

Ôn tập HĐH ngày  
30/11/2022

SUS Imposter Detected

# Các thuật toán điều độ tiến trình

## Các thuật toán điều độ:

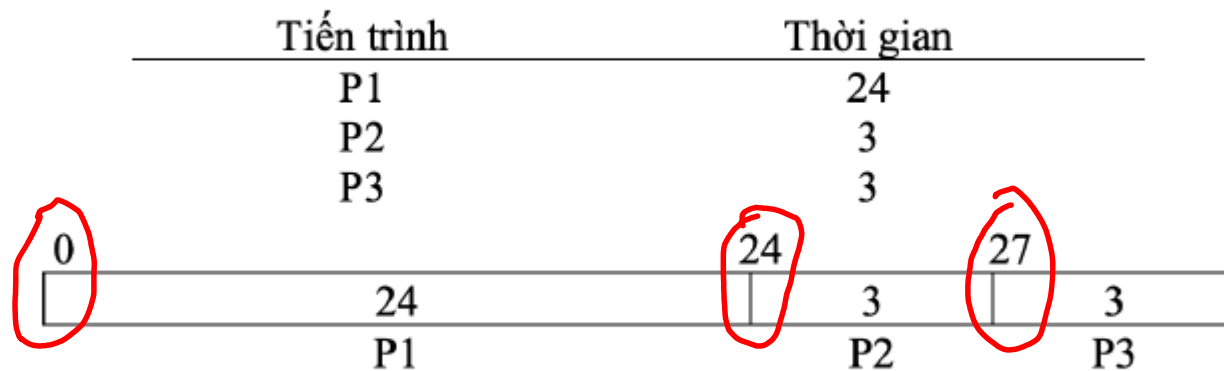
1. Thuật toán đến trước phục vụ trước (FCFS: First Come, First Served)
2. Điều độ quay vòng (RR: round robin)
3. Điều độ ưu tiên tiến trình ngắn nhất (SPF: Shortest Job First)
4. Điều độ ưu tiên thời gian còn lại ngắn nhất (SRTF: Shortest Remaining Time First)
5. Điều độ có mức ưu tiên (Priority Scheduling)

# FCFS: First Come, First Served

## Nguyên tắc:

- ▶ Tiến trình được quyền sử dụng CPU theo trình tự xuất hiện
- ▶ Tiến trình sở hữu CPU tới khi kết thúc hoặc chờ đợi vào ra

Ví dụ: Cho 3 tiến trình với thứ tự xuất hiện và độ dài chu kỳ CPU như sau:



Thời gian chờ đợi của P1, P2, P3 lần lượt là: 0, 24, 27. ✓ ✓ ✓

Thời gian chờ đợi trung bình  $(0 + 24 + 27)/3 = 17$  (s)

# FCFS: First Come, First Served

Nguyên tắc:

- ▶ Tiến trình được quyền sử dụng CPU theo trình tự xuất hiện

Process	Burst Time
P1	5
P2	3
P3	1
P4	7
P5	4



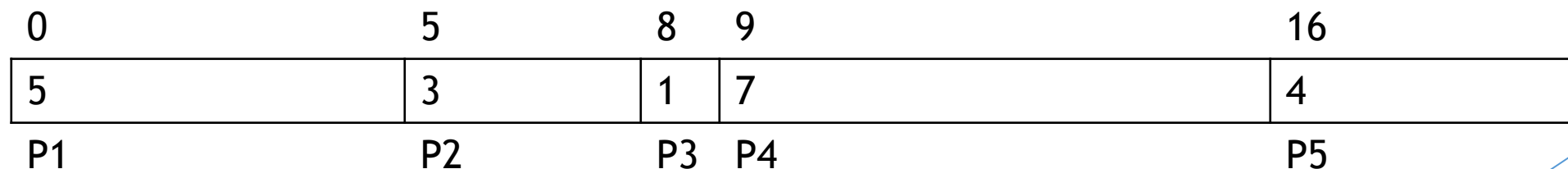
5:00

# FCFS: First Come, First Served

Nguyên tắc:

- ▶ Tiến trình được quyền sử dụng CPU theo trình tự xuất hiện

Process	Burst Time
P1	5
P2	3
P3	1
P4	7
P5	4



$$t = \frac{0 + 5 + 8 + 9 + 16}{5} = 7.6$$

# RR: round robin

## Nguyên tắc:

- ▶ Mỗi tiến trình được cấp một lượng tử thời gian  $t$  để sử dụng
- ▶ Khi hết thời gian, tiến trình bị trưng dụng VXL và được đưa vào cuối hàng đợi sẵn sàng
- ▶ Nếu có  $n$  tiến trình, thời gian chờ đợi nhiều nhất là  $(n-1)t$

Ví dụ:

Tiến trình		Độ dài chu kỳ CPU	
P1		10	
P2		4	
P3		2	

	2	4	6	8	10	12	14
2	2	2	2	2	2	2	2
P1	P2	P3	P1	P2	P1	P1	P1

# RR: round robin

5:00

## Nguyên tắc:

- Mỗi tiến trình được cấp một lượng tử thời gian  $t$  để sử dụng

Process	Burst Time (Quantum = 2)
P1	5
P2	3
P3	1
P4	7
P5	4

# RR: round robin

## Nguyên tắc:

- Mỗi tiến trình được cấp một lượng tử thời gian  $t$  để sử dụng

Process	Burst Time (Quantum = 2)
P1	5
P2	3
P3	1
P4	7
P5	4

- P1:  $0 + (10 - 2) + (18 - 12) = 14$
- P2:  $2 + (12 - 4) = 10$
- P3: 4
- P4:  $6 + (14 - 8) + (20 - 16) = 16$
- P5:  $8 + (16 - 10) = 14$

0	2	4	6	8	10	12	14	16	18	20	22
2	2	2	2	2	2	2	2	2	2	2	2
P1	P2	P3	P4	P5	P1	P2	P4	P5	P1	P4	P4

$$t = \frac{14 + 10 + 4 + 16 + 14}{5} = 11,6$$

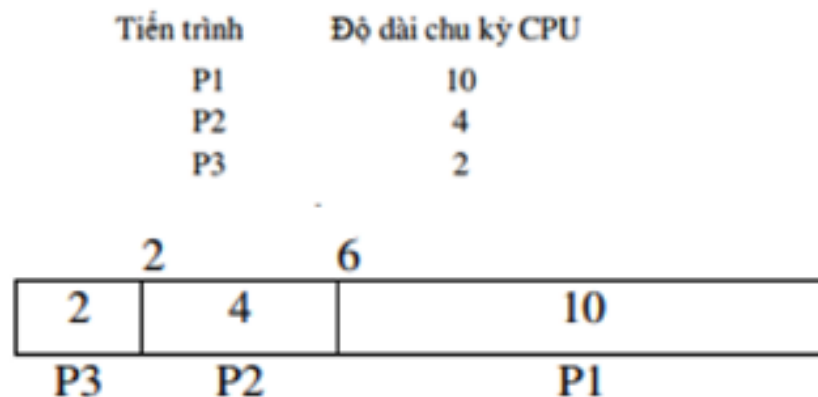


# SPF: Shortest Job First

## Nguyên tắc:

- ▶ Chọn trong hàng đợi tiến trình có chu kỳ sử dụng CPU tiếp theo ngắn nhất để phân phối CPU
- ▶ Nếu có nhiều tiến trình với chu kỳ CPU tiếp theo bằng nhau, chọn tiến trình đứng trước
- ▶ Thời gian chờ đợi trung bình nhỏ hơn nhiều so với FCFS

## Ví dụ:



# SJF: Shortest Job First

5:00

## Nguyên tắc:

- ▶ Chọn trong hàng đợi tiến trình có chu kỳ sử dụng CPU tiếp theo ngắn nhất để phân phối CPU
- ▶ Nếu có nhiều tiến trình với chu kỳ CPU tiếp theo bằng nhau, chọn tiến trình đứng trước

Process	Burst Time
P1	5
P2	3
P3	1
P4	7
P5	4

# SJF: Shortest Job First

## Nguyên tắc:

- ▶ Chọn trong hàng đợi tiến trình có chu kỳ sử dụng CPU tiếp theo ngắn nhất để phân phối CPU
- ▶ Nếu có nhiều tiến trình với chu kỳ CPU tiếp theo bằng nhau, chọn tiến trình đứng trước

Process	Burst Time
P1	5
P2	3
P3	1
P4	7
P5	4

0	1	4	8	13
1	3	4	5	7
P3	P2	P5	P1	P4

$$t = \frac{0 + 1 + 4 + 8 + 13}{5} = 5,2$$

# SRTF: Shortest Remaining Time First

## Nguyên tắc:

- ▶ SPF có thêm cơ chế phân phối lại (SRTF)
- ▶ Khi 1 tiến trình mới xuất hiện trong hàng đợi, HDH so sánh thời gian còn lại của tiến trình đang chạy với thời gian còn lại của tiến trình mới xuất hiện
- ▶ Nếu tiến trình mới xuất hiện có thời gian còn lại ngắn hơn, HDH thu hồi CPU của tiến trình đang chạy, phân phối cho tiến trình mới

Tiến trình	Thời điểm xuất hiện	Độ dài chu kỳ CPU
P1	0	8
P2	0	7
P3	2	4

Kết quả điều độ sử dụng SRTF được thể hiện trên biểu đồ sau:

0	2	6	11
2	4	5	8
P2	P3	P2	P1

# SRTF: Shortest Remaining Time First

Nguyên tắc:

- ▶ SPF có thêm cơ chế phân phối lại (SRTF)
- ▶ Khi 1 tiến trình mới xuất hiện trong hàng đợi, HDH so sánh thời gian còn lại của tiến trình đang chạy với thời gian còn lại của tiến trình mới xuất hiện
- ▶ Nếu tiến trình mới xuất hiện có thời gian còn lại ngắn hơn, HDH thu hồi CPU của tiến trình đang chạy, phân phối cho tiến trình mới

Process	Burst	Arrival
P1	15	0
P2	20	0
P3	20	20
P4	20	25
P5	5	45
P6	15	55



10:00

# SRTF: Shortest Remaining Time First

Process	Burst	Arrival
P1	15	0
P2	20	0
P3	20	20
P4	20	25
P5	5	45
P6	15	55

At Second	Event	Decision
0	P1 and P2 arrival	P1 go to CPU (P1<P2)
15	P1 done	P2 go to CPU
20	P3 Arrival	P2: Remain 15s P3: 20s P2 continue
25	P4 Arrival	P2: Remain 10s P3: 20s P4: 25s P2 continue
35	P2 done	P3: 20s P4: 20s P3 go to CPU

0	15	35	45	50	60	75
P1	P2	P3	P5	P3	P6	P4
15	20	10	5	10	15	20

# SRTF: Shortest Remaining Time First

Process	Burst	Arrival
P1	15	0
P2	20	0
P3	20	20
P4	20	25
P5	5	45
P6	15	55

At Second	Event	Decision
35	P2 done	P3: 20s P4: 20s P3 go to CPU
45	P5 arrival	P3: 10s P4: 20s P5: 5s P3 out P5 go to CPU
50	P5 done	P3: 10s P4: 20s P3 back to CPU
55	P6 arrival	P3: 5s P4: 20s P6: 15s P3 continue

0	15	35	45	50	60	75
P1	P2	P3	P5	P3	P6	P4
15	20	10	5	10	15	20

# SRTF: Shortest Remaining Time First

Proc ess	Bur st	Arrival
P1	15	0
P2	20	0
P3	20	20
P4	20	25
P5	5	45
P6	15	55

At Second	Event	Decision
55	P6 arrival	P3: 5s P4: 20s P6: 15s P3 continue
60	P3 done	P4: 20s P6: 15s P6 go to CPU
75	P6 done	P4 go to CPU
95	P4 done	THE END

0	15	35	45	50	60	75
P1	P2	P3	P5	P3	P6	P4
15	20	10	5	10	15	20



# SRTF: Shortest Remaining Time First

Process	Burst	Arrival	Wait Time
P1	15	0	0
P2	20	0	15
P3	20	20	$(35-20) + (50-45) = 20$
P4	20	25	$(75-25) = 55$
P5	5	45	0
P6	15	55	$(60-55) = 5$

0	15	35	45	50	60	75
P1	P2	P3	P5	P3	P6	P4
15	20	10	5	10	15	20

$$t = \frac{0 + 15 + 20 + 55 + 0 + 5}{6} = 15,83$$

# SRTF: Shortest Remaining Time First

Nguyên tắc:

- ▶ SPF có thêm cơ chế phân phối lại (SRTF)
- ▶ Khi 1 tiến trình mới xuất hiện trong hàng đợi, HDH so sánh thời gian còn lại của tiến trình đang chạy với thời gian còn lại của tiến trình mới xuất hiện
- ▶ Nếu tiến trình mới xuất hiện có thời gian còn lại ngắn hơn, HDH thu hồi CPU của tiến trình đang chạy, phân phối cho tiến trình mới

5:00

<u>Thứ tự</u>	<u>Thời điểm kích hoạt</u>	<u>Thời gian hoạt động (ms)</u>
p1	3	37
p2	10	20
p3	24	14

# SRTF: Shortest Remaining Time First

<u>Thứ tự</u>	<u>Thời điểm kích hoạt</u>	<u>Thời gian hoạt động (ms)</u>
p1	3	37
p2	10	20
p3	24	14

At Second	Event	Decision
3	P1 arrival	P1 go to CPU
10	P2 arrival	P1: Remain 30 P2: 20 P1 out P2 go to CPU
24	P3 Arrival	P1: Remain 30 P2: 6 P3: 14 P2 continue

3	10
P1	P2
7	

# SRTF: Shortest Remaining Time First

<u>Thứ tự</u>	<u>Thời điểm kích hoạt</u>	<u>Thời gian hoạt động (ms)</u>
p1	3	37
p2	10	20
p3	24	14

At Second	Event	Decision
24	P3 Arrival	P1: Remain 30 P2: 6 P3: 14 P2 continue
30	P2 done	P1: Remain 30 P3: 14 P3 go to CPU
44	P3 Done	P1 go to CPU

3	10	30	44
P1	P2	P3	P1
7	20	14	30



# SRTF: Shortest Remaining Time First

<u>Thứ tự</u>	<u>Thời điểm kích hoạt</u>	<u>Thời gian hoạt động (ms)</u>
p1	3	37
p2	10	20
p3	24	14

Proccess	Wait Time
P1	$(3-3) + (44 - 10) = 34$
P2	0
P3	$30 - 24 = 6$

3	10	30	44
P1	P2	P3	P1
7	20	14	30

$$t = \frac{34 + 0 + 6}{3} = 13,3$$

# Priority Scheduling

## Nguyên tắc:

- ▶ Mỗi tiến trình có 1 mức ưu tiên
- ▶ Tiến trình có mức ưu tiên cao hơn sẽ được cấp CPU trước
- ▶ Các tiến trình có mức ưu tiên bằng nhau được điều độ theo FCFS
- ▶ Mức ưu tiên được xác định theo nhiều tiêu chí khác nhau

## Ví dụ:

Tiến trình	Mức ưu tiên
P1	4
P2	1
P3	3

P2	P3	P1



5:00

# Priority Scheduling

- ▶ Tiến trình có mức ưu tiên cao hơn sẽ được cấp CPU trước
- ▶ Các tiến trình có mức ưu tiên bằng nhau được điều độ theo FCFS
- ▶ Số lượng ưu tiên lớn hơn nghĩa là mức độ ưu tiên cao hơn

Process	Burst Time	Priority Simp
P1	5	4
P2	3	1
P3	1	2
P4	7	2
P5	4	3

# Priority Scheduling

- ▶ Tiến trình có mức ưu tiên cao hơn sẽ được cấp CPU trước
- ▶ Các tiến trình có mức ưu tiên bằng nhau được điều độ theo FCFS
- ▶ Số lượng ưu tiên lớn hơn nghĩa là mức độ ưu tiên cao hơn

Process	Burst Time	Priority Simp
P1	5	4
P2	3	1
P3	1	2
P4	7	2
P5	4	3

0	5	9	10	17
5	4	1	7	3
P1	P5	P3	P4	P2

$$t = \frac{0 + 5 + 9 + 10 + 17}{5} = 8,2$$





5:00

# Priority Scheduling

- ▶ Tiến trình có mức ưu tiên cao hơn sẽ được cấp CPU trước
- ▶ Các tiến trình có mức ưu tiên bằng nhau được điều độ theo FCFS

**Câu 1:** Cho các tiến trình với thời điểm xuất hiện, thời gian (chu kỳ) CPU tiếp theo và số ưu tiên như trong bảng sau (số ưu tiên nhỏ ứng với độ ưu tiên cao). Vẽ biểu đồ thể hiện thứ tự và thời gian cấp phát CPU cho các tiến trình sử dụng thuật toán : Điều độ theo mức ưu tiên

Tiến trình	Thời điểm xuất hiện (c)	Thời gian (độ dài)	Số ưu tiên
P1	0	10	1
P2	2	8	3
P3	4	5	2
P4	5	3	2

# Priority Scheduling

- ▶ Tiến trình có mức ưu tiên cao hơn sẽ được cấp CPU trước
- ▶ Các tiến trình có mức ưu tiên bằng nhau được điều độ theo FCFS

Tiến trình	Thời điểm xuất hiện (c)	Thời gian (độ dài)	Số ưu tiên
P1	0	10	1
P2	2	8	3
P3	4	5	2
P4	5	3	2

0	10	15	18
10	5	3	8
P1	P3	P4	P2

$$T = [(0-0) + (10-4) + (15 - 5) + (18 - 2)]/4 = 8$$

# Priority Scheduling

19

Các tiến trình sau đây đang được lập lịch như hình dưới đây. Mỗi tiến trình được chỉ định một mức độ ưu, với số lớn hơn cho thấy mức độ ưu tiên cao hơn. Xét mức độ ưu tiên dựa trên thời gian bắt đầu của các tiến trình đối với giải thuật theo mức độ ưu tiên, tính thời gian chờ trung bình: \* (4 Điểm)

<u>Process</u>	<u>Priority</u>	<u>Burst</u>	<u>Arrival</u>
$P_1$	8	15	0
$P_2$	3	20	0
$P_3$	4	20	20
$P_4$	4	20	25
$P_5$	5	5	45
$P_6$	5	15	55