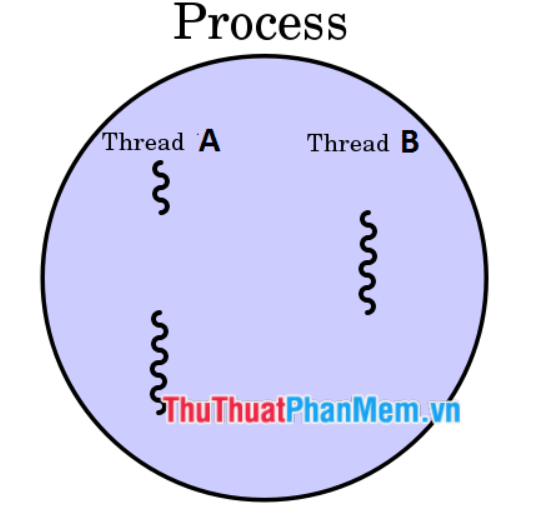
**SỰ KHÁC BIỆT GIỮA TIẾN TRÌNH VÀ LUỒNG**

**1. Process là gì?**



Process, hay tiến trình, là sự thực thi của một chương trình và thực hiện các hành động liên quan được chỉ định trong một chương trình, hoặc nó là một đơn vị thực thi nơi chương trình chạy. Hệ điều hành tạo, lên lịch và chấm dứt các tiến trình. Các tiến trình khác được tạo bởi tiến chính được gọi là tiến trình con.

Hoạt động của bất kỳ tiến trình nào cũng được kiểm soát bởi khối điều khiển tiến trình (Process Control Block, viết tắt là PCB). PCB chứa tất cả những thông tin quan trọng liên quan đến các tiến trình, chẳng hạn như: id tiến trình, ưu tiên, trạng thái, CPU,…

Vòng đời của một tiến trình có một số trạng thái nhất định như ready (sẵn sàng), running (đang chạy), blocked (bị chặn) và terminated (đã chấm dứt). Các trạng thái của tiến trình được sử dụng để theo dõi hoạt động của tiến trình đó ở thời điểm hiện tại.

Những hoạt động xen kẽ của các tiến trình giúp nâng cao tốc đô tính toán khi hoạt động I/O trong một tiến trình trùng lặp với hoạt động tính toán trong tiến trình khác.

Các thuộc tính của một tiến trình:

- Việc tạo mỗi tiến trình bao gồm các cuộc gọi hệ thống cho từng tiến trình riêng biệt.

- Một tiến trình là một thực thể bị cô lập và không chia sẻ dữ liệu thông tin.

- Các tiến trình sử dụng cơ chế IPC (Inter-process Communication, tức giao tiếp giữa các tiến trình) để liên lạc làm tăng đáng kể số lượng cuộc gọi hệ thống.

- Quản lý tiến trình tiêu thụ nhiều cuộc gọi hệ thống hơn.

- Mỗi tiến trình đều có ngăn xếp và bộ nhớ heap, dữ liệu và bản đồ bộ nhớ riêng.

**2. Thread là gì?**

Thread hay luồng, là một tiến trình nhỏ có thể được quản lý độc lập bởi một bộ lập lịch. Tất cả các luồng trong một chương trình đơn được chứa hợp lý trong một tiến trình. Nhân cấp phát một ngăn xếp và khối điều khiển luồng (TCB) cho mỗi luồng. Hệ điều hành chỉ lưu con trỏ ngăn xếp và trạng thái CPU tại thời điểm chuyển đổi giữa các luồng của cùng một tiến trình.

Luồng được triển khai theo ba cách khác nhau: luồng cấp nhân, luồng cấp người dùng, và luồng lai. Luồng có thể có ba trạng thái running (đag chạy), ready (sẵn sàng) và blocked (bị chặn); nó chỉ bao gồm trạng thái tính toán không phân bổ tài nguyên và trạng thái giao tiếp làm giảm chi phí chuyển đổi, khiến tăng cường sự tương tranh (song song) do đó tốc độ cũng tăng lên.

Đa luồng cũng đi kèm với nhiều vấn đề. Việc có nhiều luồng không tạo ra sự phức tạp, nhưng sự tương tác giữa chúng thì không như vậy.

Một luồng phải có thuộc tính ưu tiên khi có nhiều luồng đang hoạt động. Thời gian để nó thực thi tương ứng với các luồng đang hoạt động khác trong cùng tiến trình được chỉ định bởi mức độ ưu tiên của luồng.

Các thuộc tính của một luồng:

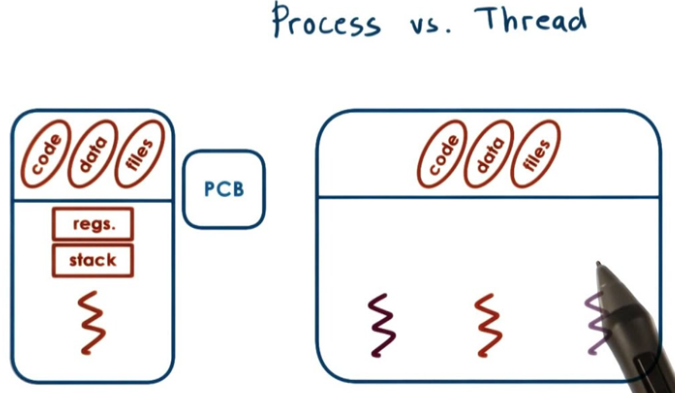
- Một cuộc gọi hệ thống có thể tạo nhiều hơn một luồng.

- Luồng chia sẻ dữ liệu và thông tin.

- Luồng chia sẻ bộ nhớ heap nhưng có ngăn xếp và thanh ghi riêng của mình.

- Quản lý luồng không tiêu thụ hoặc tiêu thị ít cuộc gọi hệ thống hơn vì giao tiếp giữa các luồng có thể đạt được bằng cách sử dụng bộ nhớ dùng chung.

- Thuộc tính cô lập của tiến trình làm tăng chi phí của nó về mặt tiêu thụ tài nguyên.





Tóm lại, ta có thể tổng kết những điểm khác biệt quan trọng nhất như sau:

- Tất cả các luồng của một chương trình được chứa một cách hợp lý trong một tiến trình.

- Luồng nhẹ hơn tiến trình.

- Một tiến trình là một đơn vị thực thi bị cô lập trong khi luồng không bị cô lập và có chia sẻ bộ nhớ.

- Một luồng không thể tồn tại riêng biệt; nó được gắn liền với một tiến trình. Mặt khác, một tiến trình có thể tồn tại riêng lẻ.

- Tại thời điểm hết hạn của một luồng, ngăn xếp liên kết của nó có thể được phục hồi vì mỗi luồng có ngăn xếp riêng. Ngược lại, nếu một tiến trình chết, tất cả các luồng cũng sẽ chết bao gồm cả tiến trình