

Chương 4: Quản lý tiến trình

Hệ điều hành

ThS. Đinh Xuân Trường
truongdx@ptit.edu.vn



Posts and Telecommunications
Institute of Technology
Faculty of Information Technology 1



CNTT1
Học viện Công nghệ Bưu chính Viễn thông

August 15, 2022

Các khái niệm liên quan đến tiến trình

Tiến trình là gì?

Trạng thái của tiến trình

Thông tin mô tả tiến trình

Bảng và danh sách tiến trình

Các thao tác với tiến trình

Luồng - Thread

Khái niệm

Tài nguyên của tiến trình và luồng

Ưu điểm của mô hình đa luồng

Luồng mức nhân và mức người dùng

Điều độ tiến trình

Khái niệm điều độ

1. Các khái niệm liên quan đến tiến trình
2. Luồng - Thread
3. Điều độ tiến trình
4. Đồng bộ hóa các tiến trình đồng thời
5. Tình trạng bế tắc và đói

Các khái niệm liên quan đến tiến trình

Tiến trình là gì?

Tiến trình là một chương trình đang trong quá trình thực hiện.

Chương trình	Tiến trình
Thực thể tĩnh	Thực thể động
Không sở hữu tài nguyên cụ thể	Được cấp một số tài nguyên để chứa tiến trình và thực hiện lệnh

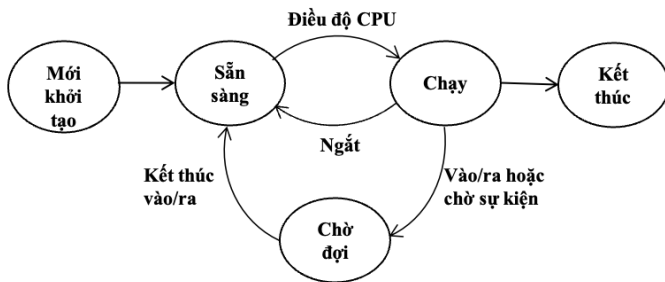
► Tiến trình sinh ra khi chương trình được tải vào bộ nhớ để:

- Thực hiện tiến trình người dùng
- Thực hiện tiến trình hệ thống

Process Name	% CPU	CPU Time	Threads	Idle Wake Ups	Kind	% GPU	GPU Time	PID	User
WindowServer	21,2	37:43:17,44	25	140	Apple	19,9	4:08:25,06	362	_windowserver
kernel_task	11,7	23:43:41,58	663	1340	Apple	0,0	0,00	0	root
♥ Safari Graphics and Media	6,0	2:17:58,10	18	164	Apple	0,0	1:18,67	45904	dinhxuantruon
launchd	4,5	3:06:10,31	3	1	Apple	0,0	0,00	1	root
coreaudiod	2,5	6:22:48,77	12	103	Apple	0,0	0,00	399	_coreaudiod
screencapture	2,0	0,51	3	0	Apple	0,0	0,00	98518	dinhxuantruon
♥ https://www.youtube.com	1,8	4:48:39	9	24	Apple	0,0	0,00	87145	dinhxuantruon
♥ Activity Monitor	1,7	5,48	5	1	Apple	0,0	0,00	98516	dinhxuantruon
Microsoft Word	1,0	7:37:22	38	27	Apple	0,0	1,48	92547	dinhxuantruon
♥ https://www.iclick22.scrs.in	0,9	55,46	4	24	Apple	0,0	0,00	97875	dinhxuantruon
loginwindow	0,9	2:13:53,12	14	726	Apple	0,0	1,27	365	dinhxuantruon
OneDrive	0,8	3:05:29,67	17	30	Intel	0,0	0,00	10545	dinhxuantruon
sysmond	0,8	1:04:01,33	3	0	Apple	0,0	0,00	823	root

- ▶ Khi thực hiện, tiến trình thay đổi trạng thái
- ▶ Trạng thái của tiến trình là một phần trong hoạt động hiện tại của tiến trình bao gồm 5 trạng thái:
 - **Mới khởi tạo (New):** tiến trình đang được tạo ra
 - **Sẵn sàng (Ready):** tiến trình chờ được cấp CPU để thực hiện lệnh
 - **Chạy (Running):** lệnh của tiến trình được CPU thực hiện
 - **Chờ đợi (Waiting):** tiến trình chờ đợi một sự kiện nào xảy ra (blocked)
 - **Kết thúc (Terminated):** tiến trình đã kết thúc việc thực hiện nhưng vẫn chưa bị xóa

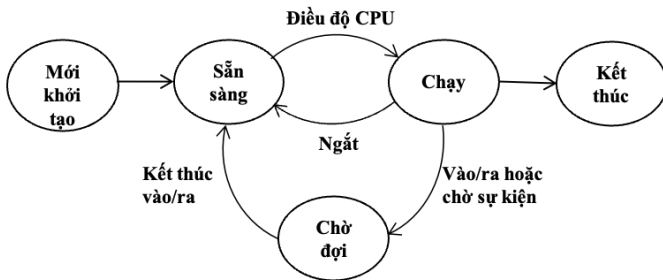
Chuyển đổi giữa các trạng :



- ▶ **Khởi tạo -> Sẵn sàng**: tiến trình khởi tạo xong và đã được tải vào bộ nhớ, chỉ chờ được cấp CPU để chạy.
- ▶ **Sẵn sàng -> Chạy**: do kết quả điều độ CPU của OS, tiến trình được OS cấp phát CPU và chuyển sang trạng thái chạy

Các khái niệm liên quan đến tiến trình (cont.)

Trạng thái của tiến trình



- ▶ **Chạy -> Sẵn sàng:** OS cấp phát CPU cho tiến trình khác, do kết quả điều độ/do ngắt xảy ra, tiến trình hiện thời chuyển sang trạng thái sẵn sàng và chờ được cấp CPU để chạy tiếp.
- ▶ **Chạy -> Chờ đợi:** Tiến trình chỉ chạy khi có 1 sự kiện nào đó xảy ra, chuyển sang trạng thái chờ được phân phối CPU để chạy tiếp.
- ▶ **Chạy -> Kết thúc:** khi tiến trình đã thực hiện

Thông tin của tiến trình được lưu trong một cấu trúc dữ liệu gọi là khối quản lý tiến trình - PCB (Process Control Block)

► Các thông tin chính trong PCB:

- Số định danh của tiến trình (PID)
- Trạng thái tiến trình: một trong năm trạng thái
- Nội dung một số thanh ghi CPU:
 - Thanh ghi con trỏ lệnh: trỏ tới lệnh tiếp theo
 - Thanh ghi con trỏ ngăn xếp : lưu tham số / tình trạng hàm khi thực hiện lời gọi hàm / thủ tục của chương trình
 - Các thanh ghi điều kiện và trạng thái
 - Các thanh ghi đa năng
- Thông tin quản lý bộ nhớ
- Thông tin tài nguyên có thể sử dụng

process state
process number
program counter
registers
memory limits
list of open files
...

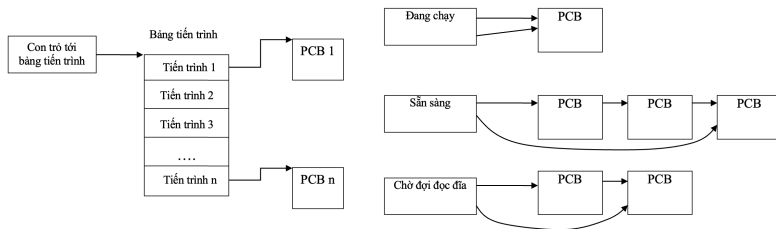
Vai trò của Khối quản lý tiến trình PCB:

- ▶ *Chứa thông tin* để OS khôi phục lại nội dung các thanh ghi, cho phép tiến trình thực hiện lại từ trạng thái trước lúc dừng
- ▶ *Thông tin phục vụ điều độ tiến trình*: mức độ ưu tiên của tiến trình, vị trí trong hàng đợi, lượng tài nguyên đang sở hữu.
- ▶ *Thông tin về bộ nhớ của tiến trình*: đang nằm ở đâu trong bộ nhớ
- ▶ *Danh sách các tài nguyên khác*: các file đang mở, thiết bị vào ra mà tiến trình sử dụng
- ▶ *Thông tin thống kê phục vụ quản lý*: thời gian sử dụng CPU, giới hạn thời gian

Các khái niệm liên quan đến tiến trình

Bảng và danh sách tiến trình

- ▶ OS lưu và xác định vị trí các PCB bằng cách sử dụng bảng tiến trình chứa con trỏ tới PCB của toàn bộ các tiến trình trong hệ thống
- ▶ Ngoài ra, PCB của các tiến trình *cùng trạng thái* hoặc *cùng chờ 1 tài nguyên* nào đó được liên kết thành 1 danh sách, mỗi danh sách gồm một số tiến trình cùng trạng thái.



- ▶ Tạo mới tiến trình
- ▶ Kết thúc tiến trình
- ▶ Chuyển đổi giữa các tiến trình
- ▶ Điều độ tiến trình
- ▶ Đồng bộ hoá tiến trình
- ▶ Đảm bảo liên lạc giữa các tiến trình

Tạo mới tiến trình:

- ▶ Để tạo ra một tiến trình mới, OS thực hiện một số bước như sau:
 - *Gán số định danh* cho tiến trình được tạo mới và tạo một ô trong bảng tiến trình
 - *Tạo không gian nhớ* cho tiến trình và PCB
 - *Khởi tạo PCB*: HĐH gán giá trị các thành phần của PCB
 - *Liên kết PCB của tiến trình* vào các danh sách quản lý, ví dụ vào ds tiến trình đang chạy.

Kết thúc tiến trình:

- ▶ Kết thúc bình thường: yêu cầu HDH kết thúc mình bằng cách gọi lời gọi hệ thống `exit()`.
- ▶ Tiến trình bị kết thúc trong các trường hợp sau:
 - Bị tiến trình cha kết thúc
 - Do các lỗi
 - Yêu cầu nhiều bộ nhớ hơn so với số lượng hệ thống có thể cung cấp
 - Thực hiện lâu hơn thời gian giới hạn
 - Do quản trị hệ thống hoặc hệ điều hành kết thúc

Chuyển đổi giữa các tiến trình - Chuyển đổi ngữ cảnh:

- ▶ Trong quá trình thực hiện, CPU có thể được chuyển từ tiến trình hiện thời sang thực hiện tiến trình khác.
- ▶ Thông tin về tiến trình hiện thời trong PCB được gọi là *ngữ cảnh* (context) của tiến trình.
- ▶ Việc chuyển đổi tiến trình xảy ra khi:
 - Có ngắt: ngắt do đồng hồ/ngắt vào/ra
 - Tiến trình gọi lời gọi hệ thống
- ▶ Khi chuyển sang thực hiện tiến trình khác, ngữ cảnh lưu vào PCB
- ▶ Khi được cấp phát CPU thực hiện trở lại, ngữ cảnh được khôi phục từ PCB vào các thanh ghi và bảng tương ứng.

Chuyển đổi giữa các tiến trình - Chuyển đổi ngữ cảnh:

Ngữ cảnh phải bao gồm những thông tin nào?

- ▶ TH đơn giản: Hệ thống chuyển sang thực hiện vào/ra rồi quay lại thực hiện tiếp tiến trình:
 - Ngữ cảnh gồm thông tin có thể bị hàm xử lý ngắt thay đổi
 - Nội dung thanh ghi, trạng thái CPU
- ▶ TH phức tạp: Sau khi ngắt, hệ thống thực hiện tiến trình khác
 - Thay đổi trạng thái tiến trình
 - Cập nhật thông tin thống kê trong PCB
 - Chuyển PCB của tiến trình vào danh sách ứng với trạng thái mới
 - Cập nhật PCB của tiến trình mới được chọn
 - Cập nhật nội dung thanh ghi và trạng thái CPU
- ▶ Chuyển đổi tiến trình đòi hỏi thời gian

- ▶ Tiến trình được xem xét từ 2 khía cạnh:
 - Tiến trình là 1 đơn vị sở hữu tài nguyên
 - Tiến trình là 1 đơn vị thực hiện công việc tính toán xử lý
- ▶ Các HĐH trước đây: mỗi tiến trình chỉ tương ứng với 1 đơn vị xử lý duy nhất, điều này có nghĩa *tiến trình không thể thực hiện nhiều hơn một công việc cùng một lúc*
- ▶ Các HĐH hiện đại: cho phép tách riêng vai trò thực hiện lệnh của tiến trình

Mỗi đơn vị thực hiện lệnh của tiến trình, tức là 1 chuỗi lệnh được cấp phát CPU để thực hiện độc lập được gọi là **một luồng thực hiện**

- ▶ HĐH hiện nay thường hỗ trợ đa luồng (multithreading):
 - Tiến trình có thể có nhiều luồng
 - Cho phép nhiều chuỗi lệnh, thực hiện nhiều nhiệm vụ cùng một lúc



tiến trình gồm một luồng



tiến trình gồm nhiều luồng

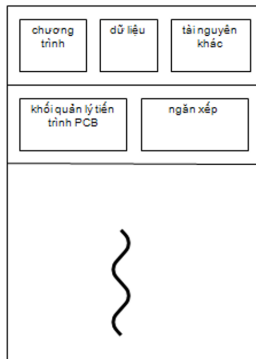
} = chuỗi lệnh

- ▶ Trong hệ thống cho phép đa luồng, tiến trình vẫn là 1 đơn vị để HDH phân phối tài nguyên
- ▶ Mỗi tiến trình sở hữu chung một số tài nguyên:
 - Không gian nhớ của tiến trình (logic): chứa chương trình (các lệnh), phần dữ liệu của tiến trình.
 - Các tài nguyên khác: các file đang mở, thiết bị hoặc cổng vào/ra

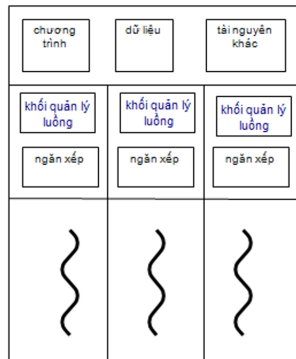
Tiến trình đơn luồng và tiến trình đa luồng.

Mô hình đơn luồng	Mô hình đa luồng
Tiến trình có khối quản lý PCB chứa đầy đủ thông tin trạng thái tiến trình, giá trị thanh ghi	Mỗi luồng cần có khả năng quản lý con trỏ lệnh, nội dung thanh ghi
Ngăn xếp chứa tham số, thủ tục, hàm, chương trình con	Luồng cũng có trạng thái riêng chứa trong khối quản lý luồng Luồng có ngăn xếp riêng
Khi tiến hành thực hiện, luồng sẽ làm chủ nội dung các thanh ghi và con trỏ lệnh	Tất cả các luồng của tiến trình chia sẻ không gian nhớ và tài nguyên

Tiến trình đơn luồng và tiến trình đa luồng.



Tiến trình đơn luồng



Tiến trình nhiều luồng

Ưu điểm của 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

- ▶ Có thể tạo và quản lý luồng ở 2 mức:
 - Mức người dùng
 - Mức nhân
- ▶ Luồng mức người dùng: được tạo ra và quản lý không có sự hỗ trợ của hệ điều hành
- ▶ Luồng mức nhân: được tạo ra và quản lý bởi hệ điều hành

Luồng ở mức người dùng:

- ▶ Do trình ứng dụng tạo ra và quản lý
- ▶ Việc phân phối CPU được thực hiện cho cả tiến trình
- ▶ OS vẫn coi tiến trình như một đơn vị duy nhất với một trạng thái duy nhất
- ▶ Ưu điểm:
 - Tiết kiệm thời gian do việc chuyển đổi luồng không đòi hỏi chuyển sang chế độ nhân
 - Trình ứng dụng có thể điều độ riêng, không phụ thuộc vào cách điều độ của OS
 - Có thể sử dụng khi OS không hỗ trợ đa luồng
- ▶ Nhược điểm
 - Không tận dụng ưu điểm về tính đáp ứng của mô hình đa luồng
 - Không cho phép tận dụng kiến trúc nhiều CPU

Luồng ở mức nhân:

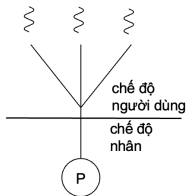
- ▶ Luồng mức nhân được HĐH tạo ra và quản lý
- ▶ OS cung cấp giao diện lập trình
- ▶ Cung cấp các lời gọi hệ thống mà trình ứng dụng có thể yêu cầu tạo, xóa luồng và thay đổi tham số liên quan quản lý dòng
- ▶ Ưu điểm:
 - Tăng tính đáp ứng và khả năng thực hiện đồng thời của các luồng trong cùng tiến trình
- ▶ Nhược điểm
 - Tốc độ chậm do tạo và chuyển đổi luồng thực hiện trong chế độ nhân
- ▶ HĐH Windows và Linux hỗ trợ luồng mức nhân

Kết hợp luồng mức nhân và mức người dùng:

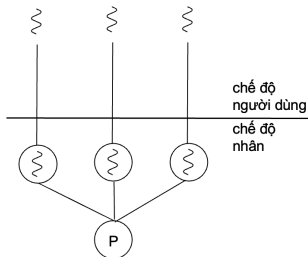
- ▶ Có thể sử dụng dòng mức người dùng và dòng mức nhân
- ▶ Theo các tổ chức này:
 - Luồng mức người dùng được tạo ra trong chế độ người dùng nhờ thư viện của trình ứng dụng
 - Luồng mức người dùng được ánh xạ lên số lượng tương ứng hoặc ít hơn các luồng mức nhân
 - Số lượng dòng mức nhân phụ thuộc vào hệ thống cụ thể, chẳng hạn hệ thống nhiều CPU sẽ có nhiều dòng mức nhân hơn.

Luồng - Thread (cont.)

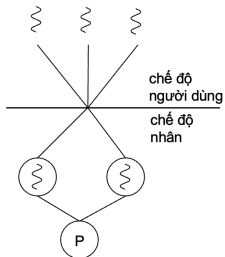
Luồng mức nhân và mức người dùng



a) Mô hình mức người dùng



b) Mô hình mức nhân



c) Mô hình kết hợp

~ LUỒNG mức người dùng

~ LUỒNG mức nhân

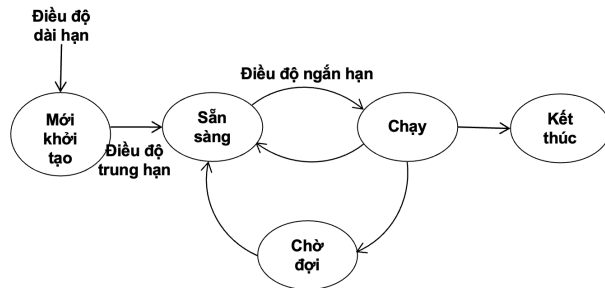
P tiến trình

- ▶ *Trong hệ thống cho phép đa chương trình, nhiều tiến trình có thể tồn tại và thực hiện cùng lúc.*
- ▶ *Kỹ thuật đa chương trình có nhiều ưu điểm do cho phép sử dụng CPU hiệu quả, đáp ứng nhu cầu tính toán của người dùng*
- ▶ *Tuy nhiên, đặt ra nhiều vấn đề phức tạp hơn đối với HĐH*

Điều độ (scheduling) hay *lập lịch* là quyết định tiến trình nào được sử dụng tài nguyên phần cứng khi nào, trong thời gian bao lâu.

- ▶ Tập trung vào vấn đề điều độ đối với CPU là quyết định thứ tự và thời gian sử dụng CPU
- ▶ Điều độ tiến trình và điều độ luồng:
 - Hệ thống trước kia: tiến trình là đơn vị thực hiện chính, điều độ thực hiện với tiến trình
 - Hệ thống hỗ trợ luồng: luồng mức nhân là đơn vị thực hiện được HDH cấp CPU chứ không phải tiến trình.
 - Sử dụng thuật ngữ điều độ tiến trình rộng rãi tương đương điều độ luồng

Điều độ dài hạn và ngắn hạn



► Điều độ dài hạn:

- Thực hiện khi mới tạo ra tiến trình
- HDH quyết định tiến trình có được thêm vào danh sách đang hoạt động?

- Nếu được chấp nhận, hệ thống sẽ có thêm tiến trình mới. Ngược lại, chờ tới thời điểm khác để tạo và thực hiện
 - Ảnh hưởng tới mức độ đa chương trình
- Điều độ trung hạn:
- Quyết định tiến trình có được cấp MEM để thực hiện?
- Điều độ ngắn hạn:
- Quyết định tiến trình nào được cấp CPU để thực hiện
 - Thực hiện với tiến trình ở trạng thái sẵn sàng

Điều độ có phân phối lại và không phân phối lại:

- ▶ Điều độ có phân phối lại (preemptive):
 - HDH có thể sử dụng cơ chế ngắt để thu hồi CPU của một tiến trình đang trong trạng thái chạy
- ▶ Điều độ không phân phối lại (nonpreemptive):
 - Tiến trình đang ở trạng thái chạy sẽ được sử dụng CPU cho đến khi xảy ra một trong các tình huống sau:
 - ▶ Tiến trình kết thúc
 - ▶ Tiến trình phải chuyển sang trạng thái chờ đợi do thực hiện I/O
 - Điều độ hợp tác: do chỉ thực hiện được khi tiến trình hợp tác và nhường CPU
 - Nếu tiến trình không hợp tác, dùng CPU vô hạn nên các tiến trình khác không được cấp CPU (Windows 96, NT)

Ưu điểm của điều độ có phân phối lại:

- ▶ So với điều độ không phân phối lại, điều độ có phân phối lại có nhiều ưu điểm hơn.
- ▶ HĐH chủ động hơn, không phụ thuộc vào hoạt động của tiến trình
- ▶ Đảm bảo chia sẻ thời gian thực sự
- ▶ Đòi hỏi phần cứng có bộ định thời gian và một số hỗ trợ khác
- ▶ Vấn đề quản lý tiến trình phức tạp hơn

Một số tiêu chí thường được sử dụng:

- ▶ Lượng tiến trình được thực hiện xong:
 - Số lượng tiến trình thực hiện xong trong 1 đơn vị thời gian
 - Đo tính hiệu quả của hệ thống
- ▶ Hiệu suất sử dụng CPU
 - Cố gắng để CPU càng ít phải nghỉ càng tốt
- ▶ Thời gian vòng đời trung bình của tiến trình:
 - Từ lúc có yêu cầu tạo tiến trình đến khi kết thúc
- ▶ Thời gian chờ đợi:
 - Tổng thời gian tiến trình nằm trong trạng thái sẵn sàng và chờ cấp CPU
 - Ảnh hưởng trực tiếp của thuật toán điều độ tiến trình

Một số tiêu chí thường được sử dụng:

- ▶ Thời gian đáp ứng:
 - Đây là tiêu chí hướng tới người dùng và thường được sử dụng trong hệ thống tương tác trực tiếp.
- ▶ Tính dự đoán được:
 - Vòng đời, thời gian chờ đợi, thời gian đáp ứng phải ổn định, không phụ thuộc vào tải của hệ thống
- ▶ Tính công bằng:
 - Các tiến trình cùng độ ưu tiên phải được đối xử như nhau

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)

Thuật toán đến trước phục vụ trước (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:

Tiến trình		Thời gian	
P1		24	
P2		3	
P3		3	
0		24	27
24		3	3
P1		P2	P3

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

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

Thuật toán đến trước phục vụ trước (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
- ▶ Đặc điểm
 - Đơn giản, dễ thực hiện
 - Tiến trình ngắn phải chờ đợi như tiến trình dài (Nếu P1 thực hiện sau cùng ?)

Điều độ quay vòng (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
P1	P2	P3	P1	P2	P1	P1

Điều độ ưu tiên tiến trình ngắn nhất (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:

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

2	4	10
P3	P2	P1

Điều độ ưu tiên thời gian còn lại ngắn nhất (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

Ví dụ:

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

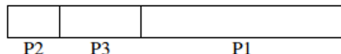
Điều độ có mức ưu tiên (Priority Scheduling)

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



Chương 4 Quản lý tiến trình

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

Chương 4 Quản lý tiến trình

- ▶ Đồng bộ hóa các tiến trình đồng thời
- ▶ Tình trạng bế tắc và đói

1. **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 :

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

- Điều độ theo mức ưu tiên không có phân phối lại
- Điều độ ưu tiên tiến trình ngắn nhất
- Điều độ ưu tiên thời gian còn lại ngắn nhất

Tính thời gian chờ đợi trung bình cho từng trường hợp