

Thông tin dưới đây được dịch từ tài liệu StudyGuide tại trang OS-Book của sách giáo trình chính và phần Summary của sách. Mọi lỗi sai sót hay góp ý, xin gửi về cho tôi qua email trantrungtin@tdtu.edu.vn

Study Guide for Lecture notes ch03

- Tiến trình (Process) chứa mã nguồn bộ đếm chương trình, ngăn xếp, đồng và dữ liệu.

- Text section: mã nguồn của chính chương trình đó.

- Stack: dữ liệu tạm thời (các tham số hàm, địa chỉ trả về, các biến cục bộ)

- Data section: các biến số toàn cục (còn có thể chia ra biến số có giá trị khởi tạo và không có khởi tạo).

- Heap: vùng nhớ dành cho cấp phát động trong thời gian chạy.

- Khối điều khiển tiến trình (Process Control Block / PCB): chứa các thông tin liên quan đến từng tiến trình như là: trạng thái tiến trình, giá trị bộ đếm chương trình, thông tin định thời CPU, giá trị các thanh ghi, thông tin định thời, thông tin thống kê, trạng thái nhập / xuất.

- Có 2 loại tiến trình:

- I/O Bound: dành nhiều thời gian cho nhập xuất hơn là tính toán, ít cần năng lực CPU.

- CPU Bound: dành nhiều thời gian cho tính toán, rất cần CPU để chạy.

- Khi CPU chuyển sang xử lý tiến trình khác, hệ thống cần lưu lại trạng thái của tiến trình sắp tạm ngừng và trong PCB của nó và tải trạng thái đã lưu từ PCB của tiến trình sắp chuyển đến. Quá trình đó gọi là chuyển ngữ cảnh (context switch)

- Thời gian chuyển ngữ cảnh phụ thuộc vào phần cứng.

- Tiến trình cha (Parent processes) tạo ra các tiến trình con, hình thành nên cây tiến trình.

- + Process ID (PID) là chỉ số để quản lý các tiến trình.

- + Cha và con có thể: chia sẻ toàn bộ tài nguyên / chia sẻ một phần tài nguyên hoặc không chia sẻ gì cả.

- + Tiến trình cha có thể thực thi đồng thời với tiến trình con, hoặc là chờ cho đến khi tiến trình con kết thúc.

- + Lệnh gọi hệ thống fork() tạo ra một tiến trình mới.

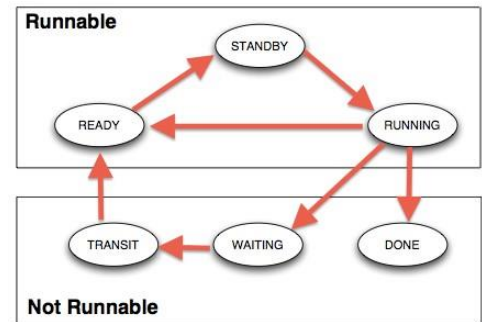
- + Lệnh gọi hệ thống exec() được sử dụng sau lệnh gọi fork() để thay thế không gian bộ nhớ bằng một chương trình mới.

- Tiến trình cộng tác (Cooperating processes) cần cơ chế interprocess communication (IPC): có thể hiện thực bằng vùng nhớ chia sẻ hoặc gửi thông điệp.

- Gửi thông điệp (Message passing) có thể là chặn (blocking) hoặc không (non-blocking).

- + Blocking là đồng bộ (synchronous).

- Blocking send là người gửi bị chặn cho đến khi thông điệp đã gửi được nhận.



- Blocking receive là người nhận bị chặn cho đến khi có thông điệp mới.
- Non-blocking là bất đồng bộ (asynchronous)
 - Non-blocking send là người gửi cứ gửi thông điệp và tiếp tục gửi.
 - Non-blocking receive là người nhận có thể nhận thông điệp hoặc thông điệp rỗng (null).

Summary Chapter 03 of book “OS concepts 10th edition”

- Tiến trình là một chương trình đang được thực thi và trạng thái của hoạt động hiện tại của một tiến trình được thể hiện ở bộ đếm chương trình, cũng như giá trị các thanh ghi khác.
- Bộ cục của một tiến trình trong bộ nhớ chính có bốn phần khác nhau: (1) mã chương trình, (2) dữ liệu, (3) đống (heap) và (4) ngăn xếp.
- Khi một chương trình được khởi chạy (bằng cách click chuột, gõ lệnh, chạm vào icon...), nó trở thành tiến trình, được nạp vào trong bộ nhớ chính, và mang trạng thái “mới”.
- Khi một tiến trình thực thi, nó thay đổi trạng thái. Có bốn trạng thái chung của một tiến trình: (1) sẵn sàng, (2) đang chạy, (3) chờ đợi nhập xuất và (4) chấm dứt.
- Khối điều khiển tiến trình (PCB) là cấu trúc dữ liệu nhân đại diện cho một tiến trình bên trong một hệ điều hành.
- Vai trò của bộ định thời / lập lịch tiến trình là chọn một tiến trình đang có trạng thái “sẵn sàng” để thực thi trên CPU.
- Hệ điều hành thực hiện chuyển đổi ngữ cảnh khi chuyển thực thi một tiến trình đến thực thi một tiến trình khác.
- Các lời gọi hệ thống fork () và CreateProcess () được sử dụng để tạo các tiến trình các hệ thống UNIX và Windows.
- Khi bộ nhớ dùng chung được sử dụng để liên lạc giữa các tiến trình, hai (hoặc nhiều hơn) các tiến trình chia sẻ cùng một vùng bộ nhớ. POSIX cung cấp một API cho bộ nhớ chia sẻ.
- Hai tiến trình có thể giao tiếp bằng cách trao đổi tin nhắn với một tiến trình khác bằng cách sử dụng cơ chế chuyển tin (message passing). Hệ điều hành Mach sử dụng cơ chế chuyển tin nhắn làm hình thức chính để các tiến trình giao tiếp với nhau. Windows cũng cung cấp một khuôn mẫu để chuyển tin nhắn.
- Một đường ống (pipe) cung cấp một ống dẫn cho hai tiến trình giao tiếp với nhau. Có hai dạng đường ống: thông thường và được đặt tên. Đường ống thông thường (Ordinary pipe) được thiết kế cho giao tiếp giữa các tiến trình có mối quan hệ cha - con. Đường ống được đặt tên (Named pipe) phổ thông hơn và cho phép vài tiến trình giao tiếp với nhau.
- Các hệ thống UNIX cung cấp các đường ống thông thường thông qua lời gọi hệ thống pipe(). Đường ống thông thường có một đầu đọc và đầu ghi. Một tiến trình cha có thể gửi dữ liệu đến đường ống bằng cách sử dụng đầu ghi của nó và tiến trình con có thể đọc được thông tin từ đầu đọc của nó. Các đường ống được đặt tên trong UNIX được gọi là các FIFO.
- Các hệ thống Windows cũng cung cấp hai dạng đường ống: ẩn danh (Anonymous pipe) và có đặt tên (Named pipe). Các đường ống ẩn danh tương tự như các đường ống thông thường của UNIX. Các đường ống đó là đơn hướng và sử dụng các mối quan hệ cha - con giữa các tiến trình đang giao tiếp với nhau. Các đường ống được đặt tên cung cấp một hình thức giao tiếp giữa các tiến trình phong phú hơn là UNIX (là các FIFO).
- Hai hình thức giao tiếp phổ biến giữa máy khách-máy chủ (client-server) chủ là socket và các cuộc gọi thủ tục từ xa (RPC - remote procedure calls). Socket cho phép hai tiến trình trên hai máy tính khác nhau giao tiếp qua mạng. RPC trừu tượng hóa khái niệm về lời gọi Hàm (thủ tục) theo cách mà một hàm nào đó có thể được gọi trên một tiến trình khác đang nằm trên một máy tính khác.
- Hệ điều hành Android sử dụng những RPC như là một khuôn mẫu giao tiếp giữa các tiến trình sử dụng khung kết dính của nó.