

Bài 12.1: Tổng quan về đa luồng

- ✓ Tổng quan
- ✓ Tiến trình và luồng
- ✓ Khi nào sử dụng đa luồng?

Tổng quan

- ✓ Đa luồng cho phép ta tăng khả năng phản hồi của ứng dụng.
- ✓ Nếu chạy ứng dụng trên vi xử lý đa lõi hoặc hệ điều hành đa nhân giúp tận dụng tối đa nền tảng sẵn có.

Tiến trình vs luồng

- ✓ Một tiến trình(process) là một chương trình có thể chạy được.
- ✓ Hệ điều hành sử dụng các tiến trình để phân tách các ứng dụng đang được thực thi.
- ✓ Một luồng(thread) là đơn vị cơ bản mà hệ điều hành phân bổ thời gian sử dụng vi xử lý.
- ✓ Mỗi luồng có một bộ lập lịch ưu tiên và duy trì một tập cấu trúc hệ thống sử dụng để lưu trữ ngữ cảnh luồng khi việc thực thi của luồng bị tạm dừng.
- ✓ Ngữ cảnh của luồng bao gồm tất cả các thông tin mà luồng cần để tiếp tục thực thi một cách liền mạch, bao gồm một tập các thanh ghi của CPU và ngăn xếp luồng.
- ✓ Một tiến trình có thể chứa nhiều luồng. Tất cả các luồng của một tiến trình chia sẻ chung không gian địa chỉ ảo của tiến trình chứa nó.
- ✓ Một luồng có thể thực thi bất kỳ phần nào của mã chương trình, bao gồm các phần hiện đang được thực thi bởi một luồng khác.

Tiến trình vs luồng

- ✓ Theo mặc định, chương trình .NET sẽ khởi chạy với một luồng đơn. Thường gọi là luồng chính.
- ✓ Sau đó nó có thể tạo ra các luồng khác để thực thi mã lệnh ở chế độ chạy song song.
- ✓ Các luồng được tạo ra và chạy song song với luồng chính được gọi là **worker threads**.

Khi nào sử dụng đa luồng

- ✓ Sử dụng đa luồng trong trường hợp phân chia nhiệm vụ cho từng thành phần của ứng dụng. Ví dụ 1 luồng chính quản lý giao diện và nhận tương tác người dùng. Luồng phụ thực hiện các nhiệm vụ nặng, tốn thời gian như tải file lớn, sao chép file, xử lý tính toán...
- ✓ Sử dụng đa luồng tạo các luồng chuyên biệt cho kết nối mạng, giao tiếp giữa các thiết bị nhằm tăng khả năng phản hồi với những thông tin hay sự kiện xảy ra.
- ✓ Sử dụng đa luồng để thực thi các hành động chạy đồng thời nhằm giảm thời gian chờ đợi. Lúc này mỗi công việc sẽ được gắn vào 1 luồng và thực hiện đồng thời.

Khi nào sử dụng đa luồng

- ✓ Từ phiên bản .NET 4, khuyến nghị sử dụng Task Parallel Library(TPL) và Parallel LINQ(PLINQ).
- ✓ Cả 2 thư viện trên đều dựa trên luồng ThreadPool. Mỗi ThreadPool chứa một nhóm các luồng worker threads.
- ✓ Cuối cùng bạn có thể sử dụng lớp Thread của System.Threading để tạo ra các thread được quản lý.



Nội dung tiếp theo

Tạo và truyền dữ liệu vào luồng