**1. Tiến trình**

**1.1 Khái niệm.**

- Tiến trình là 1 bộ phận của 1 chương trình đang trong quá trình thực hiện, đơn vị thực hiện tiến trình là processer.

- Một tiến trình thường bao gồm các thành phần sau:

* Các lệnh, tức là các lệnh chr thị cho CPU thực hiện.
* Phần dữ liệu chứa các biến toàn cục.
* Ngăn xếp (stack) tiến trình: chứa các dữ liệu tạm thời, ví dụ khi gọi 1 hàm, các tham số cần thiết để khôi phục lại trạng thái trước khi gọi hàm sẽ được lưu vào ngăn xếp.

**1.2 Các loại tiến trình.**

Các tiến trình trong hệ thống có thể chia thành 2 loại: Tiến trình tuần tự và tiến trình song song.

* *Tiến trình tuần tự*: là các tiến trình mà điểm khởi tạo của nó là điểm kết thúc của tiến trình trước đó.
* *Tiến trình song song*: là các tiến trình mà điểm khởi tạo của tiến trình này nằm ở thân các tiến trình khác, tức là có thể khởi tạo một tiến trình mới khi các tiến trình trước đó chưa kết thúc.Tiến trình song song đước chia tành nhiều loại:

- Tiến trình song song độc lập

- Tiến trình song song có quan hệ thông tin

- Tiến trình song song phân cấp

- Tiến trình song song đồng mức

**1.3 Quá trình tạo một tiến trình.**

Một tiến trình Windows được tạo khi mà ứng dụng gọi hàm tiến trình như là hàm Create Process, Create Process Á User, Create Process With token Who hoặc Create Process WithlogonW. Để tạo được một tiến trinhftthif cần những thông tin trong thư viện client- server Kernel32.dll, trình thực thi của Windows và tiến trình hệ thống con của Windows.

Để tạo 1 tiến trình với hàm API CreateProcess thì phải qua 6 bước cơ bản sau:

* Mở tệp thực thi (.exe).
* Tạo đối tượng thực thi tiến trình.
* Tạo luồng khởi tạo stack, và ngữ cảnh của nó.
* Thông báo cho hệ thống con windows về tiến trình mới được tạo.
* Bắt đầu thực thi luồng khởi tạo.
* Trong ngữ cảnh của luồng và tiến trình mới, hoàn thành việc khởi tạo của không gian địa chỉ mục đích chính là để nạp những thư viện liên kết động DLL và bắt đầu thực thi chương trình.

Trước khi gọi hàm image, hàm Createprocess thực hiện những bước sau:

* Trong hàm CreateProcess, mỗi thứ tự ưu tiên cho các tiến tình mới là một bit độc lập trong cờ CreationFlags, do đó có thể tạo một tiến trình có nhiều mức ưu tiên, Windows sẽ xem xét và ưu tiên từ thấp đến cao để gán cho trình tiến mới tạo.
* Nếu không có một thứ tự ưu tiên nào thì mặc định sẽ được đặt là Nornal.
* Nếu ứng dụng có mức ưu tiên là Real-time và tiên trình gọi không có khả năng Nâng quyền ưu tiên, thì tiến trình mới tạo ra sẽ được gán mức ưu tiên là mức cao.
* Tất cả các tiến trình tạo ra đều được gắn với một destop nào đó.

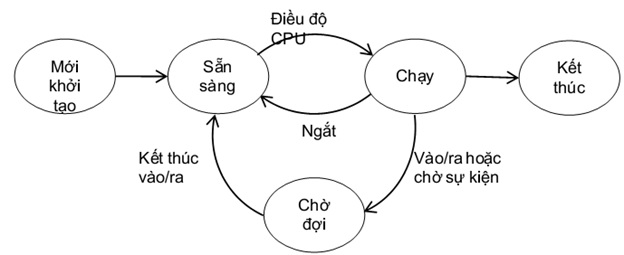
**1.4 Các trạng thái của một tiến trình.**

Là một thực thể động, tiến trình, tiến trình có thể thuộc những trạng thái khác nhau. Có nhiều cách phân biệt trạng thái tiến trình. Theo cách đơn giản nhất, tiến trình thuộc một trong 2 trạng thái: chạy và không chạy. Chạy là khi các lệnh của tiến trình được CPU thực hiện và không chạy là trường hợp ngược lại, ví dụ khi CPU đang được phân phối cho tiến trình khác hoặc khi tiến trình phải dừng để chờ kết quả vào ra.

Cách sử dụng hai trạng thái tiến trình là quá đơn giản và kkhong đủ để phản ánh đầy đủ thông tin về trạng thái tiến trình. Trên thực tế, hệ điều hành thường phân biệt năm trạng thái khác nhau của tiến trình: mới khởi tạo, sẵn sang, chạy, chờ đợi và kết thúc.

Ý nghĩa cụ thể 5 trạng thái như sau:

* *Trạng thái mới khởi tạo*; tiến trình đang được tạo ra. Hệ điều hành đã tạo ra các thông tin về tiến trình tuy nhiên tiến trình chưa được them vào danh sách những tiến trình được phép thực hiện. Thông thường, tiến trình này ở trạng thái chưa nằm trong bộ nhớ.
* *Trạng thái sẵn sàng*: tiến trình chờ được cấp CPU để thực hiện lệnh của mình.
* *Trạng thái chạy*; lệnh của tiến trình được CPU thực hiện. Với những máy tính có 1 CPU và CPU có một lõi, tại mỗi thời điểm chỉ có 1 tiến trình nằm trong trạng thái chạy.
* *Trạng thái chờ đợi*: tiến trình chờ đợi một sự kiện gì đó xảy ra, ví dụ chờ tín hiệu từ tiến trình khác hoặc chờ kết thúc quá trình vào/ra.
* *Trạng thái kết thúc*: tiến trình không còn nằm trong danh sách các tiến trình được thực hiện nhưng vẫn chưa bị xóa. Tiến trình thuộc về trạng thái này sau khi đã thực hiện xong hoặc bị tiến trình khác kết thúc



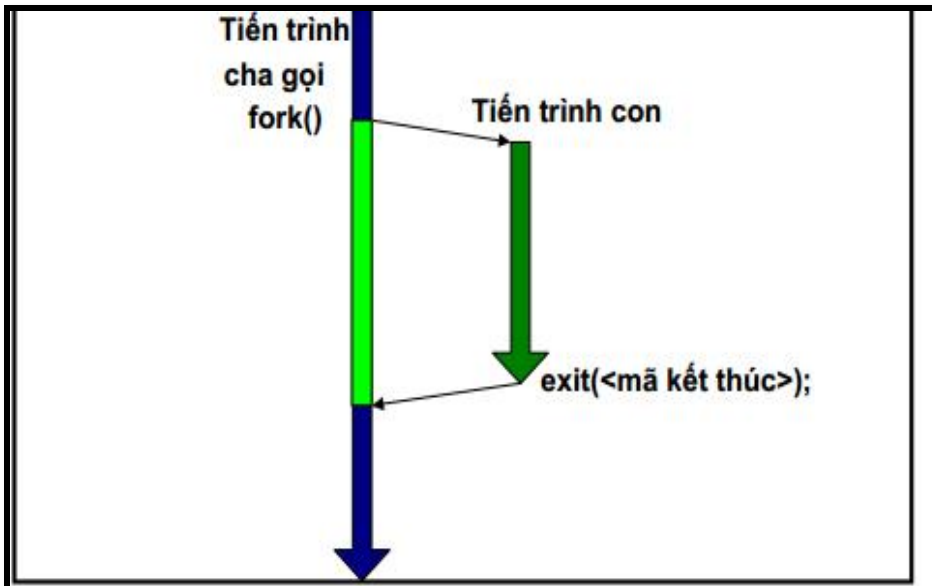
Hình 1: Sơ đồ chuyển trạng thái của tiến trình

**1.5 Liên lạc giữa các tiến trình.**

**1.5.1 Nhu cầu liên lạc giữa các tiến trình.**

Trong môi trường đa chương, một tiến trình không đơn độc trong hệ thống, mà có thể ảnh hưởng đến các tiến trình khác, hoặc bị các tiến trình khác tác động. Nói cách khác, các tiến trình là những thực thể độc lập, nhưng chúng vẫn có nhu cầu liên lạc với nhau để:

* **Chia sẻ thông tin:** nhiều tiến trình có thể cùng quan tâm đến những dữ liệu nào đó, do vậy hệ điều hành cần cung cấp 1 môi trường cho phép sự truy cập đông thời đến các dữ liệu chung.
* **Hợp tác hoàn thành tác vụ**: đôi khi để đạt được một xử lý nhanh chóng, người ta phân chia một tác vụ thành các công việc nhỏ có thể riến hành song song. Thường thì các công việc nhỏ này cần hợp tác với nhau để cùng hoàn thành nhiêm vụ ban đầu, ví dụ dữ liệu kết xuất của tiến trình này lại là dữ liệu nhập cho tiến trình khác… Trong các trường hợp đó, hệ điều hành cần cung cấp cơ chế để các tiến trình có thể trao đổi thông tin với nhau.

****

Hình 2. Mô tả tiến trình cha con

**1.5.2 Các vấn đề nảy sinh trong việc liên lạc giữa các tiến trình.**

Do mỗi tiến trình sở hữu một không gian địa chỉ riêng biệt, nên các tiến trình không thể liên lạc trực tiếp dễ dàng mà phải nhờ vào các cơ chế do hệ điều hành cung cấp. Khi cung cấp cơ chế liên lạc cho các tiến trình, hệ điều hành thường tìm giải pháp cho vấn đề chính yếu sau:

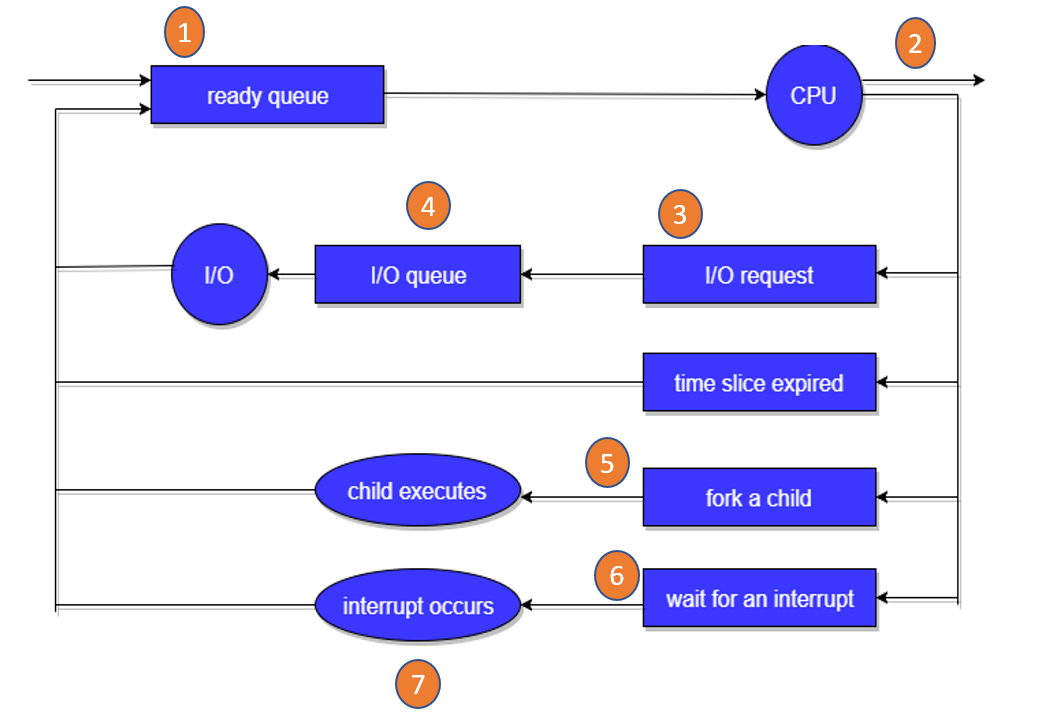
* Liên kết tường minh hay tiềm ẩn: Mối liên kết được gọi là tường minh khi được thiết lập rõ ràng, trực tiếp giữa các tiến trình và tiềm ẩn khi các tiến trình liên lạc với nhau thông qua một qui ước ngầm nào đó.
* Liên lạc theo chế độ đồng bộ hay không đồng bộ hay không đồng bộ: các tiến trình liên lạc theo cơ chế đồng bộ sẽ chờ nhau hoàn tất việc liên lạc, còn các tiên trình liên lạc theo cơ chế nonblocking thì không.
* Hầu hết các hệ điều hành đưa ra nhiều cơ chế liên lạc khác nhau, mỗi cơ chế có những đặc tính riêng và thích hợp hoàn cảnh chuyên biệt.

**2. Lập lịch tiến trình**

**2.1 Các hàng đợi tiến trình.**

* Hàng đợi công việc - Nó giúp bạn lưu trữ tất cả các quy trình trong hệ thống.
* Hàng đợi sẵn sàng - Loại hàng đợi này giúp bạn thiết lập mọi tiến trình nằm trong bộ nhớ chính, bộ nhớ này đã sẵn sàng và đang chờ thực thi.
* Hàng đợi thiết bị - Đây là một quá trình bị chặn do không có thiết bị I / O.

**2.2 Sơ đồ định thời tiến trình.**



Hình 3 Sơ đồ tiến trình

Trong Sơ đồ đã cho ở trên,

* Hình chữ nhật đại diện cho một hàng đợi.
* Vòng tròn biểu thị tài nguyên
* Mũi tên chỉ ra quy trình.

1. Mọi quy trình mới lần đầu tiên được đưa vào hàng đợi Sẵn sàng. Nó sẽ đợi trong hàng đợi sẵn sàng cho đến khi nó được xử lý cuối cùng để thực thi. Tại đây, tiến trình mới được đưa vào hàng đợi sẵn sàng và đợi cho đến khi nó được chọn để thực thi hoặc nó được gửi đi.
2. Một trong các quy trình được cấp phát CPU và nó đang thực thi
3. Quy trình sẽ đưa ra một yêu cầu I / O
4. Sau đó, nó sẽ được đặt trong hàng đợi I / O.
5. Quá trình này sẽ tạo ra một quy trình con mới
6. Quá trình này sẽ được chờ đợi cho đến khi kết thúc.
7. Nó sẽ loại bỏ mạnh mẽ khỏi CPU, do đó làm gián đoạn. Sau khi hoàn thành ngắt, nó sẽ được gửi trở lại hàng đợi sẵn sàng.

**2.3 Mục tiêu của lập lịch tiến trình.**

* Tối đa hóa số lượng người dùng tương tác trong thời gian phản hồi có thể chấp nhận được.
* Đạt được sự cân bằng giữa phản hồi và sử dụng.
* Tránh trì hoãn vô thời hạn và thực thi các ưu tiên.
* Nó cũng nên tham chiếu đến các quá trình nắm giữ các nguồn tài nguyên quan trọng.
  1. **Các trình lập lịch.**
* Trình lập lịch dài kỳ(Long-term schedulers): còn được gọi là trình lên lịch công việc . trình lập lịch này điều chỉnh chương trình và quá trình chọn từ hàng đợi và tải chúng vào bộ nhớ để thực thi. Nó cũng quy định mức độ đa chương trình. Tuy nhiên, mục tiêu chính của loại trình lập lịch này là cung cấp sự kết hợp cân bằng giữa các công việc, như Bộ xử lý, các công việc I / O., Cho phép quản lý đa chương trình.
* Trình lập lịch trung kỳ (Intermediate scheduler): là một phần quan trọng của việc hoán đổi . Nó cho phép bạn xử lý các quy trình ngoài được hoán đổi. Trong trình lập lịch này, một quá trình đang chạy có thể bị tạm ngưng, điều này tạo ra một yêu cầu I / O. Quá trình đang chạy có thể bị tạm ngưng nếu nó đưa ra yêu cầu I / O. Quá trình bị tạm ngừng không thể đạt được bất kỳ tiến độ nào để hoàn thành. Để xóa quá trình khỏi bộ nhớ và tạo không gian cho các quá trình khác, quá trình bị treo nên được chuyển sang bộ nhớ thứ cấp.
* Trình lập lịch ngắn kỳ(Short-term schedulers): còn được gọi là **bộ lập lịch CPU** . Mục tiêu chính của trình lập lịch này là tăng hiệu suất hệ thống theo các tiêu chí đã đặt. Điều này giúp bạn chọn từ một nhóm các quy trình sẵn sàng thực thi và phân bổ CPU cho một trong số chúng. Người điều phối trao quyền kiểm soát CPU cho quá trình được chọn bởi trình lập lịch ngắn hạn.

**3. Các hoạt động trên tiến trình**

**3.1 Thao tác trên tiến trình.**

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

http://dembinhyen.free.fr/UDS/Ebook/CD1/He%20Dieu%20Hanh/Htm/images/red.gif tạo lập tiến trình (create)

http://dembinhyen.free.fr/UDS/Ebook/CD1/He%20Dieu%20Hanh/Htm/images/red.gif kết thúc tiến trình (destroy)

http://dembinhyen.free.fr/UDS/Ebook/CD1/He%20Dieu%20Hanh/Htm/images/red.gif tạm dừng tiến trình (suspend)

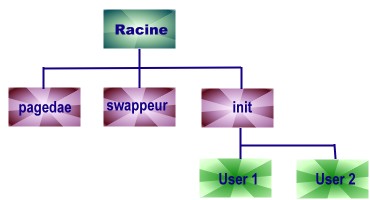
http://dembinhyen.free.fr/UDS/Ebook/CD1/He%20Dieu%20Hanh/Htm/images/red.gif tái kích hoạt tiến trình (resume)

 thay đổi độ ưu tiên tiến trình



**3.2 Sự tạo tiến trình.**

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*.



**Hình 4.** Một cây tiến trình trong hệ thống UNIX

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

**3.3 Sự kết thúc tiên 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 yêu cầu hệ điều hành kết thúc xử lý của một tiến trình khác. 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.