



COMPUTER ENGINEERING



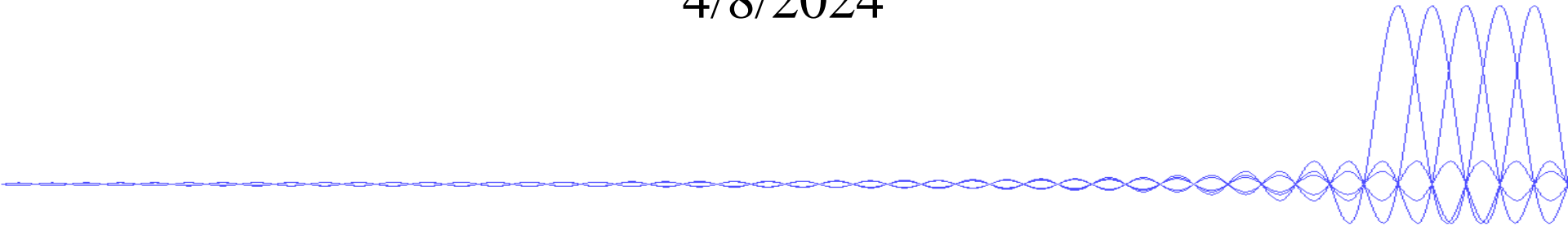
**UIT**  
TRƯỜNG ĐẠI HỌC  
CÔNG NGHỆ THÔNG TIN

# HỆ ĐIỀU HÀNH

## Chương 4 (2)

### Định thời CPU

4/8/2024





# Câu hỏi ôn tập chương 4 (1)

- 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 (2)

## ■ 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



# Round Robin (RR)

- Mỗi process nhận được một đơn vị nhỏ thời gian CPU (time slice, quantum time), thông thường từ 10-100 msec để thực thi
- Sau khoảng thời gian đó, process bị đoạt quyền và trở về cuối hàng đợi ready
- Nếu có  $n$  process trong hàng đợi ready và quantum time =  $q$  thì không có process nào phải chờ đợi quá  $(n - 1)q$  đơn vị thời gian



# Round Robin (RR) (tt)

## ■ Hiệu suất:

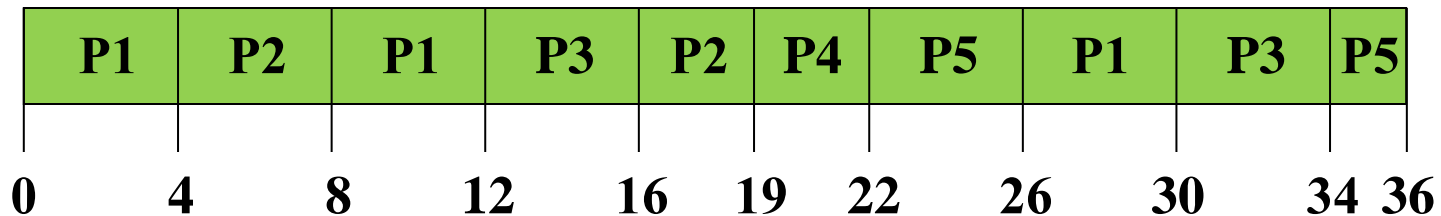
- Nếu  $q$  lớn: RR  $\Rightarrow$  FCFS
- Nếu  $q$  nhỏ:  $q$  không được quá nhỏ bởi vì phải tốn chi phí chuyển ngữ cảnh
- Thời gian chờ đợi trung bình của giải thuật RR thường khá lớn nhưng thời gian đáp ứng nhỏ



# Round Robin (RR) (tt)

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



# Round Robin (RR) (tt)

## ■ 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

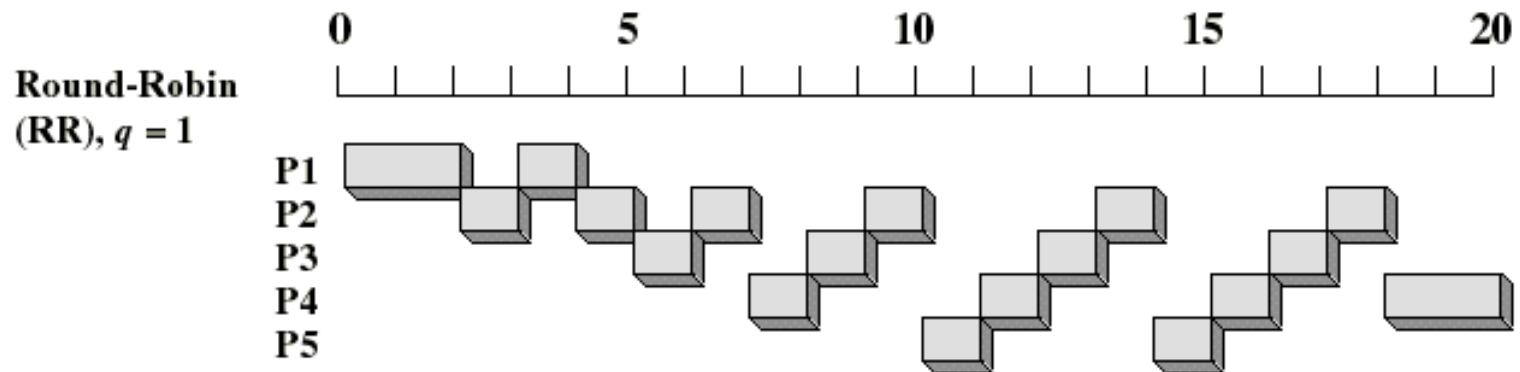
## ■ Nhận xét:

□ Thời gian hoàn thành trung bình lớn hơn SJF, nhưng đáp ứng tốt hơn.



# Round Robin (RR) (tt)

## ■ Quantum time = 1:



□ Thời gian turn-around trung bình cao hơn so với SJF nhưng có thời gian đáp ứng trung bình tốt hơn

□ Ưu tiên CPU-bound process

■ I/O-bound

■ CPU-bound





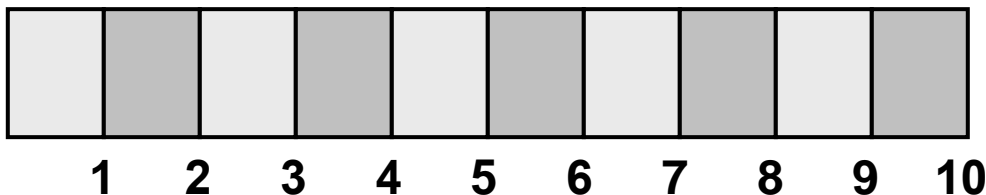
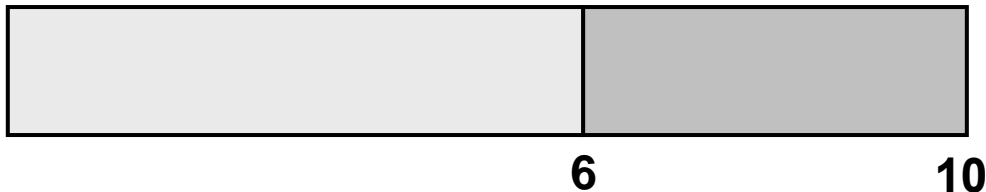
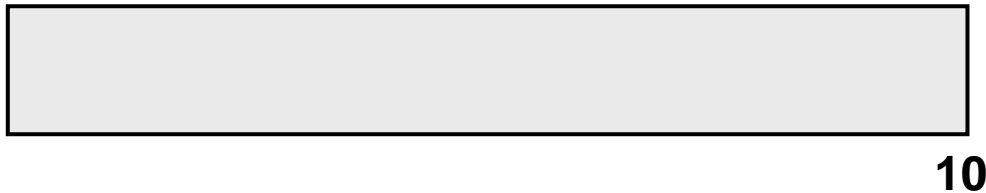
# Round Robin (RR) (tt)

## ■ Quantum time và context switch:

Process time = 10

quantum

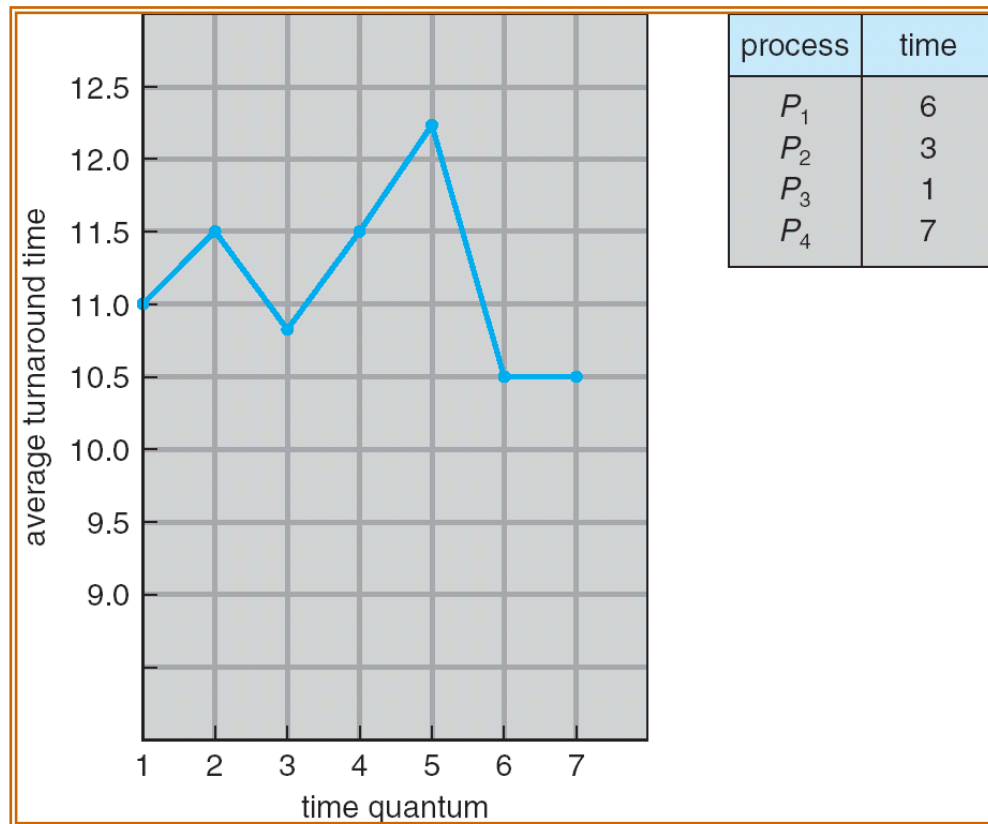
context  
switch





# Round Robin (RR) (tt)

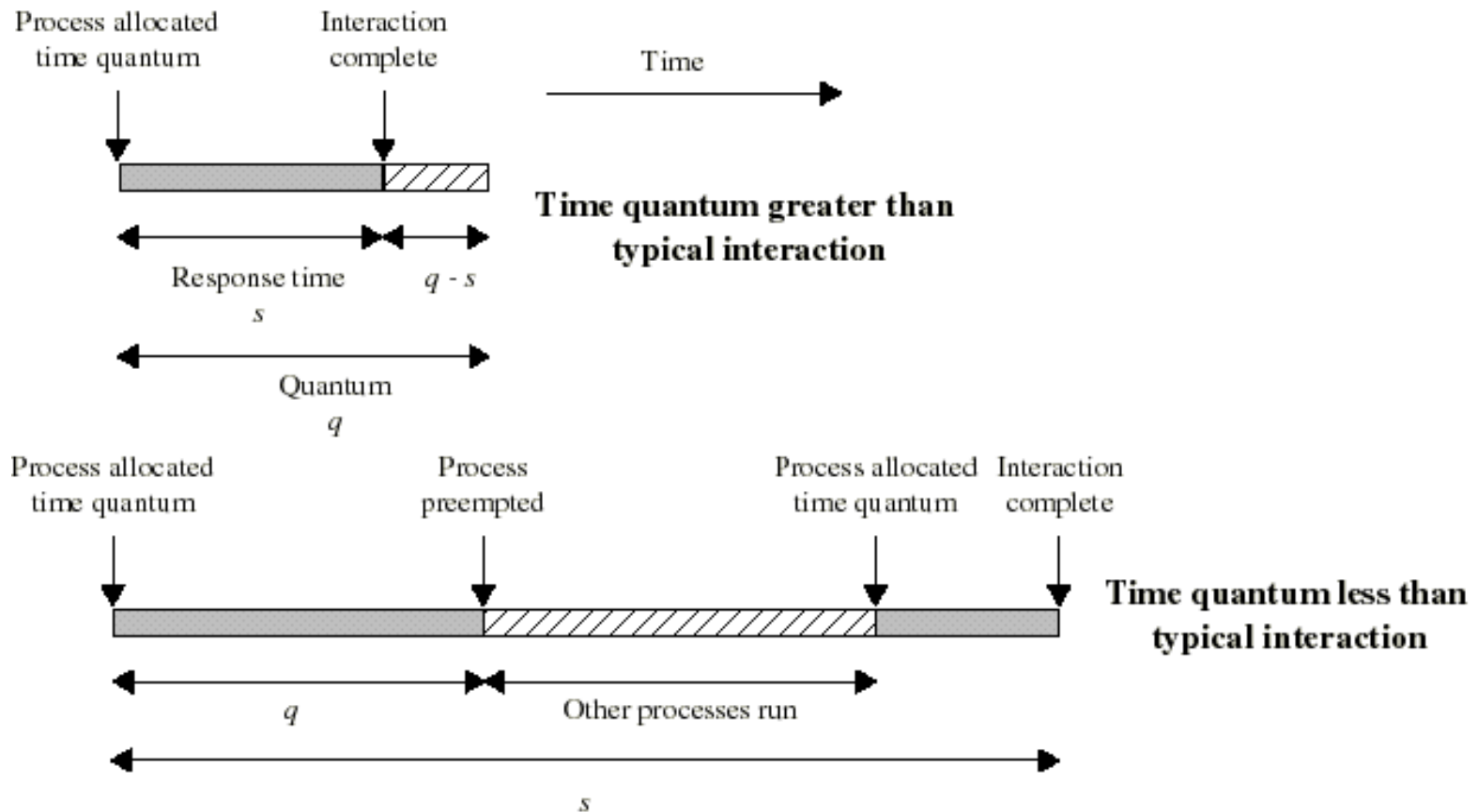
- Thời gian hoàn thành trung bình (average turnaround time) không chắc sẽ được cải thiện khi quantum lớn





# Round Robin (RR) (tt)

## ■ Quantum time và response time





# Quantum time cho Round Robin

- Khi thực hiện process switch thì OS sẽ sử dụng CPU chứ không phải process của người dùng (OS overhead)
  - Dừng thực thi, lưu tất cả thông tin, nạp thông tin của process sắp thực thi
- Performance tùy thuộc vào kích thước của quantum time (còn gọi là time slice), và hàm phụ thuộc này không đơn giản
- Time slice ngắn thì đáp ứng nhanh
  - Vấn đề: có nhiều chuyển ngữ cảnh. Phí tổn sẽ cao.
- Time slice dài hơn thì throughput tốt hơn (do giảm phí tổn OS overhead) nhưng thời gian đáp ứng lớn
  - Nếu time slice quá lớn, RR trở thành FCFS



# Quantum time cho Round Robin (tt)

## ■ Quantum time và thời gian cho process switch:

- Nếu quantum time = 20 ms và thời gian cho process switch = 5 ms, như vậy phí tổn OS overhead chiếm  $5/25 = 20\%$
- Nếu quantum = 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
- Time slice nên lớn trong tương quan so sánh với thời gian cho process switch
  - Ví dụ với 4.3 BSD UNIX, time slice là 1 giây



# Quantum time cho Round Robin (tt)

- Nếu có  $n$  process trong hàng đợi ready, và quantum time là  $q$ , như vậy mỗi process sẽ lấy  $1/n$  thời gian CPU theo từng khối có kích thước lớn nhất là  $q$ 
  - Sẽ không có process nào chờ lâu hơn  $(n - 1)q$  đơn vị thời gian
- RR sử dụng một giả thiết ngầm là tất cả các process đều có tầm quan trọng ngang nhau
  - Không thể sử dụng RR nếu muốn các process khác nhau có độ ưu tiên khác nhau



# Nhược điểm của Round Robin

■ Các process dạng CPU-bound vẫn còn được “ưu tiên”

□ Ví dụ:

- Một I/O-bound process sử dụng CPU trong thời gian ngắn hơn quantum time và bị blocked để đợi I/O.
- Một CPU-bound process chạy hết time slice và lại quay trở về hàng đợi ready queue (ở phía trước các process đã bị blocked)



# Highest Response Ratio Next

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

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





# Multilevel Queue Scheduling

- Hàng đợi ready được chia thành nhiều hàng đợi riêng biệt theo một số tiêu chuẩn như
  - Đặc điểm và yêu cầu định thời của process
  - Foreground (interactive) và background process,...
- Process được gán cố định vào một hàng đợi, mỗi hàng đợi sử dụng giải thuật định thời riêng



# Multilevel Queue Scheduling (tt)

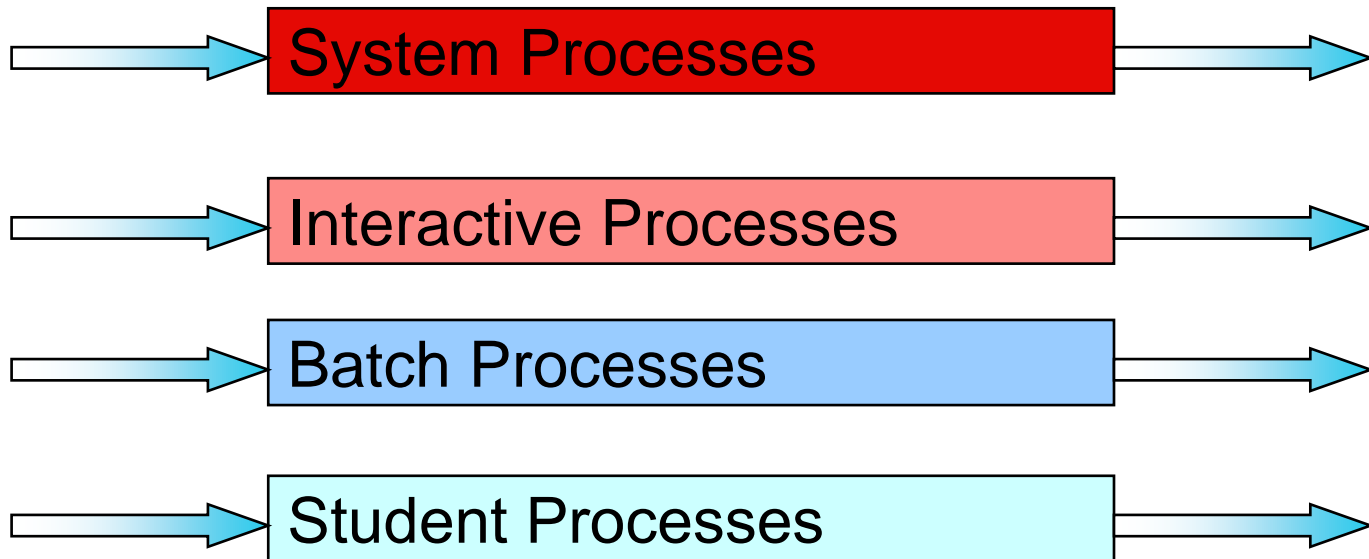
- Hệ điều hành cần phải định thời cho các hàng đợi.
  - Fixed priority scheduling: phục vụ từ hàng đợi có độ ưu tiên cao đến thấp. Vấn đề: có thể có starvation.
  - Time slice: mỗi hàng đợi được nhận một khoảng thời gian chiếm CPU và phân phối cho các process trong hàng đợi khoảng thời gian đó. Ví dụ: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



# Multilevel Queue Scheduling (tt)

## ■ Ví dụ phân nhóm các quá trình

Độ ưu tiên cao nhất



Độ ưu tiên thấp nhất



# Multilevel Feedback Queue

## ■ Vấn đề của multilevel queue

□ Process không thể chuyển từ hàng đợi này sang hàng đợi khác

=> Khắc phục bằng cơ chế feedback: cho phép process di chuyển một cách thích hợp giữa các hàng đợi khác nhau.



# Multilevel Feedback Queue (tt)

## ■ Multilevel Feedback Queue

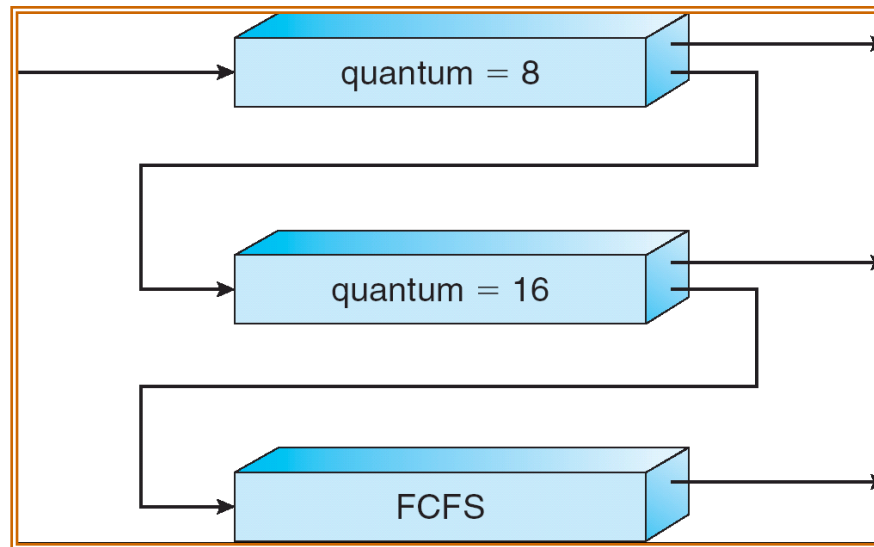
- Phân loại processes dựa trên các đặc tính về CPU-burst
- Sử dụng decision mode preemptive
- Sau một khoảng thời gian nào đó, các I/O-bound process và interactive process sẽ ở các hàng đợi có độ ưu tiên cao hơn còn CPU-bound process sẽ ở các queue có độ ưu tiên thấp hơn
- Một process đã chờ quá lâu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



# Multilevel Feedback Queue (tt)

## ■ Ví dụ: Có 3 hàng đợi

- Q0, dùng RR với quantum 8 ms
- Q1, dùng RR với quantum 16 ms
- Q2, dùng FCFS





# Multilevel Feedback Queue (tt)

- 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 process 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

