

NGUYÊN LÝ HỆ ĐIỀU HÀNH

Giảng viên: TS. Đoàn Thị Quế
Bộ môn Kỹ thuật máy tính & mạng

CHƯƠNG 2: TIẾN TRÌNH VÀ LUỒNG

- Các khái niệm liên quan đến tiến trình
- Luồng
- Điều độ tiến trình

2.1 – Các khái niệm liên quan đến tiến trình

- Tiến trình là gì?
- Mô hình tiến trình
- Trạng thái của tiến trình
- Thông tin mô tả tiến trình
- Khối quản lý, bảng và danh sách tiến trình
- Các thao tác với tiến trình

Tiến trình là gì?

- Một tiến trình (Process) là một chương trình đang thi hành.
- Mỗi tiến trình cần:
 - Một không gian địa chỉ trong vùng nhớ: chứa chương trình thi hành, dữ liệu của chương trình và ngăn xếp của nó.
 - Các thanh ghi (bao gồm con trỏ lệnh, con trỏ ngăn xếp và các thanh ghi khác) để lưu các thông tin cần thiết cho việc chạy chương trình.

- Hai đặc điểm để phân biệt tiến trình với chương trình:
 - Chương trình là một thực thể tĩnh, tiến trình là thực thể động:
 - ◆ Chương trình bao gồm tập lệnh và dữ liệu chứa trong file, file không thay đổi theo thời gian.
 - ◆ Tiến trình bao gồm các lệnh, dữ liệu, không gian nhớ, con trỏ lệnh chỉ tới lệnh và các thanh ghi. Hầu hết các thành phần này đều thay đổi trong quá trình tiến trình tồn tại.
 - Chương trình không được cấp tài nguyên cụ thể, tiến trình được cấp tài nguyên (bộ nhớ, các thiết bị vào/ra, thời gian sử dụng CPU).

Mô hình tiến trình

- **Thực hiện chương trình trong máy tính:**

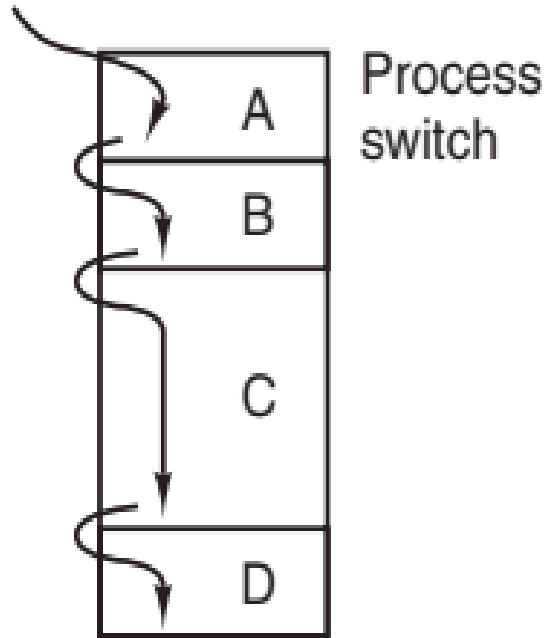
- **Thực hiện tuần tự:** Tại một thời điểm chỉ có thể thực hiện một chương trình, khi chương trình đó thực hiện xong thì mới chạy được chương trình khác
- **Thực hiện song song:** Có thể chạy nhiều chương trình cùng một lúc (hệ thống đa xử lý)

■ Thực hiện chương trình trong hệ thống đa chương trình:

- Tại một thời điểm CPU chỉ chạy được một tiến trình, sau đó sẽ chuyển CPU cho tiến trình khác
- Mỗi tiến trình sẽ được chạy trong khoảng thời gian vài chục hay vài trăm mili giây. Trong một giây, CPU thực hiện được vài tiến trình → *Tạo ra cảm giác các chương trình chạy song song (giả song song)*
- Việc quản lý các hoạt động song song có nhiều khó khăn.

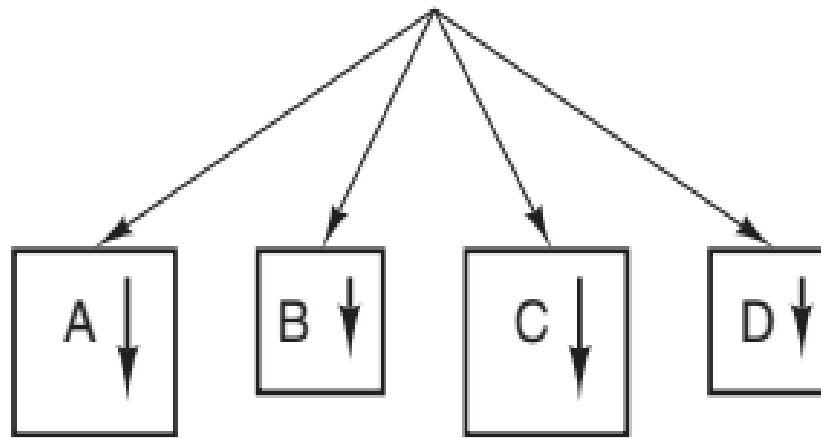
➔ *Cần xây dựng mô hình khái niệm (về tiến trình) để quản lý hoạt động song song một cách dễ dàng hơn*

One program counter

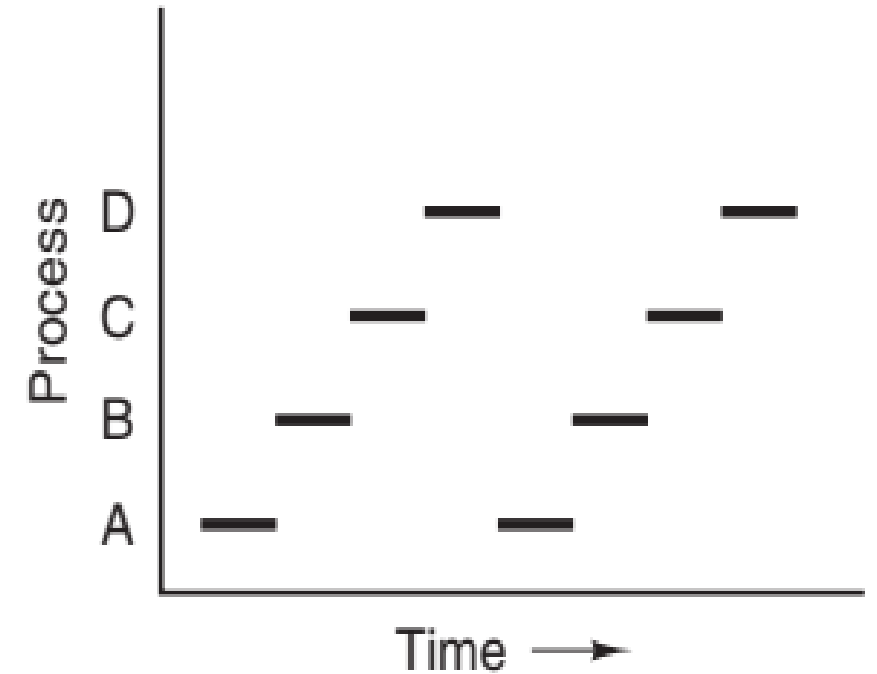


(a)

Four program counters



(b)



(c)

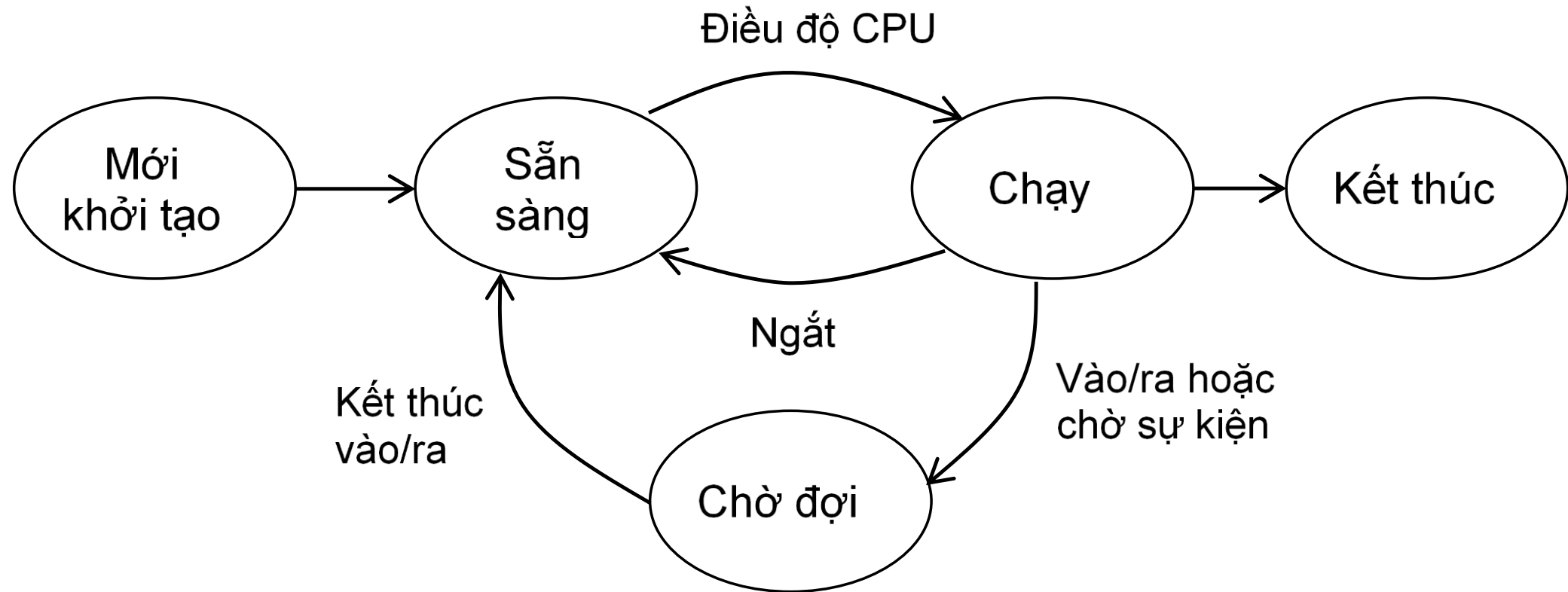
- (a) Thực hiện đa chương trình với bốn tiến trình đồng thời.
- (b) Mô hình khái niệm của bốn tiến trình tuần tự, độc lập.
- (c) Chỉ có một tiến trình hoạt động tại một thời điểm.

Trạng thái của tiến trình

Tại một thời điểm, tiến trình có thể ở một trong các trạng thái sau:

- **Trạng thái mới khởi tạo:** tiến trình đang được tạo ra
- **Trạng thái sẵn sàng:** Tiến trình đã được phân phối đầy đủ mọi tài nguyên, chỉ trừ CPU
- **Trạng thái chạy:** là trạng thái mà tiến trình đang sử dụng CPU để thực hiện lệnh
- **Trạng thái chờ đợi:** Tiến trình còn thiếu một vài điều kiện nào đó thì mới chuyển sang trạng thái sẵn sàng
- **Trạng thái kết thúc:** tiến trình không còn nằm trong danh sách các tiến trình được thực hiện.

Quá trình chuyển đổi trạng thái



Sơ đồ chuyển đổi giữa các trạng thái của tiến trình

Thông tin mô tả tiến trình

- *Làm cách nào để phân biệt được tiến trình này với tiến trình khác trong hệ thống và quản lý việc thực hiện các tiến trình đó?*

➔ *Mỗi tiến trình cần có thông tin riêng để mô tả tiến trình*

Thông tin mô tả tiến trình

Thông tin mô tả tiến trình gồm:

- **Số định danh của tiến trình (PID-Process Identification)** cho phép phân biệt với tiến trình khác.
- **Trạng thái tiến trình:** một trong năm trạng thái liệt kê ở phần trước.
- **Nội dung một số thanh ghi CPU:** thanh ghi con trỏ lệnh, con trỏ ngăn xếp ngăn xếp, thanh ghi trạng thái, ...

Thông tin mô tả tiến trình

Thông tin mô tả tiến trình gồm (tiếp):

- **Thông tin về bộ nhớ của tiến trình:** chỉ ra nơi chứa tiến trình.
- **Thông tin phục vụ việc điều độ tiến trình:** mức độ ưu tiên của tiến trình, vị trí tiến trình trong các hàng đợi, tài nguyên tiến trình đang sở hữu.

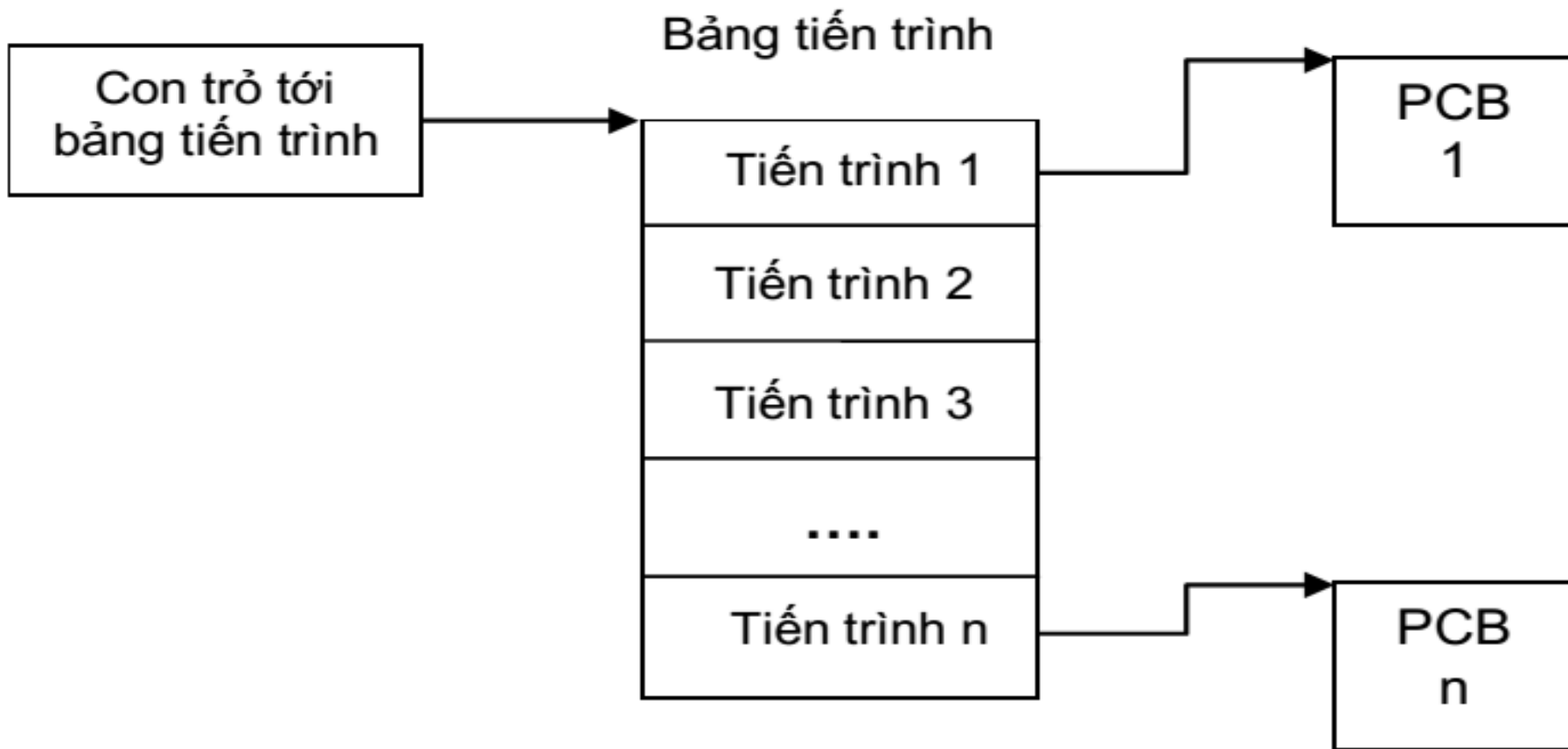
Thông tin mô tả tiến trình

Thông tin mô tả tiến trình gồm (tiếp):

- **Danh sách các tài nguyên khác:** bao gồm danh sách các file đang mở của tiến trình, các thiết bị vào ra tiến trình đang sử dụng.
- **Thông tin thống kê phục vụ quản lý:** thông tin loại này thường được sử dụng phục vụ thống kê hoặc tính toán chi phí đối với các hệ thống dùng chung (như khi đi thuê máy tính) và bao gồm thông tin về thời gian sử dụng CPU, giới hạn thời gian, tài khoản của người sở hữu tiến trình .v.v.

Khối quản lý, bảng và danh sách tiến trình

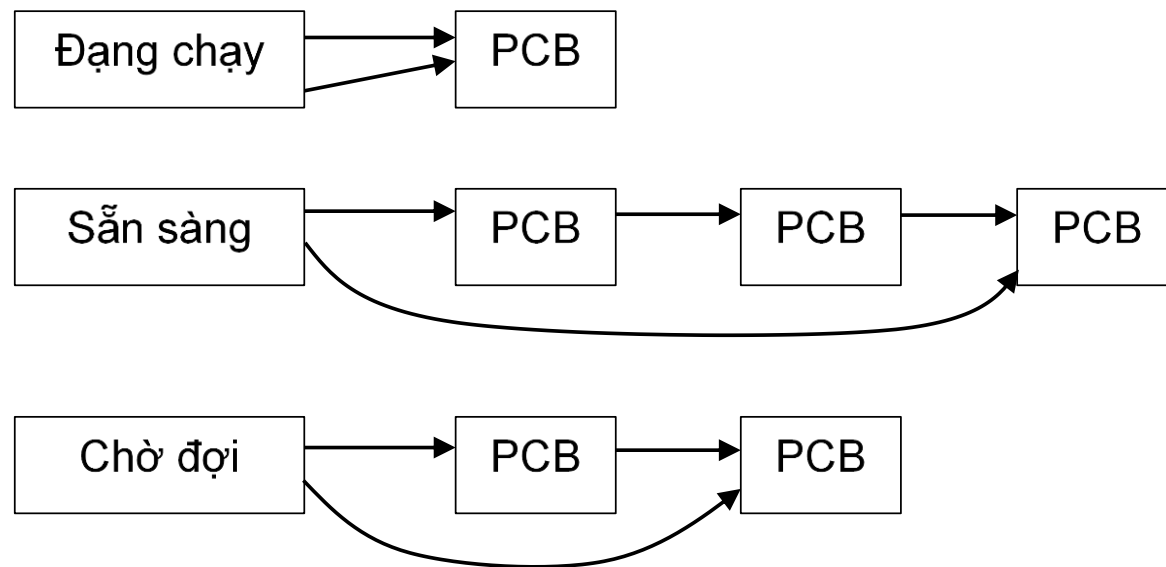
- **Khối quản lý tiến trình (PCB-Process Control Block):**
 - là một **cấu trúc dữ liệu chứa các thông tin mô tả tiến trình**, được lưu trong bộ nhớ trong và có thể nằm ở các vị trí khác nhau.
- **Hệ điều hành cần lưu PCB và có cách xác định PCB**
 - Hệ điều hành sử dụng **bảng tiến trình** chứa con trỏ tới PCB của toàn bộ tiến trình có trong hệ thống.



Bảng tiến trình chứa con trỏ tới các PCB.

■ Danh sách tiến trình:

- bao gồm tiến trình có cùng trạng thái hoặc tiến trình đang cùng chờ đợi một tài nguyên nào đó



Danh sách liên kết PCB thuộc các trạng thái khác nhau

Các thao tác với tiến trình

- Tạo tiến trình
- Kết thúc tiến trình
- Chuyển đổi giữa các tiến trình
- Điều độ tiến trình
- ...

Tạo tiến trình

- **Các tiến trình được tạo ra trong các tình huống sau:**

- Khởi động hệ thống
- Người dùng kích hoạt một chương trình
- Một tiến trình đang chạy gọi hàm hệ thống để tạo tiến trình mới.
 - ◆ Unix: `fork()`
 - ◆ Windows: `CreateProcess()`

- **Hai kiểu tạo mới tiến trình:**

- Tiến trình con là một bản sao của tiến trình cha, tức là có cùng phần mã chương trình và phần dữ liệu (Unix: `fork()`)
- Tiến trình con được tạo thành từ một chương trình mới (Windows: `CreateProcess()`).

Kết thúc tiến trình

- Xử lý xong lệnh cuối cùng hay gọi lệnh kết thúc
 - Unix: `Exit()`
 - Windows: `ExitProcess()`
- Bị tiến trình cha kết thúc
- Do các lỗi
- ...

Chuyển đổi giữa các tiến trình

- Việc chuyển tiến trình xảy ra trong hai trường hợp sau:
 - Khi có ngắt
 - Khi tiến trình gọi lời gọi hệ thống
- Các hoạt động xảy ra khi chuyển tiến trình:
 - Trước khi chuyển sang thực hiện tiến trình khác, ngữ cảnh (bao gồm nội dung các thanh ghi, thông tin về bộ nhớ, ...) được lưu vào PCB.
 - Khi tiến trình được cấp phát CPU để thực hiện trở lại, ngữ cảnh sẽ được khôi phục từ PCB vào thanh ghi và những bảng tương ứng.

CHƯƠNG 2: TIẾN TRÌNH VÀ LUỒNG

- Các khái niệm liên quan đến tiến trình
- **Luồng**
- Điều độ tiến trình

2.2 - Luồng

- **Luồng là gì?**
- **Tài nguyên của tiến trình và luồng**
- **Ưu điểm của mô hình đa luồng**

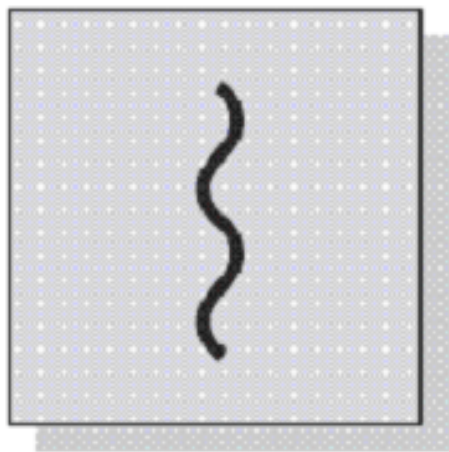
Luồng là gì?

- **Tiến trình có thể được xem xét từ hai khía cạnh:**
 - **Tiến trình là một đơn vị sở hữu tài nguyên:**
 - ◆ Tiến trình được cấp bộ nhớ để chứa mã lệnh, dữ liệu, cũng như các tài nguyên khác.
 - **Tiến trình là một đơn vị thực hiện công việc tính toán xử lý:**
 - ◆ Tiến trình được cấp CPU để thực hiện các lệnh của mình

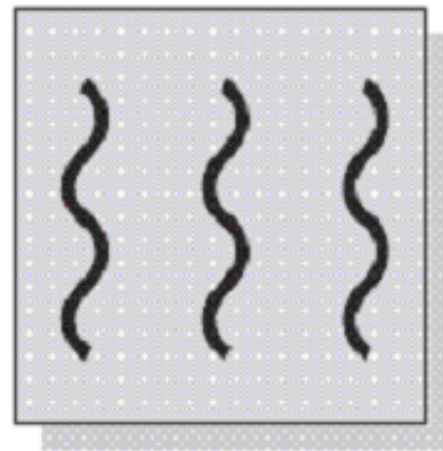
Luồng là gì?

- Hệ điều hành trước đây, mỗi tiến trình chỉ tương ứng với một đơn vị xử lý duy nhất:
 - các lệnh được thực hiện một cách tuần tự
 - mỗi tiến trình chỉ thực hiện được một công việc ở một thời điểm
- Hệ điều hành hiện đại, mỗi tiến trình có nhiều đơn vị xử lý:
 - Mỗi đơn vị thực hiện của tiến trình là một **chuỗi các lệnh** được cấp phát CPU để **thực hiện độc lập** (song song), được gọi là **luồng**

Minh họa khái niệm luồng trong tiến trình



tiến trình gồm một luồng



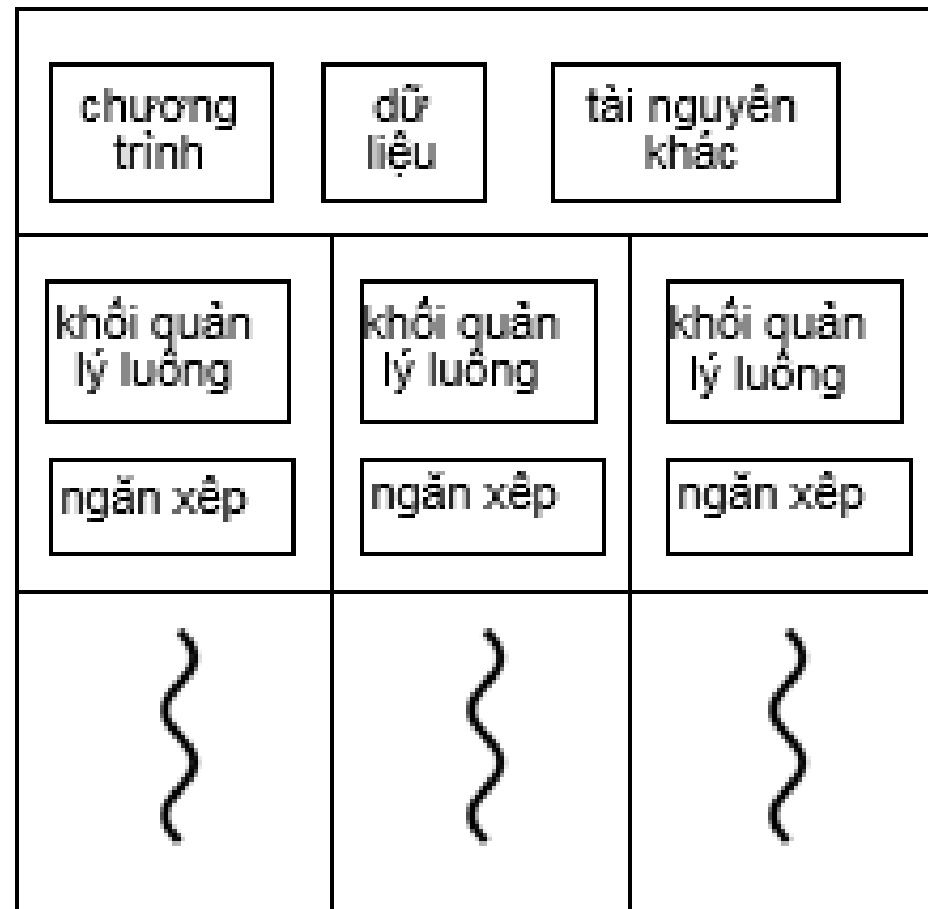
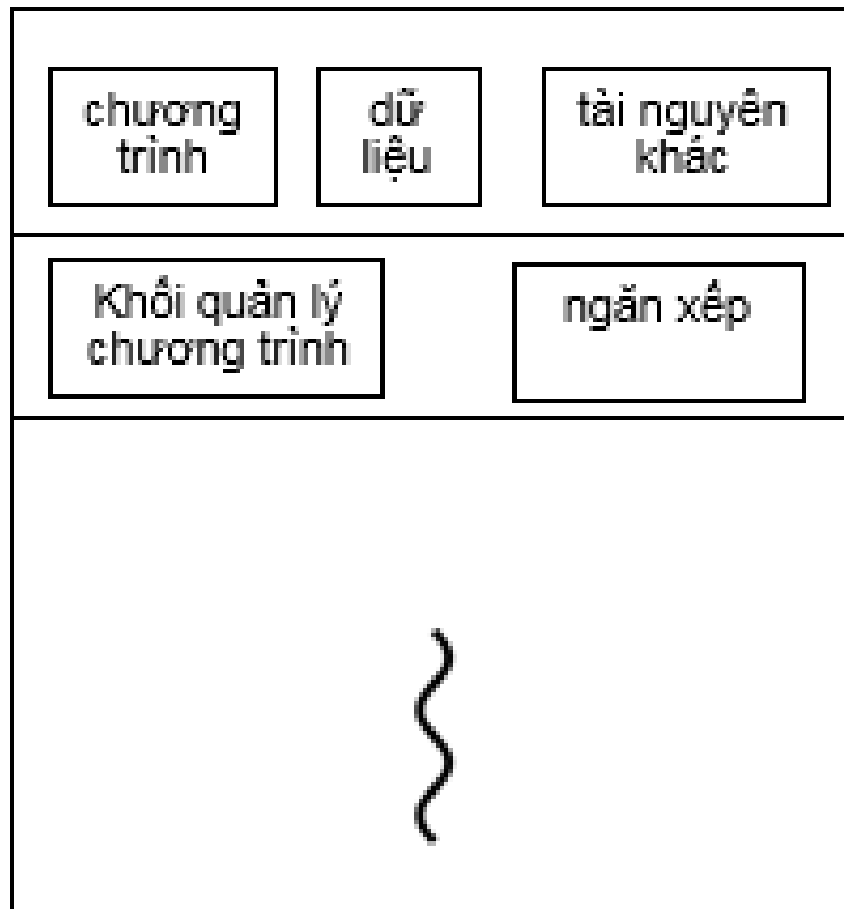
tiến trình gồm nhiều luồng

} = chuỗi lệnh

Tiến trình và luồng

Tài nguyên của tiến trình và luồng

- Trong hệ thống cho phép đa luồng, tiến trình vẫn là đơn vị được hệ điều hành sử dụng để phân phối tài nguyên.
- Mỗi tiến trình và tất cả các luồng thuộc tiến trình đó sẽ sở hữu chung một số số tài nguyên bao gồm:
 - Không gian nhớ của tiến trình. Đây là không gian nhớ logic, có thể là không gian nhớ ảo, được sử dụng để chứa phần chương trình (các lệnh), phần dữ liệu của tiến trình.
 - Các tài nguyên khác như file do tiến trình mở, thiết bị hoặc cổng vào/ra.



Mô hình đơn luồng và đa luồng

Ưu điểm của mô hình đa luồng

- Tăng hiệu năng và tiết kiệm thời gian.
- Dễ dàng chia sẻ tài nguyên và thông tin.
- Tăng tính đáp ứng.
- Tận dụng được kiến trúc xử lý với nhiều CPU.
- Thuận lợi cho việc tổ chức chương trình.

CHƯƠNG 2: TIẾN TRÌNH VÀ LUỒNG

- Các khái niệm liên quan đến tiến trình
- Luồng
- **Điều độ tiến trình**

2.3 – Điều độ tiến trình

- Giới thiệu điều độ
- Yêu cầu điều độ
- Mục tiêu điều độ
- Các thuật toán điều độ

Giới thiệu điều độ

- **Điều độ tiến trình** (*scheduling*) (theo nghĩa tổng quát) là sự tổ chức thực hiện các tiến trình theo một giải thuật điều độ (*scheduling algorithm*) nào đó, nhằm tránh sự xung đột giữa các tiến trình về mặt tài nguyên.
- **Điều độ tiến trình** (theo nghĩa hẹp) là sự tổ chức, phân phối tài nguyên CPU cho các tiến trình theo một giải thuật nào đó, nhằm đảm bảo tốc độ thực hiện của các tiến trình và hiệu quả sử dụng CPU.

Giới thiệu điều độ

■ Hệ thống đa chương trình:

- Nhiều tiến trình cùng muốn giành quyền sử dụng CPU tại một thời điểm.
- Tại một thời điểm, chỉ có duy nhất một tiến trình được thực hiện tại CPU.
- Nếu chỉ có một CPU, sẽ phải lựa chọn tiến trình nào được sử dụng CPU khi nào và trong thời gian bao lâu.
 - ◆ Việc lựa chọn gọi là điều độ tiến trình (**scheduling**), gọi tắt là điều độ.
 - ◆ Bộ phận thực hiện việc lựa chọn gọi là bộ điều độ (**scheduler**).
 - ◆ Thuật toán để lựa chọn được gọi là thuật toán điều độ (**scheduling algorithm**)

Yêu cầu điều độ

- Tài nguyên găng

- Tài nguyên găng là loại tài nguyên mà tại một thời điểm chỉ có thể phục vụ cho một đối tượng

- ◆ Ví dụ: CPU, một ô nhớ, máy in, ...

- Công việc điều độ phải thoả mãn các yêu cầu sau:

- 1) Mỗi thời điểm chỉ có một tiến trình nằm trong đoạn găng
- 2) Không tiến trình nào được phép ở lâu vô hạn trong đoạn găng
- 3) Không tiến trình nào phải chờ vô hạn trước đoạn găng

Mục tiêu điều độ

Đối với tất cả các hệ thống:

Công bằng – các tiến trình đều công bằng trong việc chia sẻ CPU

Tuân thủ chính sách – các chính sách đã đề ra phải được thực hiện

Cân bằng - đảm bảo tất cả các bộ phận của hệ thống đều phải làm việc

Đối với hệ thống nhóm:

Thông lượng - tối đa hoá lượng công việc được hiện trong mỗi giờ

Thời gian thực hiện - giảm thiểu thời gian từ lúc bắt đầu tới khi kết thúc

Tận dụng CPU – CPU phải luôn hoạt động

Đối với hệ thống tương tác:

Thời gian đáp ứng – đáp ứng lại các yêu cầu một cách nhanh chóng

Sự cân xứng – đáp lại sự mong đợi của người dùng

Đối với hệ thống thời gian thực:

Phải có hạn chót – tránh mất mát dữ liệu

Khả năng dự báo – tránh làm giảm chất lượng trong các hệ thống đa phương tiện

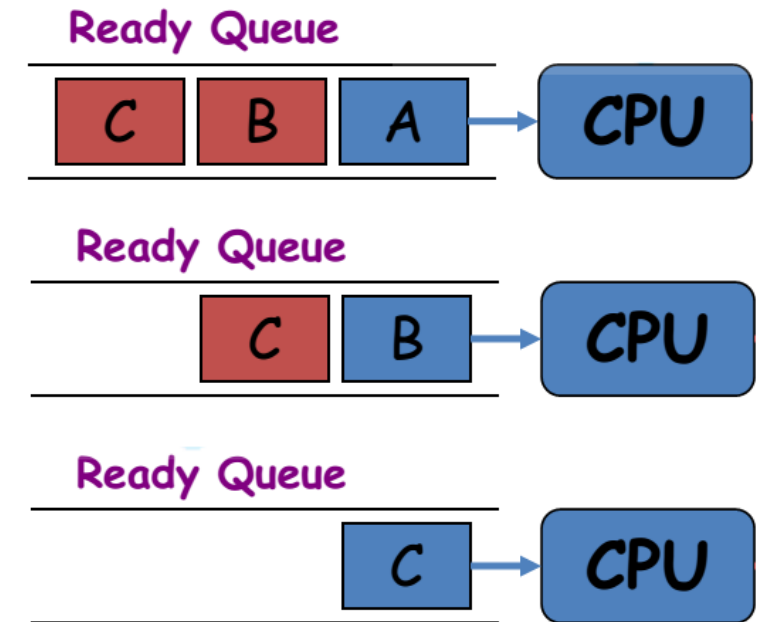
Các thuật toán điều độ

- **FCFS (First-Come First-Served)**
Đến trước phục vụ trước
- **RR (Round-Robin)**
Quay vòng
- **SJF (Shortest Job First)**
Ưu tiên tiến trình ngắn nhất
- **SRTN (Shortest Remaining Time Next)**
Ưu tiên tiến trình có thời gian còn lại ngắn nhất

FCFS (First-Come First-Served)

Đến trước phục vụ trước

- Tiến trình yêu cầu CPU trước sẽ được cấp CPU trước. Tiến trình độc quyền chiếm dụng CPU.
- Hệ điều hành xếp tiến trình sẵn sàng vào hàng đợi FIFO. Tiến trình mới được xếp vào cuối hàng đợi.
- Hệ điều hành sẽ lấy tiến trình từ đầu hàng đợi và cấp CPU cho tiến trình đó thực hiện.



Tiến trình	T
P_1	
P_2	
P_3	

Tiến trình	Thời điểm xuất hiện	Thời gian sử dụng CPU
P_1	0	23
P_2	1	2
P_3	2	5

*Thời gian lưu lại hệ thống của một tiến trình
= Thời điểm tiến trình kết thúc sử dụng CPU
– Thời điểm xuất hiện*

Thời gian chờ của tiến trình = Thời gian lưu lại hệ thống của tiến trình – Thời gian tiến trình sử dụng CPU



0
30

0: P_1 vào RQ
 P_1 dùng CPU

1: P_2 vào RQ

2: P_3 vào RQ

23: P₁ kết thúc
P₂ dùng CPU

25: P₂ kết thúc
P₃ dùng CPU

30: P₃ kết thúc

Tiến trình	Thời gian lưu lại hệ thống	Thời gian chờ
P_1	$23 - 0 = 23$	$23 - 23 = 0$
P_2	$25 - 1 = 24$	$24 - 2 = 22$
P_3	$30 - 2 = 28$	$28 - 5 = 23$

Thời gian chờ trung bình = $(0+22+23)/3 = 15$

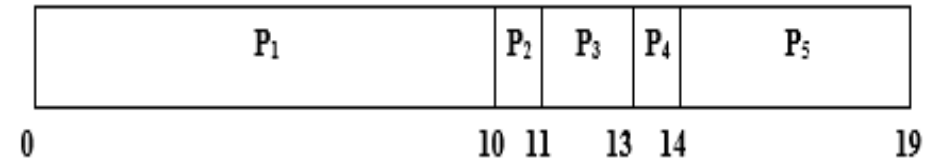
Bài tập áp dụng

- Thực hiện điều phối theo thuật toán FCFS
- Tính thời gian chờ trung bình của mỗi tiến trình

Tiến trình	Thời điểm xuất hiện	Thời gian sử dụng CPU
P_1	0	10
P_2	1	1
P_3	2	2
P_4	3	1
P_5	4	5

Kết quả

Tiến trình	Thời điểm xuất hiện	Thời gian sử dụng CPU
P ₁	0	10
P ₂	1	1
P ₃	2	2
P ₄	3	1
P ₅	4	5



Thời gian chờ của mỗi tiến trình

$$P_1: 10 - (0 + 10) = 0$$

$$P_2: 11 - (1 + 1) = 9$$

$$P_3: 13 - (2 + 2) = 9$$

$$P_4: 14 - (3 + 1) = 10$$

$$P_5: 19 - (4 + 5) = 10$$

$$\text{Thời gian chờ trung bình} = (0 + 9 + 9 + 10 + 10) / 5 = 7.6$$

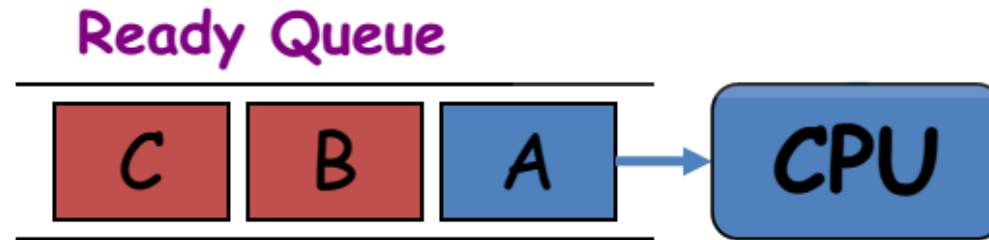
Nhận xét FCFS

- Đơn giản, dễ cài đặt
- Chịu đựng hiện tượng tích lũy thời gian chờ
 - Tiến trình có thời gian xử lý ngắn đợi tiến trình có thời gian xử lý dài
- Có thể xảy ra tình trạng độc chiếm CPU
- Không phù hợp với hệ thống tương tác

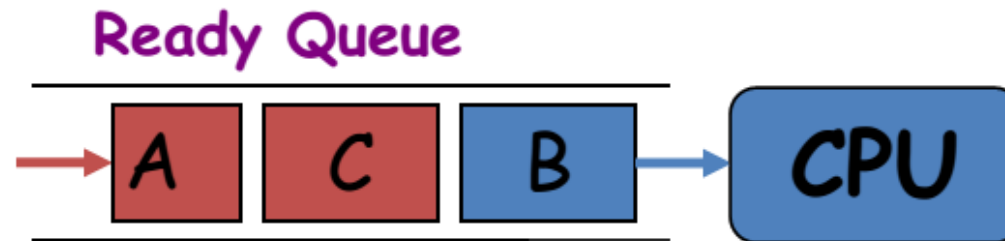
RR (Round-Robin)

Quay vòng

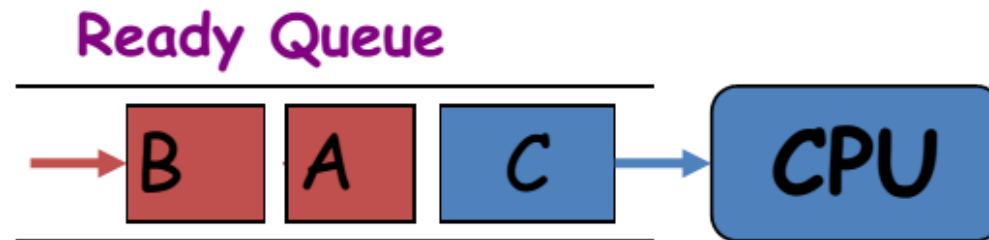
- RR tương tự FCFS nhưng có thể cơ chế phân phối lại bằng cách sử dụng ngắt của đồng hồ.
- Tiến trình sẽ lần lượt được cấp CPU trong những khoảng thời gian q (time quantum/time slice)



A chỉ được chiếm CPU trong khoảng thời gian q



B được giao quyền sử dụng CPU trong khoảng thời gian q kế tiếp



C được giao quyền sử dụng CPU trong khoảng thời gian q kế tiếp

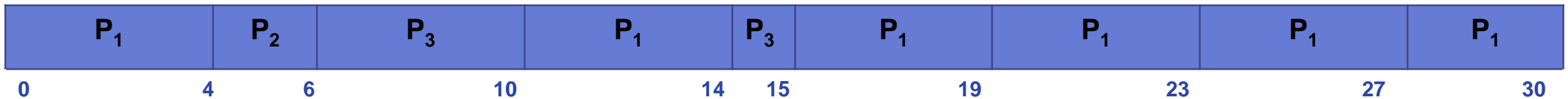
Minh họa RR

Tiến trình	Thời điểm xuất hiện	Thời gian sử dụng CPU
P_1	0	23
P_2	1	2
P_3	2	5

$$q = 4$$

Tiến trình	Thời gian lưu lại hệ thống	Thời gian chờ
P_1	$30 - 0 = 30$	$30 - 23 = 7$
P_2	$6 - 1 = 5$	$5 - 2 = 3$
P_3	$15 - 2 = 13$	$13 - 5 = 8$

$$\text{Thời gian chờ trung bình} = (7 + 3 + 8) / 3 = 6$$



0: P_1 vào RQ
 P_1 dùng CPU

4: P_1 hết lượt, xếp vào RQ (sau P_3)
 P_2 dùng CPU

14: P_1 hết lượt, xếp vào RQ (sau P_3)
 P_3 dùng CPU

23: P_1 dùng CPU

1: P_2 vào RQ

6: P_2 kết thúc
 P_3 dùng CPU

15: P_3 kết thúc
 P_1 dùng CPU

27: P_1 dùng CPU

2: P_3 vào RQ

10: P_3 hết lượt, xếp vào RQ (sau P_1)
 P_1 dùng CPU

19: P_1 dùng CPU

30: P_1 kết thúc

Bài tập áp dụng

- Thực hiện điều phối theo thuật toán RR
- Tính thời gian chờ trung bình, $q = 2$

Tiến trình	Thời điểm xuất hiện	Thời gian sử dụng CPU
P_1	0	10
P_2	1	1
P_3	2	2
P_4	3	1
P_5	4	5

Kết quả

Tiến trình	Thời điểm xuất hiện	Thời gian sử dụng CPU
P_1	0	10
P_2	1	1
P_3	2	2
P_4	3	1
P_5	4	5

Nhận xét RR

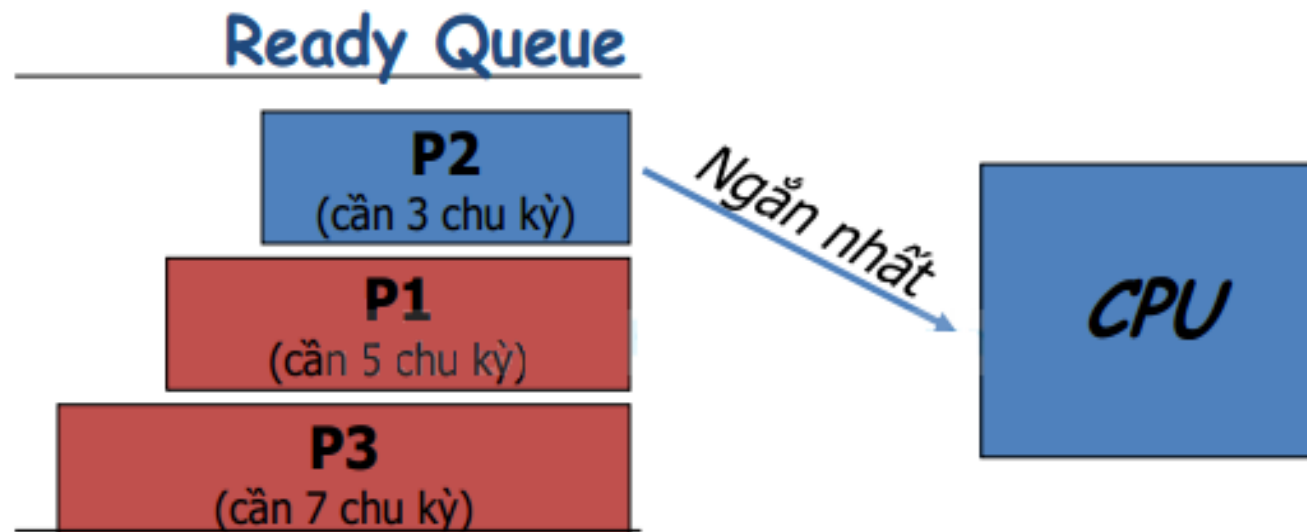


- Giảm thời gian chờ đợi trung bình
- Loại bỏ hiện tượng độc chiếm CPU
- Phù hợp với hệ thống tương tác người dùng

SJF (Shortest Job First)

Ưu tiên tiến trình ngắn nhất

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



Minh họa SJF

Tiến trình	Thời điểm xuất hiện	Thời gian sử dụng CPU
P ₁	0	10
P ₂	1	1
P ₃	2	2
P ₄	3	1
P ₅	4	5

Tiến trình	Thời gian lưu lại hệ thống	Thời gian chờ
P ₁	$10 - 0 = 10$	$10 - 10 = 0$
P ₂	$11 - 1 = 10$	$10 - 1 = 9$
P ₃	$14 - 2 = 12$	$12 - 2 = 10$
P ₄	$12 - 3 = 9$	$9 - 1 = 8$
P ₅	$19 - 4 = 15$	$15 - 5 = 10$



0: P₁ vào RQ
P₁ dùng CPU

10: P₁ kết thúc
P₂ dùng CPU

12: P₄ kết thúc
P₃ dùng CPU

19: P₅ kết thúc

1: P₂ vào RQ
2: P₃ vào RQ
3: P₄ vào RQ
4: P₅ vào RQ

11: P₂ kết thúc
P₄ dùng CPU

14: P₃ kết thúc
P₅ dùng CPU

Thời gian chờ trung bình:
 $(0+9+10+8+10)/5 = 7.4$

Bài tập áp dụng

- Thực hiện điều phối theo thuật toán SJF
- Tính thời gian chờ trung bình

Tiến trình	Thời điểm xuất hiện	Thời gian sử dụng CPU
P_1	0	6
P_2	1	3
P_3	2	4
P_4	3	2

Kết quả

Tiến trình	Thời điểm xuất hiện	Thời gian sử dụng CPU
P_1	0	6
P_2	1	3
P_3	2	4
P_4	3	2

Nhận xét SJF

- Tối ưu thời gian chờ trung bình
- Phải biết trước độ dài chu kỳ sử dụng CPU của tiến trình tiếp theo (rất khó)
 - ⇒ Không khả thi

SRTN (Shortest Remaining Time Next)

Ưu tiên tiến trình có thời gian còn lại ngắn nhất

- Ưu tiên xử lý tiến trình có thời gian thực hiện còn lại ngắn nhất
- Khi một tiến trình mới xuất hiện, thời gian thực hiện của nó được so sánh với thời gian thực hiện còn lại của tiến trình đang chạy. Nếu tiến trình mới có thời gian thực hiện ngắn hơn, nó sẽ được chọn để chạy, còn tiến trình đang chạy sẽ bị treo.

Minh họa SRTN

Tiến trình	Thời điểm xuất hiện	Thời gian sử dụng CPU
P ₁	0	10
P ₂	1	1
P ₃	2	2
P ₄	3	1
P ₅	4	5

Tiến trình	Thời gian lưu lại hệ thống	Thời gian chờ
P ₁	$19 - 0 = 19$	$19 - 10 = 9$
P ₂	$2 - 1 = 1$	$1 - 1 = 0$
P ₃	$4 - 2 = 2$	$2 - 2 = 0$
P ₄	$5 - 3 = 2$	$2 - 1 = 1$
P ₅	$10 - 4 = 6$	$6 - 5 = 1$



0: P₁ vào RQ
P₁ dùng CPU
1: P₂ vào RQ
P₂ dùng CPU
2: P₃ vào RQ
P₂ kết thúc
P₃ dùng CPU

3: P₄ vào RQ
P₃ vẫn được dùng CPU
4: P₅ vào RQ
P₃ kết thúc
P₄ dùng CPU

5: P₄ kết thúc
P₅ dùng CPU
10: P₅ kết thúc
P₁ dùng CPU
19: P₁ kết thúc

Thời gian chờ trung bình:
 $(9+0+0+1+1)/5 = 2.2$

Bài tập áp dụng

- Thực hiện điều phối theo thuật toán SRTN
- Tính thời gian trễ trung bình

Tiến trình	Thời điểm xuất hiện	Thời gian sử dụng CPU
P_1	0	6
P_2	1	3
P_3	2	4
P_4	3	2

Kết quả

Tiến trình	Thời điểm xuất hiện	Thời gian sử dụng CPU
P_1	0	6
P_2	1	3
P_3	2	4
P_4	3	2

Nhận xét SRTN

- Thời gian chờ đợi trung bình nhỏ
- Phải dự đoán được độ dài chu kỳ sử dụng CPU của tiến trình. So với điều độ quay vòng, việc chuyển đổi tiến trình diễn ra ít hơn và do vậy không tốn nhiều thời gian chuyển đổi ngữ cảnh.

Hết Chương 2