

## CHƯƠNG 2: QUẢN LÝ TIỀN TRÌNH

**Những kiến thức trọng tâm của chương gồm có:**

1. Hiểu các khái niệm về tiến trình
2. Các phương pháp giải quyết tới hạn
2. Hiểu cách lập thời biểu tiến trình
3. Biết các thao tác trên tiến trình
4. Các thuật toán lập lịch CPU cho tiến trình

### **1. Khái niệm về tiến trình**

Những hệ thống máy tính ban đầu cho phép chỉ một chương trình được thực thi tại một thời điểm. Chương trình này có toàn quyền điều khiển hệ thống và có truy xuất tới tất cả tài nguyên của hệ thống. Những hệ thống máy tính hiện nay cho phép nhiều chương trình được nạp vào bộ nhớ và được thực thi đồng hành. Sự phát triển này yêu cầu sự điều khiển mạnh mẽ hơn và phân chia nhiều hơn giữa các tiến trình. Yêu cầu này dẫn đến khái niệm tiến trình, một chương trình đang thực thi. Tiến trình là một đơn vị công việc trong một hệ điều hành chia thời hiện đại.

Một hệ điều hành phức tạp hơn được mong đợi nhiều hơn trong việc thực hiện các hành vi của người dùng. Mặc dù quan tâm chủ yếu của hệ điều hành là thực thi chương trình người dùng, nhưng nó cũng quan tâm đến các tác vụ khác nhau bên ngoài nhân. Do đó, một hệ thống chứa tập hợp các tiến trình: tiến trình hệ điều hành thực thi mã hệ thống, tiến trình người dùng thực thi mã người dùng. Tất cả tiến trình này có tiềm năng thực thi đồng hành, với một CPU (hay nhiều CPU) được đa hợp giữa chúng. Bằng cách chuyển đổi CPU giữa các tiến trình, hệ điều hành có thể làm cho máy tính hoạt động với năng suất cao hơn.

Một vấn đề cần thảo luận là cái gì được gọi trong tất cả hoạt động của CPU? Một hệ thống bó thực thi công việc, trái lại một hệ thống chia thời thực thi chương trình người dùng hay tác vụ. Thậm chí trên hệ thống đơn người dùng như Microsoft Windows và Macintosh OS, một người dùng có thể chạy nhiều chương trình tại một thời điểm: bộ xử lý văn bản, trình duyệt web, e-mail. Thậm chí nếu người dùng có thể thực thi chỉ một tiến trình tại một thời điểm, thì một hệ điều hành cần hỗ trợ những hoạt động được lập trình bên trong, như quản lý bộ nhớ. Trong nhiều khía cạnh, tất cả hoạt động là tương tự vì thế chúng ta gọi tất cả chúng là tiến trình.

### **1.1. Tiến trình là gì?**

Một tiến trình là một chương trình đang thực thi. Một tiến trình không chỉ là mã chương trình, nó còn bao gồm hoạt động hiện hành như được hiện diện bởi giá trị của bộ đếm chương trình và nội dung các thanh ghi của bộ xử lý. Ngoài ra, một tiến trình thường chứa ngăn xếp tiến trình, chứa dữ liệu tạm thời (như các tham số phương thức, các địa chỉ trả về, các biến cục bộ) và phần dữ liệu chứa các biến toàn cục.

Chúng ta nhấn mạnh rằng, một chương trình không phải là một tiến trình; một chương trình là một thực thể thụ động, như nội dung của các tập tin được

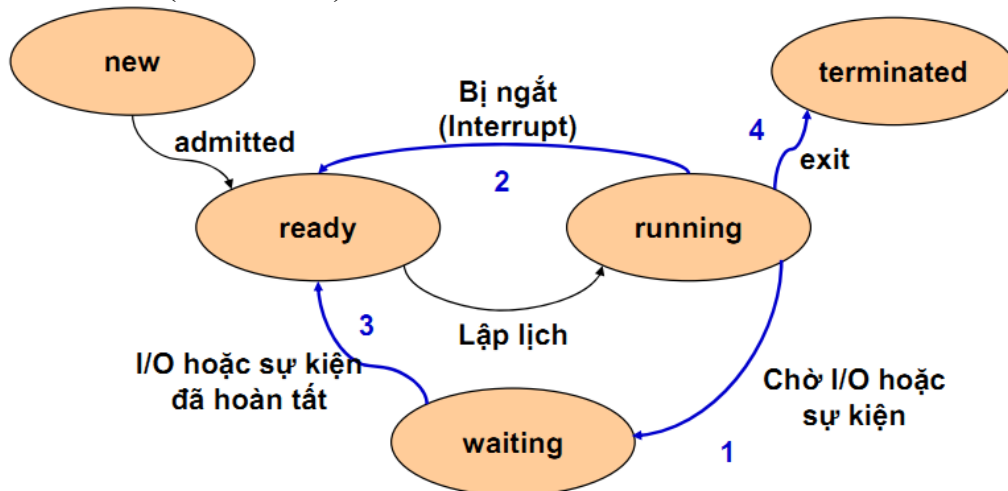
lưu trên đĩa, trái lại một tiến trình là một thực thể chủ động, với một bộ đếm chương trình xác định chỉ thị lệnh tiếp theo sẽ thực thi và tập hợp tài nguyên có liên quan.

Mặc dù hai tiến trình có thể được liên kết với cùng chương trình nhưng chúng được chứa hai thứ tự thực thi riêng rẽ. Thí dụ, nhiều người dùng có thể đang chạy các bản sao của chương trình gửi nhận thư, hay cùng người dùng có thể nạp lên nhiều bản sao của một chương trình soạn thảo văn bản. Mỗi bản sao của chúng là một tiến trình riêng và mặc dù các phần văn bản là giống nhau, các phần dữ liệu khác nhau. Ngoài ra, một tiến trình có thể tạo ra nhiều tiến trình khi nó thực thi.

## 1.2 Trạng thái tiến trình

Khi một tiến trình thực thi, nó thay đổi trạng thái. Trạng thái của tiến trình được định nghĩa bởi các hoạt động hiện hành của tiến trình đó. Mỗi tiến trình có thể ở một trong những trạng thái sau:

- Mới (new): tiến trình đang được tạo ra
- Đang chạy (running): các chỉ thị đang được thực thi
- Chờ (waiting): tiến trình đang chờ sự kiện xảy ra (như hoàn thành việc nhập/xuất hay nhận tín hiệu)
- Sẵn sàng (ready): tiến trình đang chờ được gán tới một bộ xử lý.
- Kết thúc (terminated): tiến trình hoàn thành việc thực thi



Các tên trạng thái này là bất kỳ, và chúng khác nhau ở các hệ điều hành khác nhau. Tuy nhiên, các trạng thái mà chúng hiện diện được tìm thấy trên tất cả hệ thống. Các hệ điều hành xác định mô tả trạng thái tiến trình. Chỉ một tiến trình có thể đang chạy tức thì trên bất kỳ bộ xử lý nào mặc dù nhiều tiến trình có thể ở trạng thái sẵn sàng và chờ.

## 1.3 Khối điều khiển tiến trình

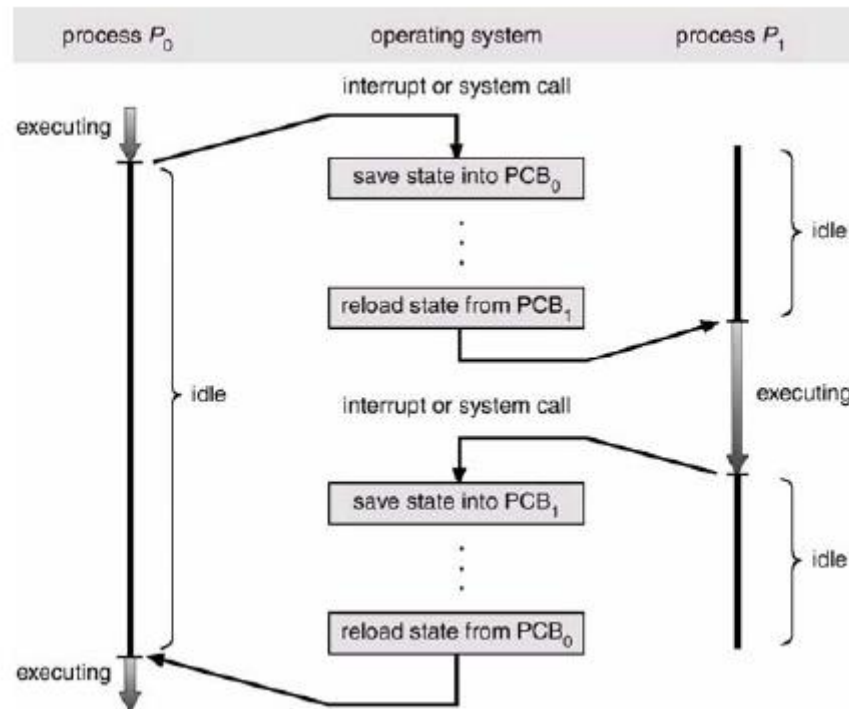
Mỗi tiến trình được hiện diện trong hệ điều hành bởi một **khối điều khiển tiến trình** (Process Control Block-PCB) – cũng được gọi khối điều khiển tác vụ. Một PCB được hiển thị trong hình. Nó chứa nhiều phần thông tin được gắn liền với một tiến trình xác định, gồm:

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

**Hình-Khối điều khiển tiến trình**

- **Trạng thái tiến trình** (process state): trạng thái có thể là mới, sẵn sàng, đang chạy, chờ đợi, kết thúc, ...
- **Bộ đếm chương trình** (program counter): bộ đếm hiển thị địa chỉ của chỉ thị kế tiếp được thực thi cho tiến trình này.
- **Các thanh ghi** (registers) CPU: các thanh ghi khác nhau về số lượng và loại, phụ thuộc vào kiến trúc máy tính. Chúng gồm các bộ tổng (accumulators), các thanh ghi chỉ mục, các con trỏ ngăn xếp, và các thanh ghi đa năng (general-purpose registers), cùng với thông tin mã điều kiện (condition-code information). Cùng với bộ đếm chương trình, thông tin trạng thái này phải được lưu khi một ngắt xảy ra, cho phép tiến trình được tiếp tục một cách phù hợp sau đó.
- **Thông tin lập thời biểu CPU** (CPU-scheduling information): thông tin gồm độ ưu tiên của tiến trình, các con trỏ chỉ tới các hàng đợi lập thời biểu, và bất kỳ tham số lập thời biểu khác.
- **Thông tin quản lý bộ nhớ** (Memory-management information): thông tin này có thể gồm những thông tin như giá trị của các thanh ghi nền và thanh ghi giới hạn, các bảng trang hay các bảng phân đoạn, phụ thuộc hệ thống bộ nhớ được dùng bởi hệ điều hành.
- **Thông tin tính toán** (accounting information): thông tin này gồm lượng CPU và thời gian thực được dùng, công việc hay số tiến trình, ...
- **Thông tin trạng thái nhập/xuất** (I/O status information): thông tin này gồm danh sách của thiết bị nhập/xuất được cấp phát tiến trình này, một danh sách các tập tin đang mở, ..

PCB đơn giản phục vụ như kho chứa cho bất cứ thông tin khác nhau từ tiến trình này tới tiến trình khác.



**Hình -Lưu đồ hiển thị việc chuyển CPU từ tiến trình này tới tiến trình khác**

#### 1.4 Luồng

Mô hình tiến trình vừa được thảo luận ngụ ý rằng một tiến trình là một chương trình thực hiện một luồng đơn thực thi. Thí dụ, nếu một tiến trình đang chạy một chương trình xử lý văn bản, một luồng đơn của chỉ thị đang được thực thi. Đây là một luồng điều khiển đơn cho phép tiến trình thực thi chỉ một tác vụ tại một thời điểm. Thí dụ, người dùng không thể cùng lúc nhập các ký tự và chạy bộ kiểm tra chính tả trong cùng một tiến trình. Nhiều hệ điều hành hiện đại mở rộng khái niệm tiến trình để cho phép một tiến trình có nhiều luồng thực thi. Do đó, chúng cho phép thực hiện nhiều hơn một tác vụ tại một thời điểm.

#### 2. Các thao tác của tiến trình

Hệ điều hành cung cấp các thao tác chủ yếu sau đây trên một tiến trình:

- Tạo lập tiến trình (create)
- Kết thúc tiến trình (destroy)
- Tạm dừng tiến trình (suspend)
- Tái kích hoạt tiến trình (resume)
- Thay đổi độ ưu tiên tiến trình

##### a) Tạo lập tiến trình

Một tiến trình được tạo lập khi:

- Người sử dụng chạy một chương trình.
- Hệ điều hành thực hiện một số dịch vụ.
- Tiến trình cha sinh tiến trình con

- Một người truy nhập hệ thống.

Trong quá trình xử lý, một tiến trình có thể tạo lập nhiều tiến trình mới bằng cách sử dụng một lời gọi hệ thống tương ứng. Tiến trình gọi lời gọi hệ thống để tạo tiến trình mới sẽ được gọi là tiến trình cha, tiến trình được tạo gọi là tiến trình con.

Mỗi tiến trình con đến lượt nó lại có thể tạo các tiến trình mới... quá trình này tiếp tục sẽ tạo ra một cây tiến trình.

Các công việc hệ điều hành cần thực hiện khi tạo lập tiến trình bao gồm:

- Định danh cho tiến trình mới phát sinh
- Đưa tiến trình vào danh sách quản lý của hệ thống
- Xác định độ ưu tiên cho tiến trình
- Tạo PCB cho tiến trình
- Cấp phát các tài nguyên ban đầu cho tiến trình

Khi một tiến trình tạo lập một tiến trình con, tiến trình con có thể sẽ được hệ điều hành trực tiếp cấp phát tài nguyên hoặc được tiến trình cha cho thừa hưởng một số tài nguyên ban đầu.

Khi một tiến trình tạo tiến trình mới, tiến trình ban đầu có thể xử lý theo một trong hai khả năng sau:

Tiến trình cha tiếp tục xử lý đồng hành với tiến trình con. Ví dụ: UNIX

Tiến trình cha chờ đến khi một tiến trình con nào đó, hoặc tất cả các tiến trình con kết thúc xử lý. Ví dụ MSDOS

Các hệ điều hành khác nhau có thể chọn lựa các cài đặt khác nhau để thực hiện thao tác tạo lập một tiến trình.

#### *b) Kết thúc tiến trình*

Một tiến trình kết thúc khi:

- Tiến trình hoàn tất công việc.
- Tiến trình kết thúc khi vượt quá thời hạn
- Tiến trình kết thúc khi sử dụng quá tài nguyên quy định
- Tiến trình kết thúc khi bộ nhớ không đủ.
- Tiến trình vi phạm một số quy định
- Tiến trình mắc một số lỗi về phép toán
- Thiết bị ngoại vi bị lỗi
- Khi các lệnh bị sai
- Khi có quyền ưu tiên
- Dữ liệu sai
- HĐH dừng một số tiến trình

Một tiến trình kết thúc xử lý khi nó hoàn tất chỉ thị cuối cùng và sử dụng một lời gọi hệ thống để yêu cầu hệ điều hành hủy bỏ nó. Đôi khi một tiến trình có thể kết thúc xử lý của một tiến trình khác bằng một lời gọi hệ thống tương ứng. Khi một tiến trình kết thúc, hệ điều hành thực hiện các công việc:

- Thu hồi các tài nguyên hệ thống đã cấp phát cho tiến trình-Hủy tiến trình khỏi tất cả các danh sách quản lý của hệ thống
- Hủy bỏ PCB của tiến trình

Hầu hết các hệ điều hành không cho phép các tiến trình con tiếp tục tồn tại nếu tiến trình cha đã kết thúc. Trong những hệ thống như thế, hệ điều hành sẽ tự động phát sinh một loạt các thao tác kết thúc tiến trình con.

### 2.1. Cấp phát tài nguyên cho tiến trình

Khi có nhiều người sử dụng đồng thời làm việc trong hệ thống, hệ điều hành cần phải cấp phát các tài nguyên theo yêu cầu cho mỗi người sử dụng. Do tài nguyên hệ thống thường rất giới hạn và có khi không thể chia sẻ, nên hiếm khi tất cả các yêu cầu tài nguyên đồng thời đều được thỏa mãn. Vì thế cần phải nghiên cứu một phương pháp để chia sẻ một số tài nguyên hữu hạn giữa nhiều tiến trình người dùng đồng thời.

Hệ điều hành quản lý nhiều loại tài nguyên khác nhau (CPU, bộ nhớ chính, các thiết bị ngoại vi ...), với mỗi loại cần có một cơ chế cấp phát và các chiến lược cấp phát hiệu quả. Mỗi tài nguyên được biểu diễn thông qua một cấu trúc dữ liệu, khác nhau về chi tiết cho từng loại tài nguyên, nhưng cơ bản chứa đựng các thông tin sau:

- Định danh tài nguyên
- Trạng thái tài nguyên: đây là các thông tin mô tả chi tiết trạng thái tài nguyên: phần nào của tài nguyên đã cấp phát cho tiến trình, phần nào còn có thể sử dụng ?
- Hàng đợi trên một tài nguyên: danh sách các tiến trình đang chờ được cấp phát tài nguyên tương ứng.
- Bộ cấp phát: là đoạn code đảm nhiệm việc cấp phát một tài nguyên đặc thù.

Một số tài nguyên đòi hỏi các giải thuật đặc biệt (như CPU, bộ nhớ chính, hệ thống tập tin), trong khi những tài nguyên khác (như các thiết bị nhập/xuất) có thể cần các giải thuật cấp phát và giải phóng tổng quát hơn. Khối quản lý tài nguyên như sau:

Định danh tài nguyên	Rid
Trạng thái tài nguyên	Danh sách các phần có thể sử dụng
Hàng đợi	Danh sách các tiến trình đang đợi tài nguyên
Bộ cấp phát	Con trỏ đến đoạn code cấp phát tài nguyên

Các mục tiêu của kỹ thuật cấp phát:

- Bảo đảm một số lượng hợp lệ các tiến trình truy xuất đồng thời đến các tài nguyên không chia sẻ được.
- Cấp phát tài nguyên cho tiến trình có yêu cầu trong một khoảng thời gian trì hoãn có thể chấp nhận được.
- Tối ưu hóa sự sử dụng tài nguyên.

Để có thể thỏa mãn các mục tiêu kể trên, cần phải giải quyết các vấn đề nảy sinh khi có nhiều tiến trình đồng thời yêu cầu một tài nguyên không thể chia sẻ.

### 2.2. Điều phối tiến trình

Trong môi trường đa chương, có thể xảy ra tình huống nhiều tiến trình đồng thời sẵn sàng để xử lý. Mục tiêu của các hệ phân chia thời gian (time-sharing) là chuyển đổi CPU qua lại giữa các tiến trình một cách thường xuyên để nhiều người sử dụng có thể tương tác cùng lúc với từng chương trình trong quá trình xử lý.

Để thực hiện được mục tiêu này, hệ điều hành phải lựa chọn tiến trình được xử lý tiếp theo. Bộ điều phối sẽ sử dụng một giải thuật điều phối thích hợp để thực hiện nhiệm vụ này. Một thành phần khác của hệ điều hành cũng tiềm ẩn trong công tác điều phối là bộ phân phối (dispatcher). Bộ phân phối sẽ chịu trách nhiệm chuyển đổi ngữ cảnh và trao CPU cho tiến trình được chọn bởi bộ điều phối để xử lý.

#### 2.2.1. Mục tiêu điều phối

Bộ điều phối không cung cấp cơ chế, mà đưa ra các quyết định. Các hệ điều hành xây dựng nhiều chiến lược khác nhau để thực hiện việc điều phối, nhưng tựu

chung cần đạt được các mục tiêu sau:

a) Sự công bằng (Fairness):

Các tiến trình chia sẻ CPU một cách công bằng, không có tiến trình nào phải chờ đợi vô hạn để được cấp phát CPU

b) Tính hiệu quả (Efficiency): Hệ thống phải tận dụng được CPU nhiều nhất có thể. Trong hệ thống thực, nó nên nằm trong khoảng từ 40% (cho hệ thống được nạp tải nhẹ) tới 90% (cho hệ thống được nạp tải nặng).

c) Thời gian đáp ứng hợp lý (Response time):

Cực tiểu hóa thời gian hồi đáp cho các tương tác của người sử dụng

d) Thời gian lưu lại trong hệ thống (Turnaround Time): Cực tiểu hóa thời gian hoàn tất các tác vụ xử lý heo lô.

e) Thông lượng tối đa (Throughput):

Cực đại hóa số công việc được xử lý trong một đơn vị thời gian.

Tuy nhiên thường không thể thỏa mãn tất cả các mục tiêu kể trên vì bản thân chúng có sự mâu thuẫn với nhau mà chỉ có thể dung hòa chúng ở mức độ nào đó.

#### 2.2.2 Điều phối không độc quyền và điều phối độc quyền (preemptive/nopreemptive)

Thuật toán điều phối cần xem xét và quyết định thời điểm chuyển đổi CPU giữa các tiến trình. Hệ điều hành có thể thực hiện cơ chế điều phối theo nguyên lý độc quyền hoặc không độc quyền.

Điều phối độc quyền: Nguyên lý điều phối độc quyền cho phép một tiến trình khi nhận được CPU sẽ có quyền độc chiếm CPU đến khi hoàn tất xử lý hoặc tự nguyện giải phóng CPU. Khi đó quyết định điều phối CPU sẽ xảy ra trong các tình huống sau:

- Khi tiến trình chuyển từ trạng thái đang xử lý (running) sang trạng thái bị khóa blocked (ví dụ chờ một thao tác nhập xuất hay chờ một tiến trình con kết thúc...).

- Khi tiến trình kết thúc.

Các giải thuật độc quyền thường đơn giản và dễ cài đặt. Tuy nhiên chúng thường không thích hợp với các hệ thống tổng quát nhiều người dùng, vì nếu cho phép một tiến trình có quyền xử lý bao lâu tùy ý, có nghĩa là tiến trình này có thể giữ CPU một thời gian không xác định, có thể ngăn cản những tiến trình còn lại trong hệ thống có một cơ hội để xử lý.

Điều phối không độc quyền: Ngược với nguyên lý độc quyền, điều phối theo nguyên lý không độc quyền cho phép tạm dừng hoạt động của một tiến trình đang sẵn sàng xử lý. Khi một tiến trình nhận được CPU, nó vẫn được sử dụng CPU đến khi hoàn tất hoặc tự nguyện giải phóng CPU, nhưng một tiến trình khác có độ ưu tiên có thể dành quyền sử dụng CPU của tiến trình ban đầu. Như vậy là tiến trình có thể bị tạm dừng hoạt động bất cứ lúc nào mà không được báo trước, để tiến trình khác xử lý.

Các quyết định điều phối xảy ra khi:

- Khi tiến trình chuyển từ trạng thái đang xử lý (running) sang trạng thái bị khóa blocked ( ví dụ chờ một thao tác nhập xuất hay chờ một tiến trình con kết thúc...).
- Khi tiến trình chuyển từ trạng thái đang xử lý (running) sang trạng thái ready (ví dụ xảy ra một ngắt).
- Khi tiến trình chuyển từ trạng thái chờ (blocked) sang trạng thái ready ( ví dụ một thao tác nhập/xuất hoàn tất).
- Khi tiến trình kết thúc.

Các thuật toán điều phối theo nguyên tắc không độc quyền ngăn cản được tình trạng một tiến trình độc chiếm CPU, nhưng việc tạm dừng một tiến trình có thể dẫn đến các mâu thuẫn trong truy xuất, đòi hỏi phải sử dụng một phương pháp đồng bộ hóa thích hợp để giải quyết.

Trong các hệ thống sử dụng nguyên lý điều phối độc quyền có thể xảy ra tình trạng các tác vụ cần thời gian xử lý ngắn phải chờ tác vụ xử lý với thời gian rất dài hoàn tất! Nguyên lý điều phối độc quyền thường chỉ thích hợp với các hệ xử lý theo lô.

Đối với các hệ thống tương tác(time sharing), các hệ thời gian thực (realtime), cần phải sử dụng nguyên lý điều phối không độc quyền để các tiến trình quan trọng có cơ hội hồi đáp kịp thời. Tuy nhiên thực hiện điều phối theo nguyên lý không độc quyền đòi hỏi những cơ chế phức tạp trong việc phân định độ ưu tiên, và phát sinh thêm chi phí khi chuyển đổi CPU qua lại giữa các tiến trình.

### 2.2.3. Các danh sách sử dụng trong quá trình điều phối.

Hệ điều hành sử dụng hai loại danh sách để thực hiện điều phối các tiến trình là danh sách sẵn sàng (ready list) và danh sách chờ đợi(waiting list).

Khi một tiến trình bắt đầu đi vào hệ thống, nó được chèn vào danh sách các tác vụ (job list). Danh sách này bao gồm tất cả các tiến trình của hệ thống. Nhưng chỉ các tiến trình đang thường trú trong bộ nhớ chính và ở trạng thái sẵn sàng tiếp nhận CPU để hoạt động mới được đưa vào danh sách sẵn sàng.

Bộ điều phối sẽ chọn một tiến trình trong danh sách sẵn sàng và cấp CPU cho tiến trình đó. Tiến trình được cấp CPU sẽ thực hiện xử lý, và có thể chuyển



sang trạng thái chờ khi xảy ra các sự kiện như đợi một thao tác nhập/xuất hoàn tất, yêu cầu tài nguyên chưa được thỏa mãn, được yêu cầu tạm dừng ... Khi đó tiến trình sẽ được chuyển sang một danh sách chờ đợi. Hình 2.7 Các danh sách điều phối

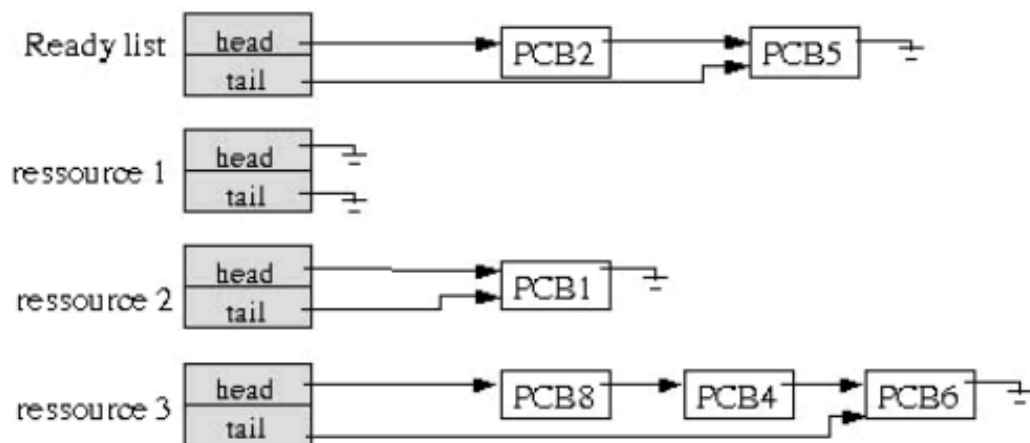
Hệ điều hành chỉ sử dụng một danh sách sẵn sàng cho toàn hệ thống, nhưng mỗi một tài nguyên ( thiết bị ngoại vi ) có một danh sách chờ đợi riêng bao gồm các tiến trình đang chờ được cấp phát tài nguyên đó.

Quá trình xử lý của một tiến trình trải qua những chu kỳ chuyển đổi qua lại giữa danh sách sẵn sàng và danh sách chờ đợi. Sơ đồ dưới đây mô tả sự điều phối các tiến trình dựa trên các danh sách của hệ thống.

Thoạt đầu tiến trình mới được đặt trong danh sách các tiến trình sẵn sàng (ready list), nó sẽ đợi trong danh sách này cho đến khi được chọn để cấp phát CPU và bắt đầu xử lý. Sau đó có thể xảy ra một trong các tình huống sau:

Tiến trình phát sinh một yêu cầu một tài nguyên mà hệ thống chưa thể đáp ứng, khi đó tiến trình sẽ được chuyển sang danh sách các tiến trình đang chờ tài nguyên tương ứng.

Tiến trình có thể bị bắt buộc tạm dừng xử lý do một ngắt xảy ra, khi đó tiến trình được đưa trở lại vào danh sách sẵn sàng để chờ được cấp CPU cho lượt tiếp theo. Sơ đồ chuyển đổi giữa các danh sách điều phối như sau:



Trong trường hợp đầu tiên, tiến trình cuối cùng sẽ chuyển từ trạng thái blocked sang trạng thái ready và lại được đưa trở vào danh sách sẵn sàng. Tiến trình lặp lại chu kỳ này cho đến khi hoàn tất tác vụ thì được hệ thống hủy bỏ khỏi mọi danh sách điều phối.

## CÂU HỎI VÀ BÀI TẬP

**Câu 1:** Động lực chính khiến cho các hệ điều hành hiện đại thường hỗ trợ môi trường đa nhiệm (multitask) là:

- A. Giảm hiệu suất sử dụng CPU, giảm tốc độ xử lý
- B. Giảm hiệu suất sử dụng CPU, tăng tốc độ xử lý
- C. Tăng hiệu suất sử dụng CPU, giảm tốc độ xử lý
- D. Tăng hiệu suất sử dụng CPU, tăng tốc độ xử lý

**Câu 2:** Chọn câu trả lời đúng:

Tiến trình (Process) là gì?

- A. Là một chương trình đang được thực thi
- B. Là một chương trình đang lưu trên đĩa
- C. Là một chương trình nạp vào bộ nhớ
- D. Là một chương trình đang chờ xử lý

**Câu 3:** Chọn đáp án sai:

Đâu không phải là trạng thái của tiến trình

- A. New (tạo mới một tiến trình)
- B. Running (tiến trình đang được thực thi)
- C. Waiting (tiến trình đang chờ sự kiện xảy ra)
- D. Swapping (trao đổi giữa các tiến trình)

**Câu 4:** Chọn câu trả lời đúng:

Để các tiến trình chia sẻ CPU một cách công bằng, không có tiến trình nào phải chờ đợi vô hạn để được cấp CPU, hệ điều hành dùng thành phần nào để giải quyết vấn đề này:

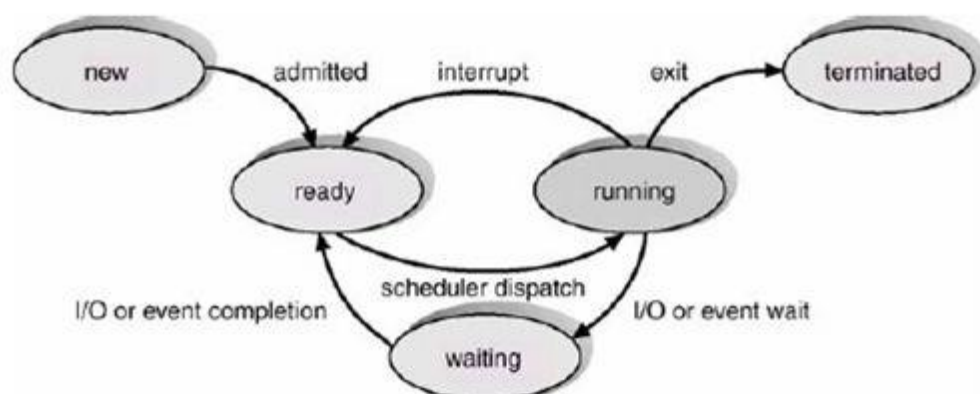
- A. Khối quản lý tiến trình
- B. Khối quản lý tài nguyên
- C. Bộ điều phối
- D. Bộ phân phối

**Câu 5:** Câu sau đây thuộc dạng chuyển trạng thái nào:

Tiến trình yêu cầu một tài nguyên nhưng chưa được đáp ứng vì tài nguyên chưa sẵn sàng, hoặc tiến trình phải chờ một sự kiện hay thao tác nhập xuất.

- A. Running -> Blocked
- B. Ready -> Running
- C. Running -> End
- D. Blocked -> Ready

**Câu 6:** Lưu đồ sau nói về nội dung gì?



- A. Lưu đồ mô tả các trạng thái của tiến trình
- B. Lưu đồ mô tả các khối của tiến trình
- C. Lưu đồ mô tả các khối điều khiển của tiến trình
- D. Lưu đồ mô tả các thông tin của tiến trình

**Câu 7:** Chọn câu trả lời đúng:

Có bao nhiêu trạng thái của tiến trình

- A. 3
- B. 4
- C. 5
- D. 6

**Câu 8:** Chọn câu trả lời đúng:

Một tiến trình (Process) bao gồm

- A. Bộ đếm chương trình và Ngăn xếp
- B. Ngăn xếp và Đoạn dữ liệu
- C. Đoạn dữ liệu và Bộ đếm chương trình
- D. Bộ đếm chương trình, Ngăn xếp và Đoạn dữ liệu

**Câu 9:** Chọn câu trả lời đúng:

Khi một tiến trình kết thúc xử lý, hệ điều hành huỷ bỏ nó bằng một số hoạt động nào sau đây?

- A. Tiếp tục ghi nhớ định danh của tiến trình.
- B. Không thu hồi các tài nguyên cấp phát cho tiến trình
- C. Thu hồi các tài nguyên cấp phát cho tiến trình, huỷ tiến trình ra khỏi tất cả các danh sách quản lý của hệ thống
- D. Tiếp tục ghi nhớ định danh của tiến trình và không thu hồi các tài nguyên cấp phát cho tiến trình, không huỷ tiến trình

**Câu 10:** Chọn câu trả lời đúng:

Tiến trình đang thực thi sẽ chuyển về loại danh sách nào khi xảy ra sự kiện đợi một thao tác nhập/xuất hoàn tất, yêu cầu tài nguyên dữ liệu chưa được thỏa mãn, yêu cầu tạm dừng:

- A. Danh sách chờ đợi (Waiting list)
- B. Danh sách sẵn sàng (Ready list)
- C. Danh sách tác vụ (Job list)
- D. Danh sách chờ đợi (Waiting list) và danh sách sẵn sàng (Ready list)

**Câu 11:** Chọn câu trả lời đúng:

Trạng thái BLOCKED của một process là do:

- A. Đang chờ nhập xuất

- B. Đang chờ một sự kiện nào đó chưa xảy ra.
- C. Đang chờ nhập xuất và đang chờ một sự kiện nào đó chưa xảy ra.
- D. Đang chờ một sự kiện nào đó chưa xảy ra và hủy process

**Câu 12:** *Chọn câu trả lời đúng:*

Hành động nào HĐH sẽ thực thi một process mới sinh ra:

- A. Cấp CPU ngay cho process.
- B. Tạo ngay khối PCB để quản lý process.
- C. Giao ngay các tài nguyên mà process cần.
- D. Cấp bộ nhớ để lưu process

**Câu 13:** *Chọn câu trả lời đúng:*

Độ ưu tiên của các process cho biết:

- A. Tầm quan trọng của process.
- B. Process chiếm nhiều hay ít vùng nhớ.
- C. Process sử dụng CPU nhiều hay ít.
- D. Thời điểm nạp Process

**Câu 14:** *Chọn câu trả lời đúng:*

Tại sao không có chuyển đổi từ trạng thái New sang trạng thái Terminate.

- A. Tiến trình đang chạy
- B. Tiến trình chưa thực hiện
- C. Tiến trình chưa đủ tài nguyên
- D. Tiến trình đang chờ

**Câu 15:** Công việc chính của lập lịch CPU là:

- A. Xác định thời điểm
- B. Chọn tiến trình được thực thi
- C. Xác định thời điểm, chọn toàn bộ tiến trình được thực thi
- D. Xác định thời điểm, chọn tiến trình được thực thi

**Câu 16:** Phát biểu nào sau đây là đúng

- A. Trong suốt thời gian tồn tại trong hệ thống, tiến trình được xem như thực hiện công việc chính: sử dụng CPU.
- B. Trong suốt thời gian tồn tại trong hệ thống, tiến trình được xem như thực hiện công việc chính: sử dụng thiết bị vào/ra.
- C. Trong suốt thời gian ở ngoài hệ thống, tiến trình được xem như thực hiện hai loại công việc chính: sử dụng CPU và sử dụng thiết bị vào/ra
- D. Trong suốt thời gian tồn tại trong hệ thống, tiến trình được xem như thực hiện hai loại công việc chính: sử dụng CPU và sử dụng thiết bị vào/ra.

**Câu 16:** Tiến trình loại CPU-bound là:

- A. Tiến trình có một hoặc nhiều phiên sử dụng CPU ngắn.
- B. Tiến trình không có nhiều phiên sử dụng CPU dài.
- C. Tiến trình không có phiên sử dụng CPU.
- D. Tiến trình có một hoặc nhiều phiên sử dụng CPU dài.

**Câu 17:** Tiến trình loại I/O-bound là:

- A. Tiến trình có một hoặc nhiều phiên sử dụng CPU dài.
- B. Tiến trình không có nhiều phiên sử dụng CPU ngắn.
- C. Tiến trình không có phiên sử dụng CPU và I/O.
- D. Tiến trình có một hoặc nhiều phiên sử dụng CPU ngắn (thời gian vào, ra nhiều).

**Câu 18:** Lập lịch non-preemptive là:

- A. Lập lịch ưu tiên
- B. Lập lịch ngắn
- C. Lập lịch dài
- D. Lập lịch không ưu tiên.

**Câu 19:** Lập lịch preemptive là:

- A. Lập lịch không ưu tiên
- B. Lập lịch ngắn
- C. Lập lịch dài
- D. Lập lịch ưu tiên.

**Câu hỏi tự luận:**

Câu 1: Tiến trình (Process) là gì?

Câu 2: Khi một tiến trình mới được sinh ra, hệ điều hành sẽ thực thi hành động gì

Câu 3: Khi một tiến trình mới được sinh ra mà bộ nhớ chưa đủ chỗ thì tiến trình đó sẽ được xử lý thế nào?

Câu 4: Các thành phần của đồ thị cấp phát tài nguyên. Điều phối tiến trình trong hệ điều hành đáp ứng những mục tiêu nào?

Câu 5: Trạng thái Blocked của một tiến trình là do nguyên nhân nào?