải định thời? Nêu các bộ định thời và mô tả về chúng?
- Các tiêu chuẩn định thời CPU?
- Có bao nhiêu giải thuật định thời? Kể tên?
- Mô tả và nêu ưu điểm, nhược điểm của từng giải thuật định thời? FCFS, SJF, SRTF, RR, Priority Scheduling, HRRN, MQ, MFQ.



# Bài tập 1

Sử dụng các giải thuật FCFS, SJF, SRTF, Priority -Pre, RR (10) để tính các giá trị thời gian đợi, thời gian đáp ứng và thời gian hoàn thành trung bình và vẽ giản đồ Gantt:

Process	Arrival	Burst	Priority
P1	0	20	20
P2	25	25	30
P3	20	25	15
P4	35	15	35
P5	10	35	5
P6	15	50	10

Với RR, điều gì sẽ xảy ra khi P5 vào tại thời điểm P1 vừa hết quantum time?





## Bài tập 2

Cho 5 tiến trình với thời gian vào và thời gian cần CPU tương ứng như bảng sau:

Process	Arrival	Burst
P1	0	10
P2	2	29
P3	4	3
P4	5	7
P5	7	12

Vẽ giản đồ Gantt và tính thời gian đợi trung bình, thời gian đáp ứng trung bình và thời gian lưu lại trong hệ thống (turnaround time) trung bình cho các giải thuật?

- FCFS
- SJF preemptive
- RR với quantum time = 10



## Bài tập 3

Xét tập các tiến trình sau (với thời gian yêu cầu CPU và độ ưu tiên kèm theo) :

Process	Arrival	Burst	Priority
P1	0	10	3
P2	1	3	2
P3	2	2	1
P4	3	1	2
P5	4	5	4

Vẽ giản đồ Gantt và tính thời gian đợi trung bình và thời gian lưu lại trong hệ thống trung bình (turnaround time) cho các giải thuật?

- SJF Preemptive
- RR với quantum time = 2
- Điều phối theo độ ưu tiên độc quyền (độ ưu tiên  $1 > 2 > \dots$ )



## Bài tập 4

Vẽ giản đồ Gantt và tính thời gian đợi trung bình và thời gian lưu lại trong hệ thống (turnaround time) trung bình cho các giải thuật:

Process	Burst Time	Arrival Time
P1	10	5
P2	29	2
P3	3	0
P4	7	1
P5	12	7

- FCFS, SJF
- RR với quantum time = 10



## Bài tập 5

Cho 4 tiến trình và thời gian vào (Arrival Time) tương ứng:

Process	Arrival Time	CPU Burst Time
P1	0	12
P2	2	7
P3	3	5
P4	5	9

Vẽ sơ đồ Gantt và tính thời gian chờ trung bình (average wait time) và thời gian xoay vòng (average turnaround time) trung bình cho các giải thuật định thời

- Shortest Remaining Time First (SRTF)
- Round Robin (RR) với quantum = 4



## Bài tập 6

Cho 5 tiến trình P1, P2, P3, P4, P5 với thời gian vào Ready List và thời gian cần CPU tương ứng như bảng sau:

Process	Arrival Time	CPU Burst Time
P1	0	8
P2	2	19
P3	4	3
P4	5	6
P5	7	12

Vẽ sơ đồ Gantt và tính thời gian chờ trung bình, thời gian đáp ứng trung bình và thời gian lưu lại trong hệ thống (turnaround time) trung bình cho các giải thuật:

- FCFS
- SJF preemptive
- RR với quantum time = 6



COMPUTER ENGINEERING



**UIT**  
TRƯỜNG ĐẠI HỌC  
CÔNG NGHỆ THÔNG TIN

# THẢO LUẬN

