



FACULTY OF INFORMATION TECHNOLOGY
>>> COURSE MATERIAL <<<

INTRODUCTION TO OPERATING SYSTEM
Course ID 502047

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 ch04

- Tiểu trình, hay còn gọi là luồng, (Thread) là những đơn vị cơ bản sử dụng CPU, tạo thành nền tảng của máy tính đa luồng.
- Tạo ra tiến trình thì tốn nhiều tài nguyên, thời gian, “nặng” (heavy-weight) trong khi tạo ra tiểu trình thì “nhẹ” hơn (light-weight).
 - Có thể tinh giản mã lệnh và tăng hiệu quả.
- Nhân của các hệ điều hành hiện nay phần lớn là đa luồng (multi-threaded).
- Mô hình hệ thống đa luồng Multi-threading bao gồm: Many-to-One, One-to-One, Many-to-Many
 - Many-to-One: Nhiều tiểu trình mức người dùng (user-level thread) ánh xạ vào một tiểu trình cấp nhân (kernel thread).
 - One-to-One: Từng tiểu trình mức người dùng ánh xạ đơn ánh vào một tiểu trình mức nhân. Mô hình thông dụng, sử dụng trong Windows và Linux.
 - Many-to-Many: Nhiều tiểu trình cấp người dùng ánh xạ vào nhiều tiểu trình cấp nhân.
- Thư viện tiểu trình (Thread library) cung cấp cho lập trình viên thư viện / API để tạo và quản lý tiểu trình.
- Những vấn đề của Tiểu trình bao gồm: hủy bỏ một tiểu trình, gửi nhận tín hiệu (đồng bộ / bất đồng bộ), xử lý dữ liệu riêng của tiểu trình và các hoạt động định thời.
 - Hủy bỏ(Cancellation):
 - Hủy bỏ không đồng bộ chấm dứt tiểu trình chỉ định ngay lập tức
 - Hủy bỏ hoãn cho phép tiến trình chỉ định kiểm tra định kỳ nếu nó cần được hủy
 - Trình xử lý tín hiệu (Signal handler) xử lý các tín hiệu được tạo bởi một sự kiện cụ thể và được gửi đến một tiến trình.
 - Bộ định thời (Scheduler) kích hoạt cung cấp upcalls - một cơ chế giao tiếp từ nhân đến thư viện tiểu trình.
 - Cho phép ứng dụng duy trì đúng số lượng tiểu trình mức nhân.

Summary Chapter 04 of book “OS concepts 10th edition”

- Một tiểu trình thể hiện như một đơn vị cơ bản của việc sử dụng CPU và các tiểu trình thuộc về cùng một tiến trình, chia sẻ nhiều tài nguyên tiến trình, bao gồm cả mã nguồn và dữ liệu.
- Có bốn lợi ích chính cho các ứng dụng đa luồng: (1) tính đáp ứng, (2) chia sẻ tài nguyên, (3) tính kinh tế và (4) khả năng mở rộng.
- Xử lý đồng thời là khi nhiều tiểu trình đang tiến triển, trong khi xử lý cùng lúc là khi nhiều tiểu trình đang tiến triển cùng nhau. Trên một hệ thống có một CPU, chỉ có thể xử lý đồng thời; xử lý song song đòi hỏi một hệ thống đa nhân cung cấp nhiều nhân xử lý.
- Có một số thách thức trong việc thiết kế các ứng dụng đa luồng. Chúng bao gồm phân chia và cân bằng công việc, phân chia dữ liệu giữa các tiểu trình khác nhau và xác định tất cả phụ thuộc dữ liệu. Cuối cùng, những chương trình đa luồng đặc biệt khó khăn để kiểm tra và gỡ lỗi.
- Song song dữ liệu phân phối các tập hợp con của cùng một dữ liệu trên các nhân tính toán khác nhau và thực hiện các thao tác tương tự trên mỗi lõi. Song song tác vụ không phân phối dữ liệu mà là các tác vụ trên nhiều lõi. Mỗi tác vụ là thực thi một thao tác độc nhất.
- Ứng dụng người dùng tạo các tiểu trình mức người dùng, và chúng được ánh xạ tới các tiểu trình mức nhân để thực thi trên CPU. Mô hình nhiều-một ánh xạ nhiều tiểu trình mức người dùng đến một tiểu trình mức nhân. Các cách tiếp cận khác bao gồm các mô hình một-một và nhiều-nhiều.
- Thư viện tiểu trình cung cấp API để tạo và quản lý tiểu trình. Có ba thư viện tiểu trình thông dụng bao gồm Windows, Pthreads và tiểu trình Java. Windows chỉ dành cho hệ thống Windows, trong khi Pthreads có sẵn cho các hệ thống tương thích POSIX như UNIX, Linux và macOS. Tiểu trình Java sẽ chạy trên bất kỳ hệ thống nào hỗ trợ máy ảo Java.
- Tiểu trình ngầm ẩn liên quan đến việc xác định các tác vụ, không phải là các tiểu trình, và cho phép các ngôn ngữ hoặc framework API để tạo và quản lý tiểu trình. Có một số cách tiếp cận để phân luồng ngầm, bao gồm thăm dò tiến trình, framework “fork-join” và Grand Central Dispatch. Luồng ngầm đang trở thành một kỹ thuật ngày càng phổ biến cho các lập trình viên sử dụng trong việc phát triển ứng dụng đồng thời và song song.
- Tiến trình có thể bị chấm dứt một cách không đồng bộ hoặc trì hoãn. Chấm dứt không đồng bộ dừng một luồng ngay lập tức, ngay cả khi nó đang thực hiện một bản cập nhật. Chấm dứt trì hoãn thông báo một tiểu trình nó cần phải đóng lại nhưng cho phép tiểu trình các tiểu trình kết thúc trong một thứ tự ngăn nắp. Trong hầu hết các trường hợp, hủy bỏ trì hoãn được ưu tiên để chấm dứt không đồng bộ.
- Không giống như nhiều hệ điều hành khác, Linux không phân biệt giữa tiến trình và tiểu trình; mà gọi chung là tác vụ. Lệnh gọi hệ thống clone() của Linux có thể được sử dụng để tạo các tác vụ mà chúng cư xử giống như tiến trình hơn hay giống như tiểu trình hơn.