

LOGO

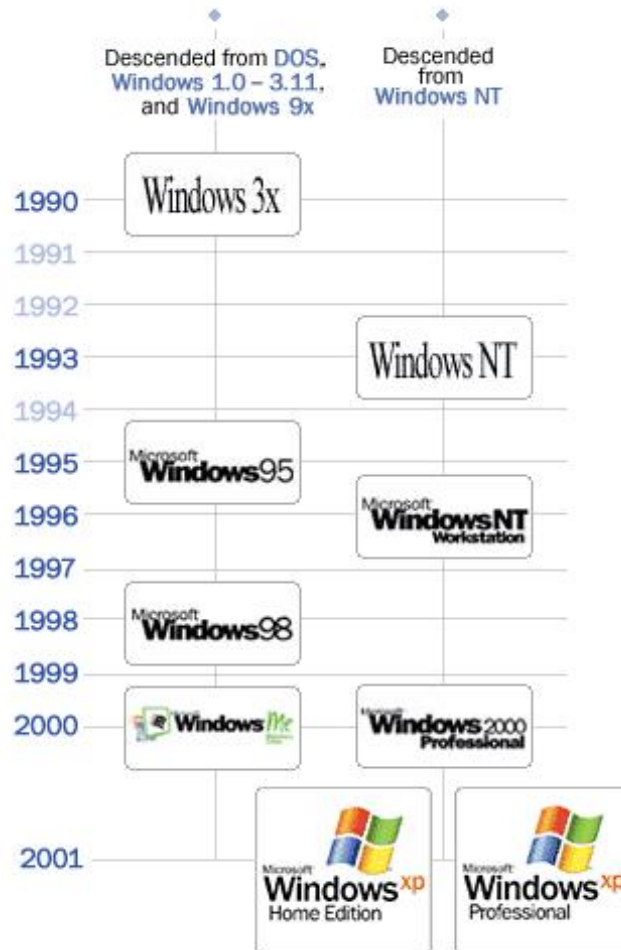


Tổng quan lập trình môi trường Windows

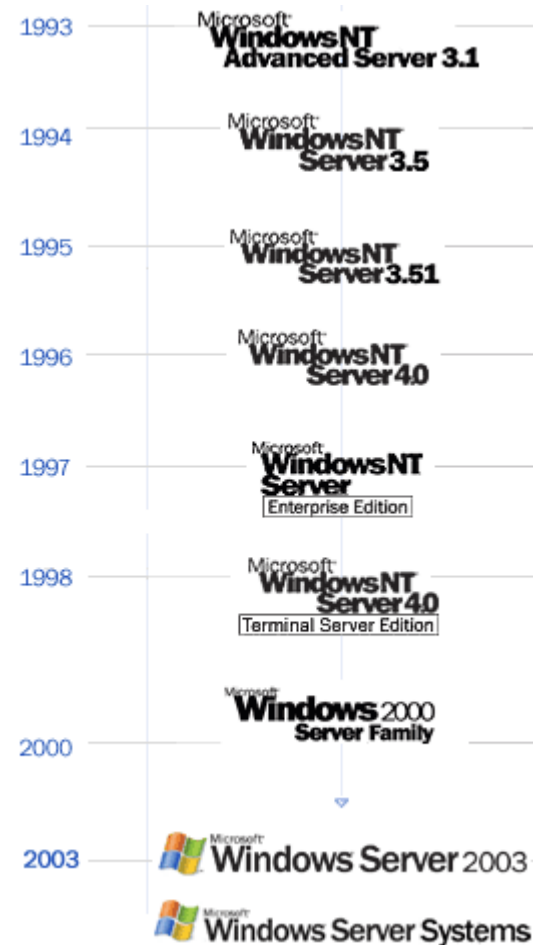
Phạm Thi Vương

Lịch sử Windows

Windows Desktop



Windows Server



Môi trường Windows

- ✓ Hệ điều hành 32/64 bit
- ✓ Giao diện người dùng kiểu đồ họa (GUI)
- ✓ Đa nhiệm

Giao diện người dùng kiểu đồ họa (GUI)

- ✓ Visual Interface
- ✓ WYSIWYG
- ✓ Các ứng dụng Windows có giao diện thống nhất với cùng dáng vẻ:
 - § một cửa sổ hình chữ nhật
 - § thanh tiêu đề
 - § menu, hộp hội thoại, thanh trượt

Đa nhiệm

- ✓ Nhiều chương trình có thể được kích hoạt và chạy cùng một lúc
- ✓ Mỗi chương trình chiếm một phần tài nguyên của hệ thống và có một phần bộ nhớ thường trú
- ✓ Windows 32 bits hoạt động theo cơ chế preemptive. Mỗi chương trình có thể được tách thành các thread thực hiện cùng lúc

Quản lý bộ nhớ

- ✓ Mã chương trình và dữ liệu có thể dịch chuyển trong bộ nhớ vật lý
- ✓ Khả năng sử dụng bộ nhớ phụ
- ✓ Các thư viện liên kết động (.DLL)

Tư tưởng hướng đối tượng

- ✓ Cửa sổ ứng dụng chính và các cửa sổ con là các thể hiện (đối tượng) của lớp cửa sổ được đăng ký trước

Giao diện đồ hoạ độc lập thiết bị

- ✓ Giao diện đồ hoạ của Window cho phép tạo các đối tượng đồ hoạ và định dạng văn bản cho cả màn hình và máy in
- ✓ Chương trình viết cho Windows không điều khiển trực tiếp thiết bị xuất như màn hình hay máy in mà thông qua một ngôn ngữ lập trình đồ hoạ gọi là Giao Diện Thiết Bị Đồ Hoạ (Graphics Device Interface: GDI)

Kiến trúc hướng thông điệp (message driven)

- ✓ Windows và các ứng dụng Windows hoạt động theo cơ chế truyền, nhận thông điệp
- ✓ Các hoạt động của chương trình thay đổi tùy theo thông điệp mà nó nhận được,
- ✓ Thông điệp được gửi qua lại giữa ứng dụng và Windows, giữa các ứng dụng với nhau
- ✓ Cửa sổ ứng dụng tự động được vẽ lại mỗi khi có sự thay đổi kích thước hay vùng bị che. Điều này được thực hiện nhờ hệ điều hành gửi thông điệp cho chương trình

Thủ tục cửa sổ

- ✓ Hệ điều hành ra lệnh cho ứng dụng nhờ thủ tục cửa sổ (Window Procedure)
- ✓ Hàm cửa sổ cho biết phản ứng của chương trình với các tác động bên ngoài như user input...
- ✓ Hệ điều hành Windows gửi thông điệp cho một chương trình ứng dụng bằng cách gọi hàm cửa sổ của ứng dụng đó, với tham số là thông điệp

Đặc điểm cơ bản

- Lập trình hướng sự kiện, dựa vào thông điệp
- Đa nhiệm
- Môi trường 32/64 bits
- Hỗ trợ thư viện liên kết động

Windows và DOS

Windows	DOS
Lập trình sự kiện, dựa vào thông điệp (message)	Thực hiện tuần tự theo chỉ định
Multi-tasking	Single task
Multi-CPU	Single CPU
Tích hợp sẵn Multimedia	Phải dùng các thư viện Multimedia riêng
Hỗ trợ 32 bits hay hơn nữa	Ứng dụng 16 bits
DLL, OLE, DDE, COM, OpenGL, DirectX	Không có

Các loại ứng dụng cơ bản

- **Windows Application:** ứng dụng 32 bits, giao diện đồ họa
- **Win32 DLL:** ứng dụng 32 bits, dạng thư viện liên kết động (Dynamic-linked library)
- **Win32 LIB:** ứng dụng 32 bits, dạng thư viện liên kết tĩnh (Static-linked library)
- **Console Application:** ứng dụng 32 bits, với giao diện dạng DOS command line

Windows Application dựa C

Năm	Ngôn ngữ	Công cụ
1985	C	Windows application programming interface (API)
1992	C++	Microsoft Foundation Class (MFC) Library
2001	C# hoặc C++ .Net	Windows Form (1 phần của .NET Framework)

Các thư viện lập trình

- Windows API
- OWL - Object Windows Library
- MFC - Microsoft Foundation Classes
- ATL - Active Template Library
- WTL - Windows Template Library
- ...

Các khái niệm cơ bản

✓ Handle

§ Một giá trị 32 bits không dấu (unsigned) do HĐH tạo ra để làm định danh cho 1 đối tượng (cửa sổ, file, vùng nhớ, menu,...)

✓ ID (Identifier)

§ Một giá trị nguyên để làm định danh phân biệt giữa các 1 đối tượng (menu item, control) trong một chương trình.

✓ Instance

§ Một giá trị nguyên do HĐH tạo ra để định danh 1 thể hiện đang thực thi của ứng dụng

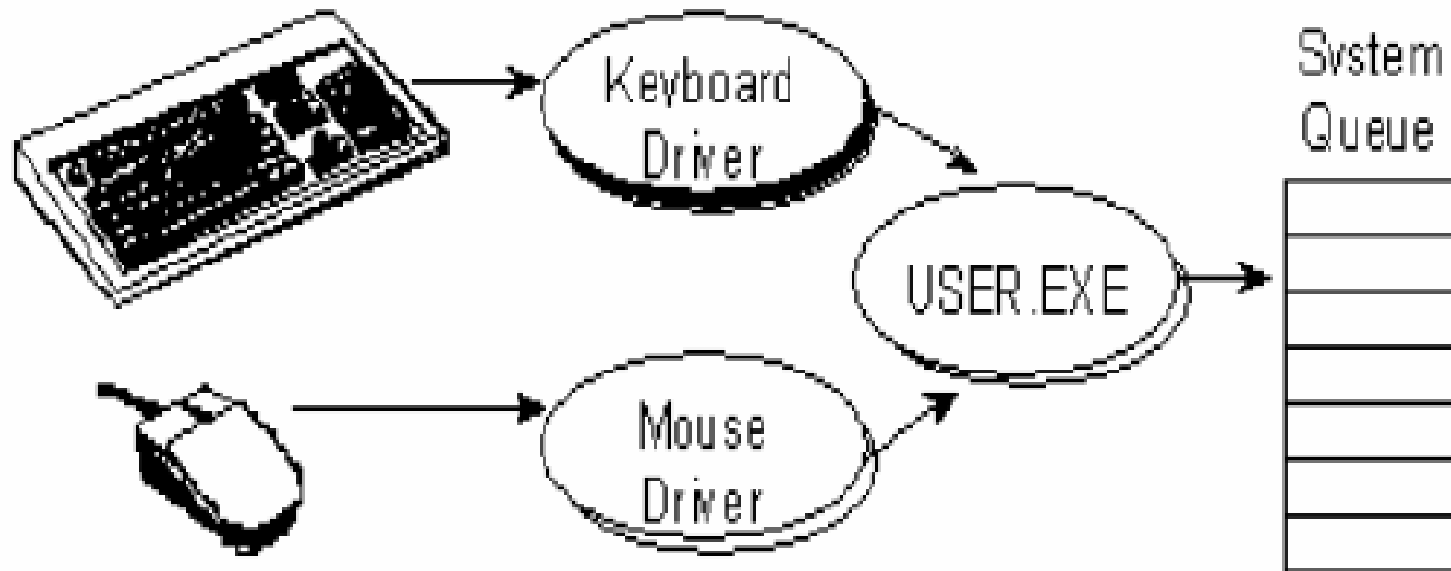
✓ Callback function

§ Các hàm được gọi tự động bởi hệ điều hành

Lập trình hướng sự kiện

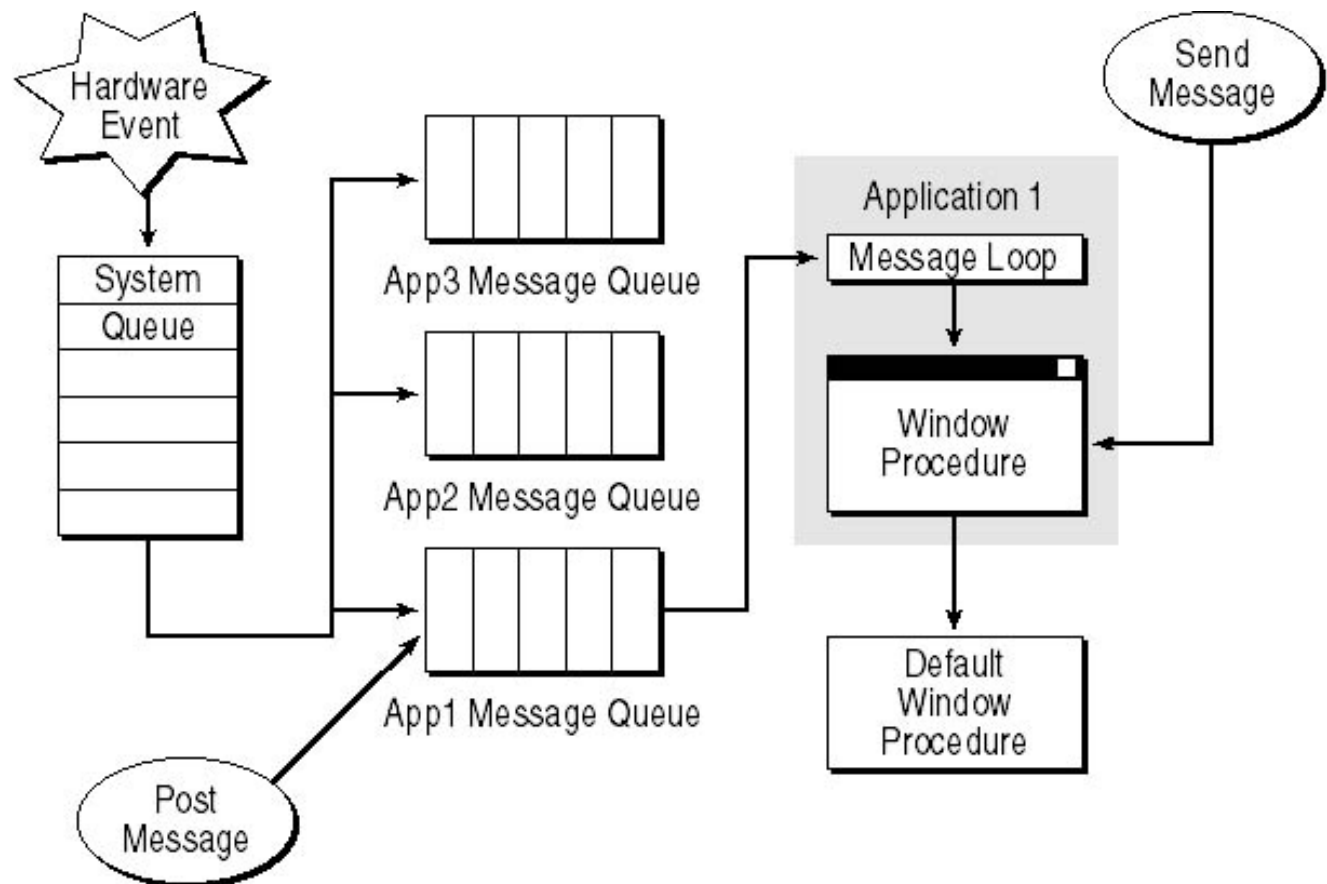
- ✓ Qui trình xử lý thông điệp
- ✓ Các loại thông điệp thường gặp
- ✓ Vòng lặp xử lý thông điệp

Phát sinh các sự kiện và thông điệp



Lập trình hướng sự kiện

- luồng điều khiển của chương trình được xác định dựa vào hành động của người dùng khi sử dụng chương trình



Các khái niệm cơ bản

Thông điệp	Được gửi khi
WM_CHAR	Có một ký tự được nhập từ bàn phím
WM_COMMAND	Người sử dụng click một mục trên menu, button...
WM_CREATE	Một cửa sổ được tạo ra
WM_DESTROY	Một cửa sổ bị huỷ.
WM_LBUTTONDOWN	Phím trái chuột được nhấn
WM_LBUTTONUP	Phím trái chuột được thả ra
WM_MOUSEMOVE	Con trỏ chuột di chuyển trên màn hình
WM_PAINT	Cửa sổ cần vẽ lại
WM_QUIT	Ứng dụng sắp kết thúc
WM_SIZE	Cửa sổ có thay đổi kích thước

Tài nguyên - Resource

- ✓ Tài nguyên (Menu bar, dialog, bitmap, icon, cursor,...) của ứng dụng được định nghĩa bên ngoài và được thêm vào trong file thực thi của ứng dụng khi biên dịch
- Các loại tài nguyên của ứng dụng
 - § Accelerator table: bảng mô tả phím tắt (hot-key)
 - § Bitmap: ảnh bitmap
 - § Caret: con trỏ văn bản
 - § Cursor: con trỏ chuột
 - § Dialog box: khung hộp thoại (Dialogbox Template)
 - § Font: font chữ

Tài nguyên - Resource

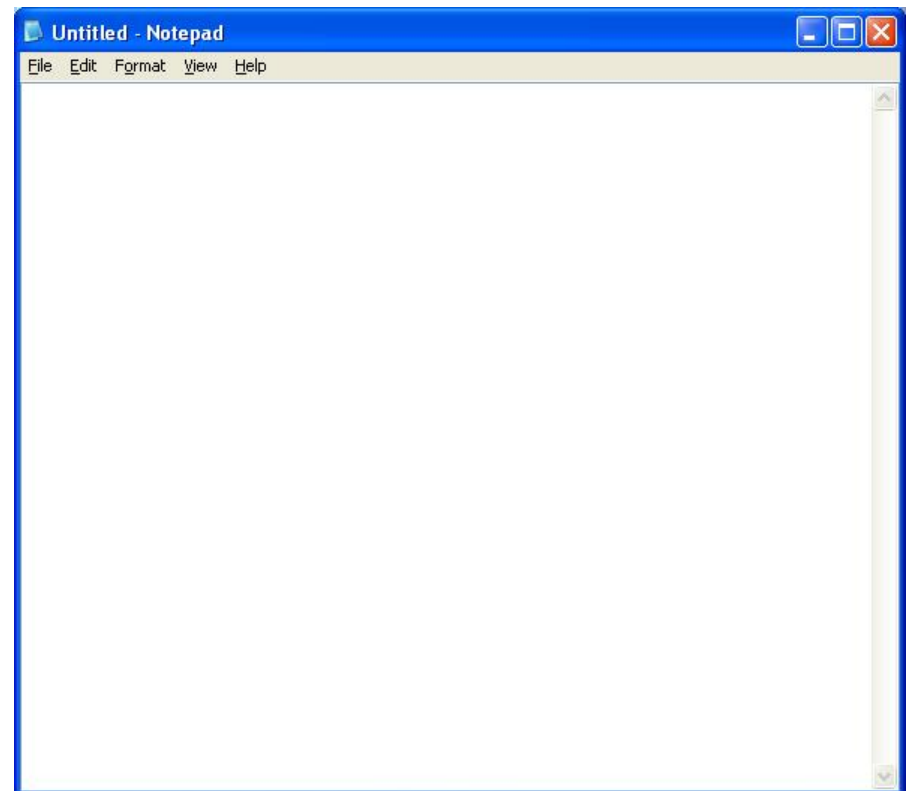
- Các loại tài nguyên của ứng dụng
 - § Icon: biểu tượng
 - § Menu: menu
 - § String-table entry: bảng mô tả các chuỗi ký tự
 - § Version information: bảng mô tả thông tin “phiên bản”

Giao diện đồ họa

- ✓ Các dạng ứng dụng giao diện đồ họa
 - § SDI - Single Document Interface
 - § MDI - Multiple Document Interface
 - § Dialog

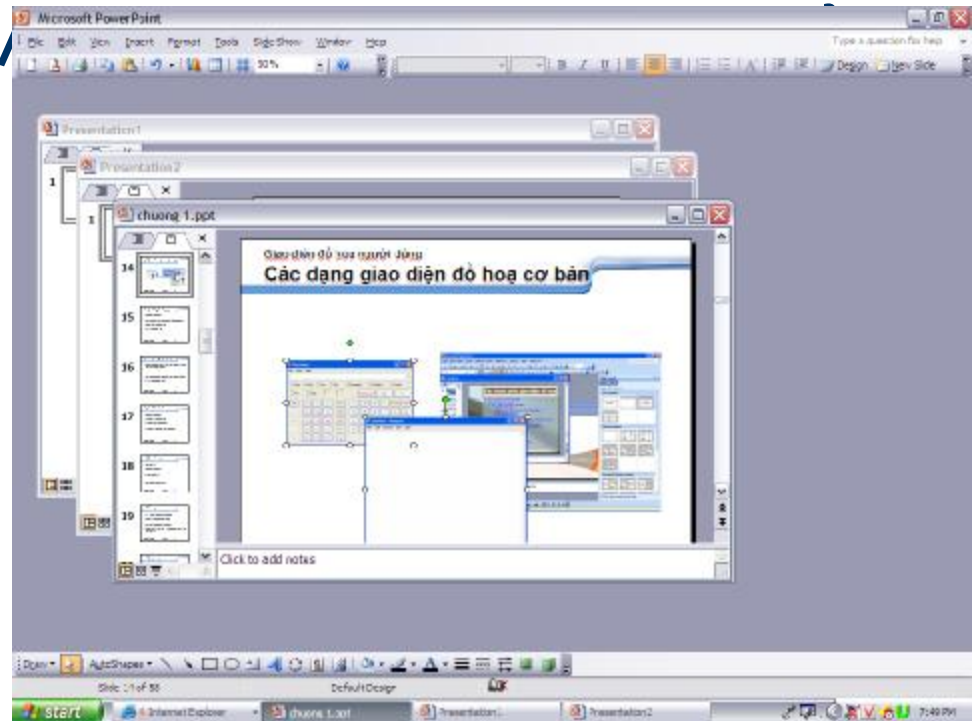
SDI - Single Document Interface

- ✓ Một cửa sổ làm việc
- ✓ Cho phép thay đổi kích thước cửa sổ (Resizable)
- ✓ Không có các cửa sổ con
- ✓ Ví dụ: Notepad, Paint,...



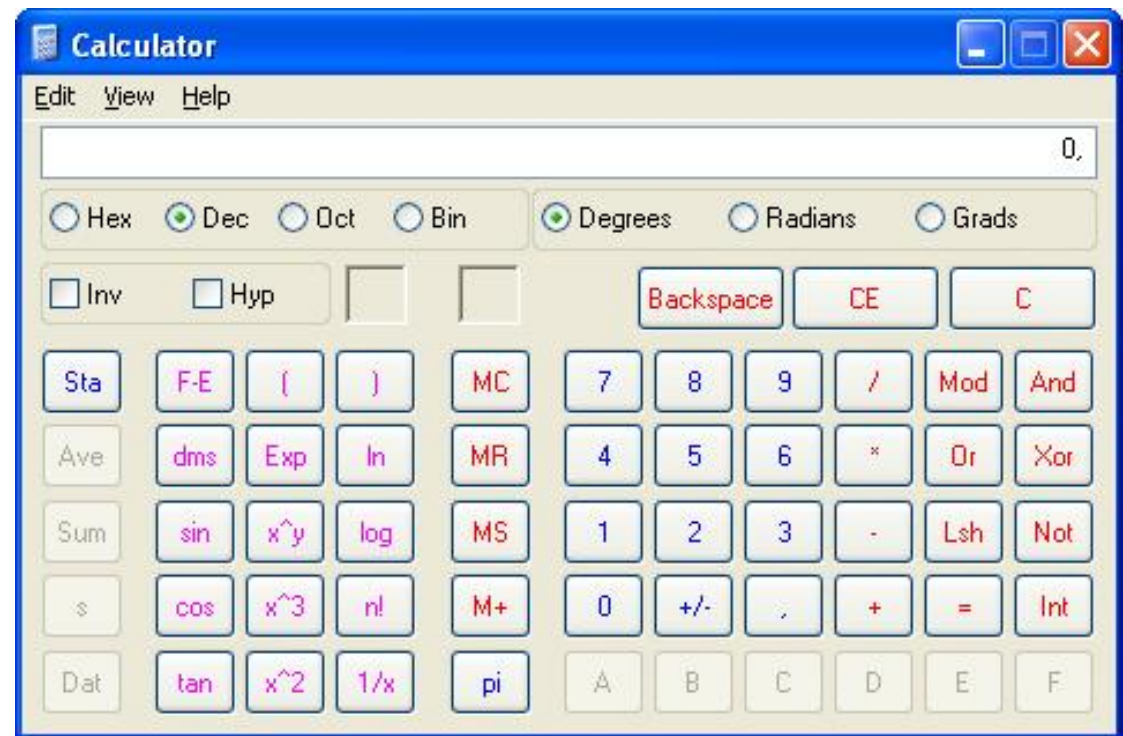
MDI - Multiple Document Interface

- ✓ Một cửa sổ làm việc chính và nhiều cửa sổ con
- ✓ Cho phép thay đổi kích thước cửa sổ (Resizable)
- ✓ Cho phép Maximize cửa sổ con
- ✓ Ví dụ: Word, Excel,



Dialog

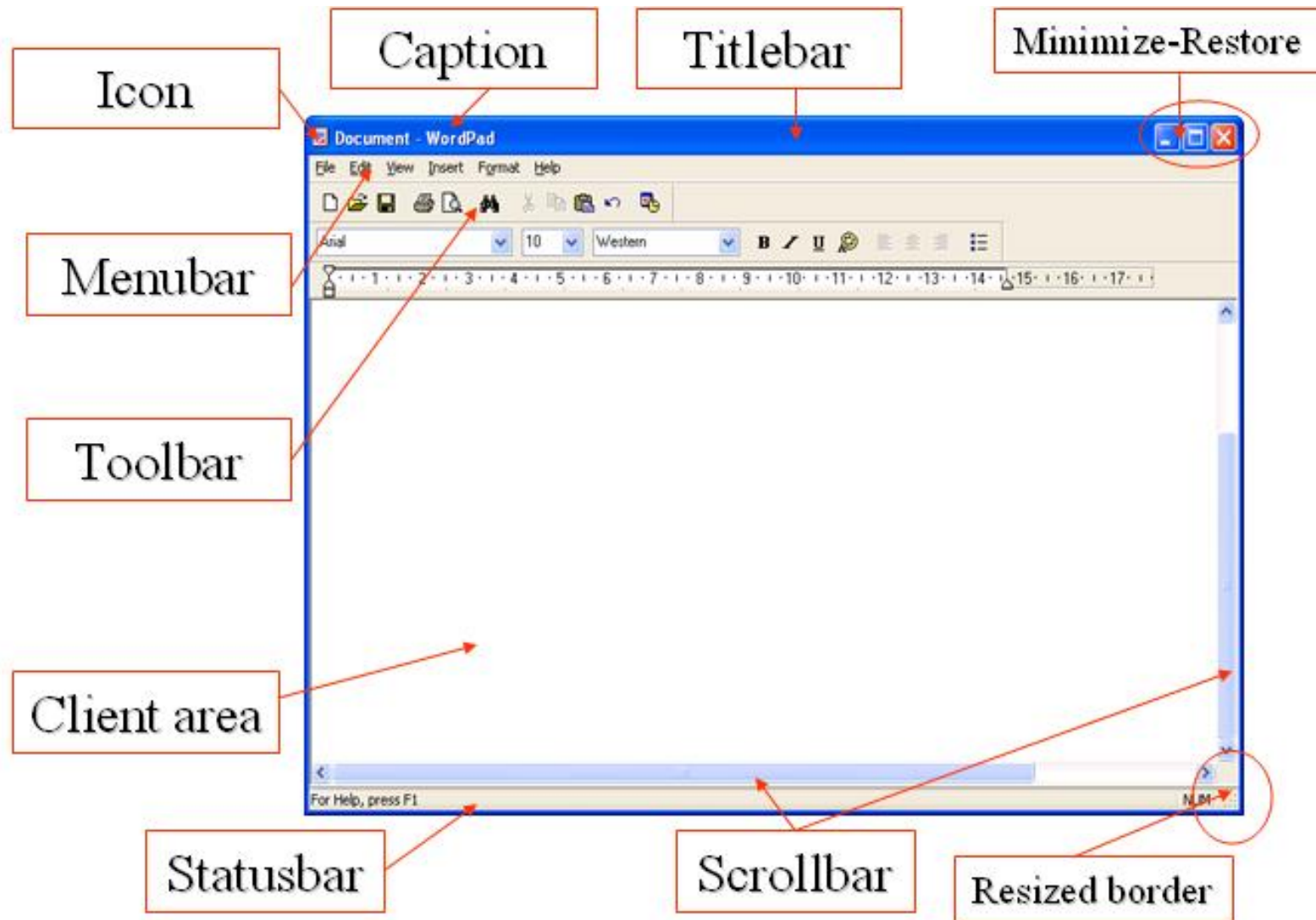
- ✓ Một cửa sổ làm việc
- ✓ Thường kích thước cố định
- ✓ Thường có các button, edit box, list box,...
- ✓ Ví dụ: Calculator, CD Player,...



Cửa sổ - Window

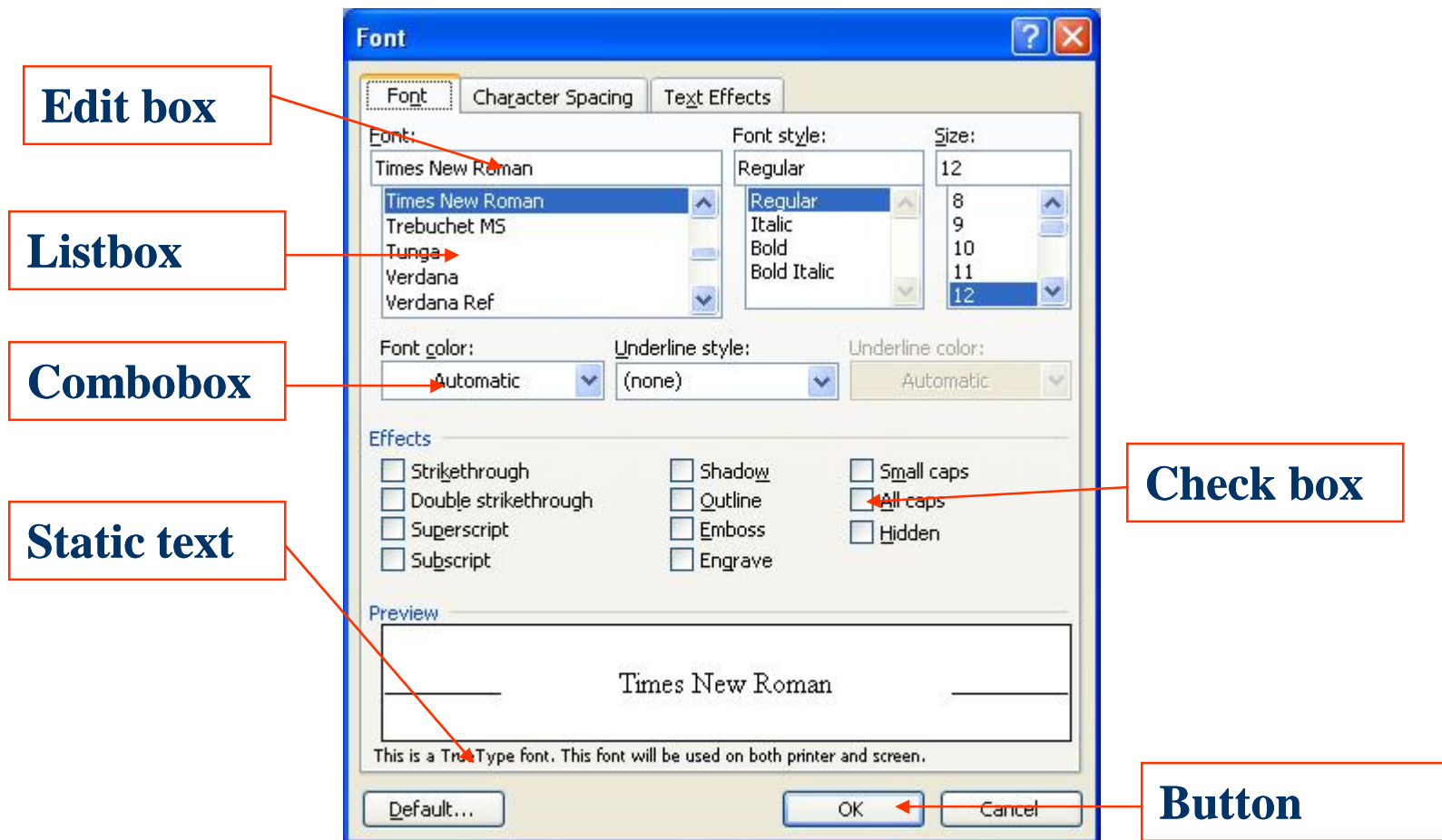
- ✓ Là một vùng chữ nhật trên màn hình dùng để hiển thị kết quả output và nhận các input từ người dùng
 - Công việc đầu tiên của 1 ứng dụng GUI là tạo một cửa sổ làm việc
- ✓ Mỗi cửa sổ đều có 1 cửa sổ cha (parent window), ngoại trừ cửa sổ nền Desktop
 - Tại mỗi thời điểm, chỉ có một cửa sổ nhận input từ người dùng (Active/Focused window)

Các thành phần của một cửa sổ

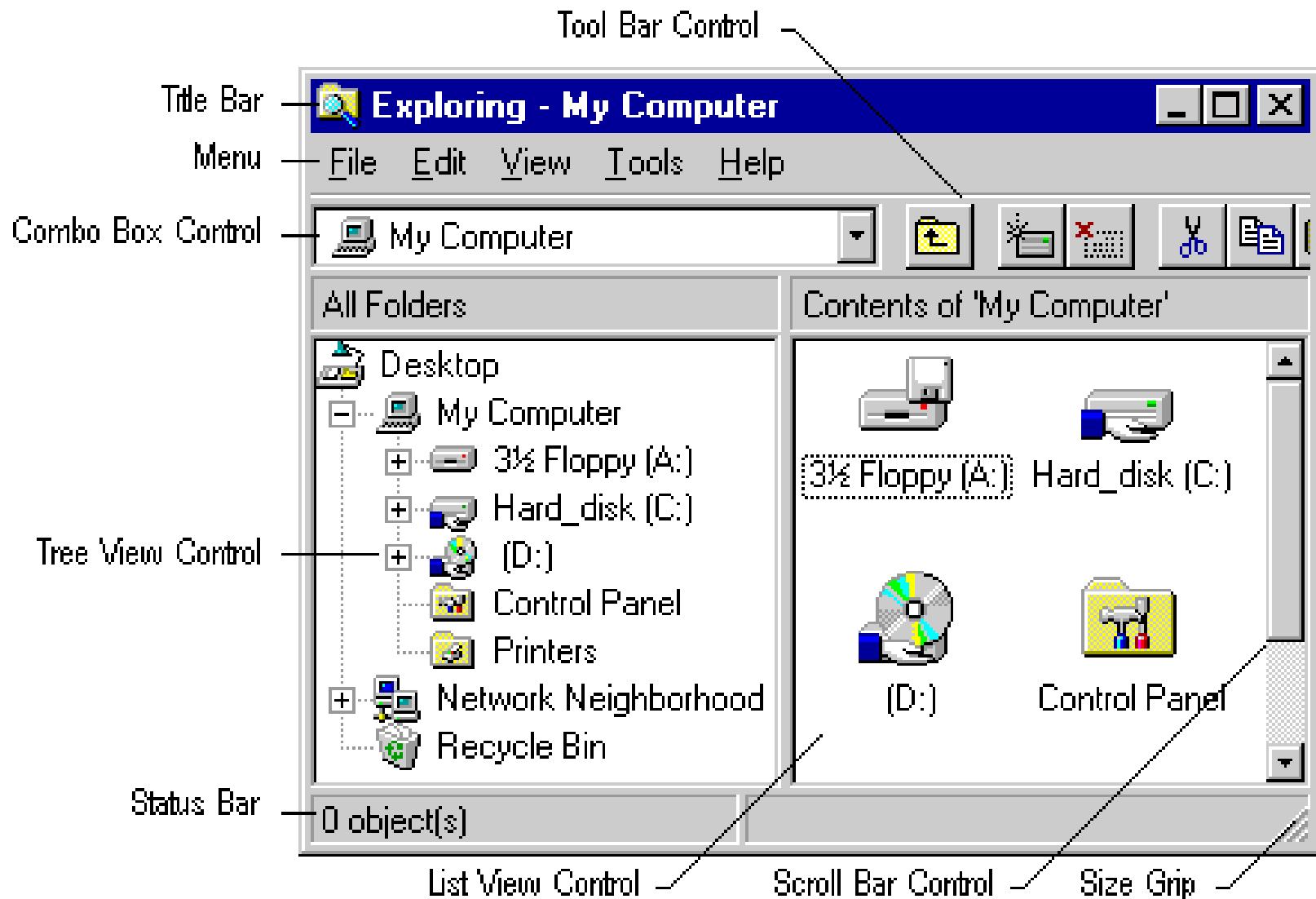


Các control thông dụng

Tab Control



Các control thông dụng



Qui ước đặt tên các hằng

✓ Dùng chữ hoa với tiếp đầu ngữ mô tả loại

CS	Class style
IDI	ID number (số nhận dạng) cho một biểu tượng
IDC	ID number (số nhận dạng) cho một cursor
WS	Window style
WM	Window message
CW	Create Window option
DT	DrawText option

CS_HREDRAW
CS_VREDRAW
CW_USEDEFAULT
DT_CENTER

DT_SINGLELINE
IDC_ARROW
DT_VCENTER
IDI_APPLICATION

WM_CREATE
WM_DESTROY
WM_PAINT
WS_OVERLAPPEDWINDOW

Giới thiệu .NET

✓ .NET bao gồm một bộ đầy đủ các công cụ giúp tạo và cài đặt các ứng dụng, .NET bao gồm:

§ Sản phẩm của .NET

- Visual studio.NET IDE
- Ngôn ngữ lập trình: C#, VB.NET...

§ Dịch vụ web của .NET

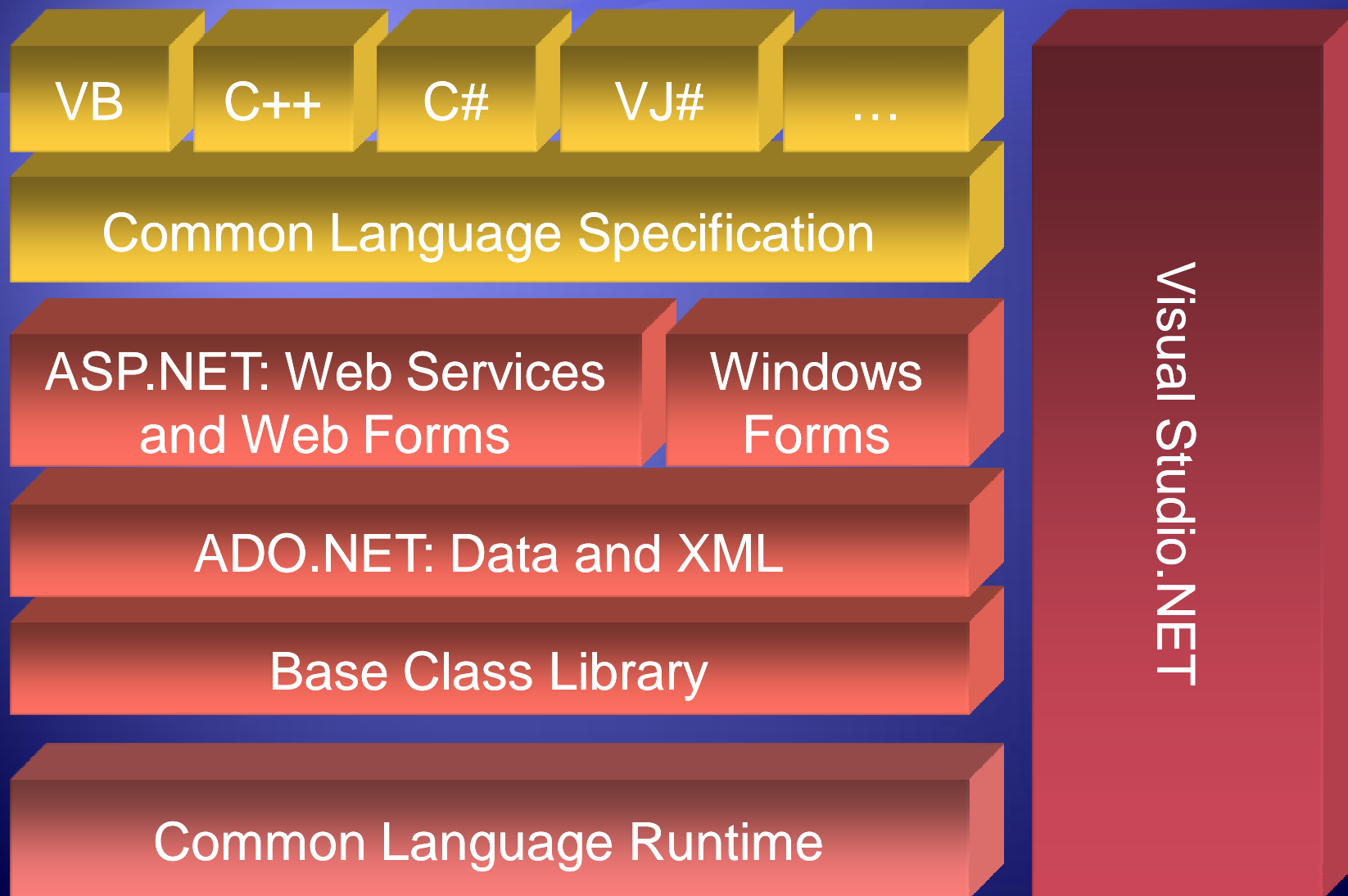
§ .NET Framework

.NET Framework

✓ .NET framework bao gồm 3 thành phần chủ yếu

- § Common Language Runtime (CLR)
- § Các lớp cơ sở của .NET Framework
- § Giao diện người dùng

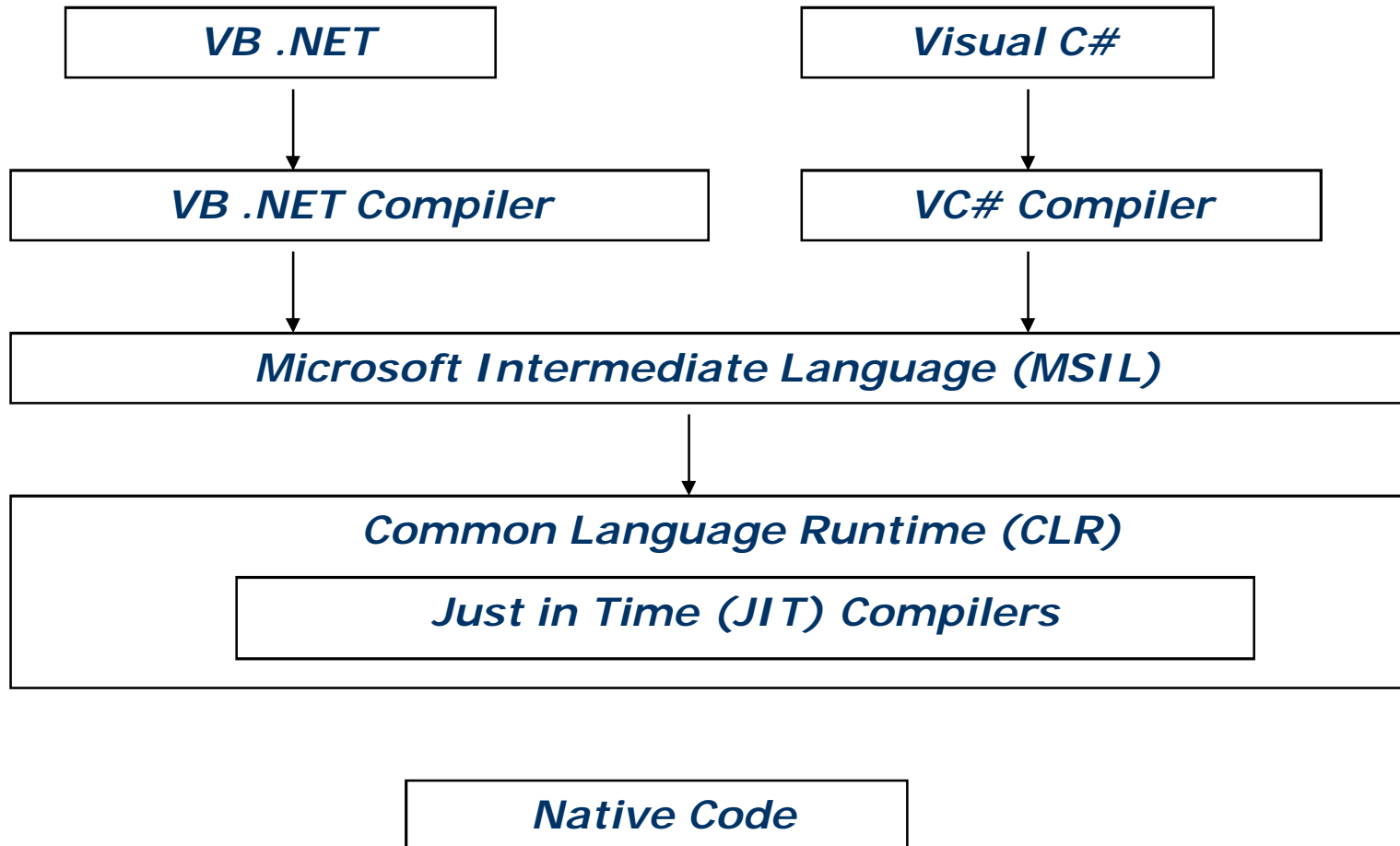
.NET Framework



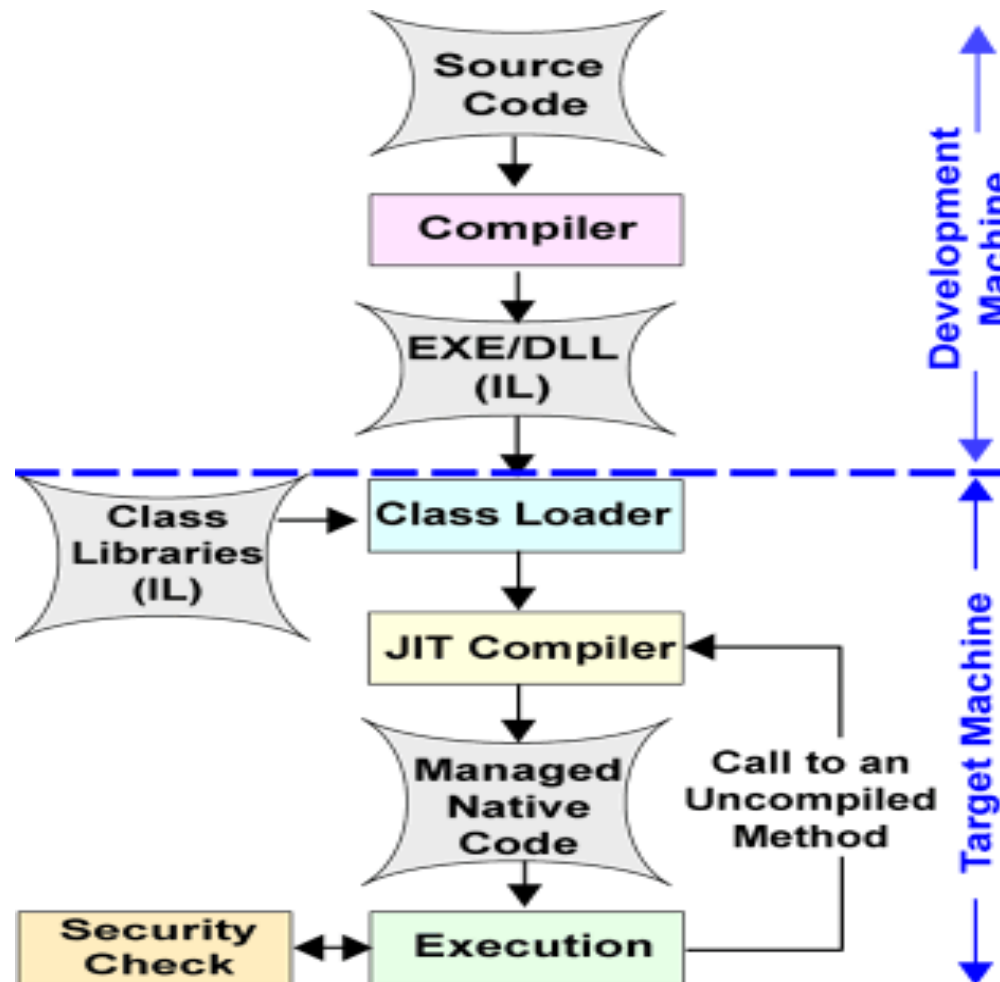
CLR

- ✓ **C**ommon **L**anguage **R**untime là môi trường để cho tất cả các ứng dụng viết trên .NET chạy
- ✓ CLR hỗ trợ nhiều ngôn ngữ và cung cấp các công cụ dùng chung cho các ngôn ngữ lập trình khác nhau, giúp cho việc tương tác qua lại giữa các ngôn ngữ lập trình khác nhau dễ dàng hơn

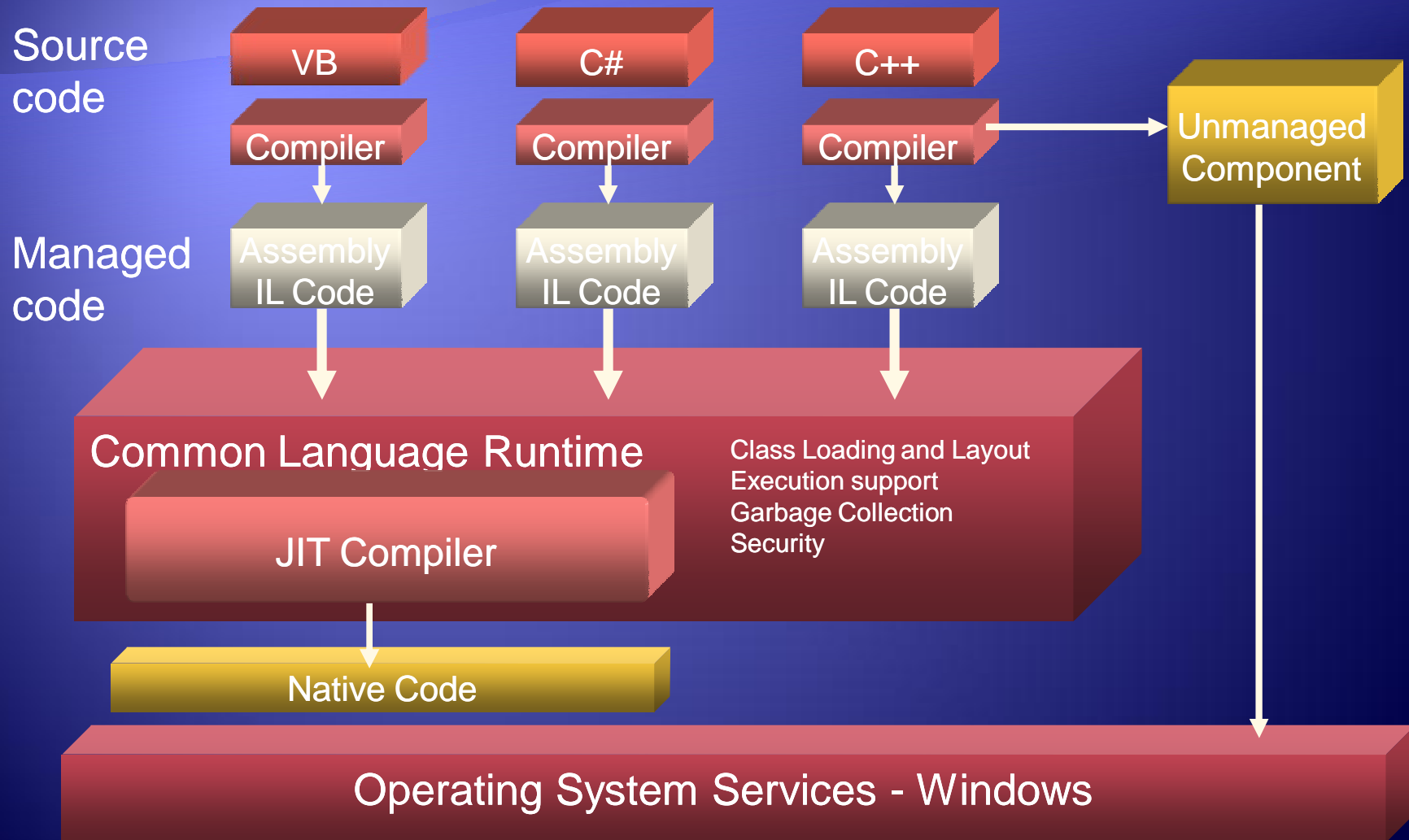
CLR



Quá trình biên dịch



CLR Execution Model



Các đặc điểm của CLR

- ✓ Tự động quản lý bộ nhớ
- ✓ Hệ thống kiểu dữ liệu dùng chung giữa các ngôn ngữ lập trình
- ✓ Tính tương tác giữa các ngôn ngữ
- ✓ Độc lập với cấu trúc phần cứng bên dưới
- ✓ Cơ chế bảo mật hệ thống
- ✓ An toàn kiểu dữ liệu

Các lớp cơ sở của .NET framework

- ✓ Được xây dựng theo phương pháp hướng đối tượng
- ✓ Cung cấp các lớp dùng để xử lý các vấn đề thường gặp phải khi phát triển ứng dụng
- ✓ Được dùng chung cho nhiều ngôn ngữ lập trình khác nhau
- ✓ Được tổ chức thành những namespace được lưu trữ trong assembly.

Các lớp cơ sở của .NET framework

- ✓ Namespace dùng để nhóm các lớp và các interface có mối quan hệ luận lý với nhau.
- ✓ Các namespace có thể được dùng trong bất cứ ngôn ngữ nào tương thích với .NET.
- ✓ Namespace được sử dụng để hạn chế việc dùng cùng một tên cho 2 hay nhiều lớp có mục đích sử dụng khác nhau

Các lớp cơ sở của .NET framework

- ✓ Assembly là một đơn vị phần mềm có chứa đầy đủ các thông tin về các lớp hiện thực, các cấu trúc và các interface để hiện thực ứng dụng
- ✓ Assembly lưu trữ các thông tin để mô tả chính nó, thông tin này gọi là meta data

Các lớp cơ sở của .NET framework

- ✓ System
- ✓ System.Collections
- ✓ System.Drawing
- ✓ System.IO
- ✓ System.Data
- ✓ System.Windows.Forms
- ✓ System.Web.UI
- ✓ System.Web.Services
- ✓

Giao diện người dùng

- Window form: dùng để tạo ứng dụng cục bộ có giao diện cửa sổ.
- ✓ Web form: dùng để tạo ứng dụng web
- ✓ Console: Dùng để tạo ứng dụng được tương tác bằng lệnh giống như giao diện Ms Dos

Ưu điểm của .NET Framework

- ✓ Dùng chung cho nhiều ngôn ngữ lập trình
- ✓ Tạo ứng dụng độc lập với môi trường phần cứng bên dưới
- ✓ Cho phép viết ứng dụng bằng nhiều ngôn ngữ tương thích với .NET
- ✓ Tự động quản lý tài nguyên
- ✓ Dễ dàng triển khai ứng dụng

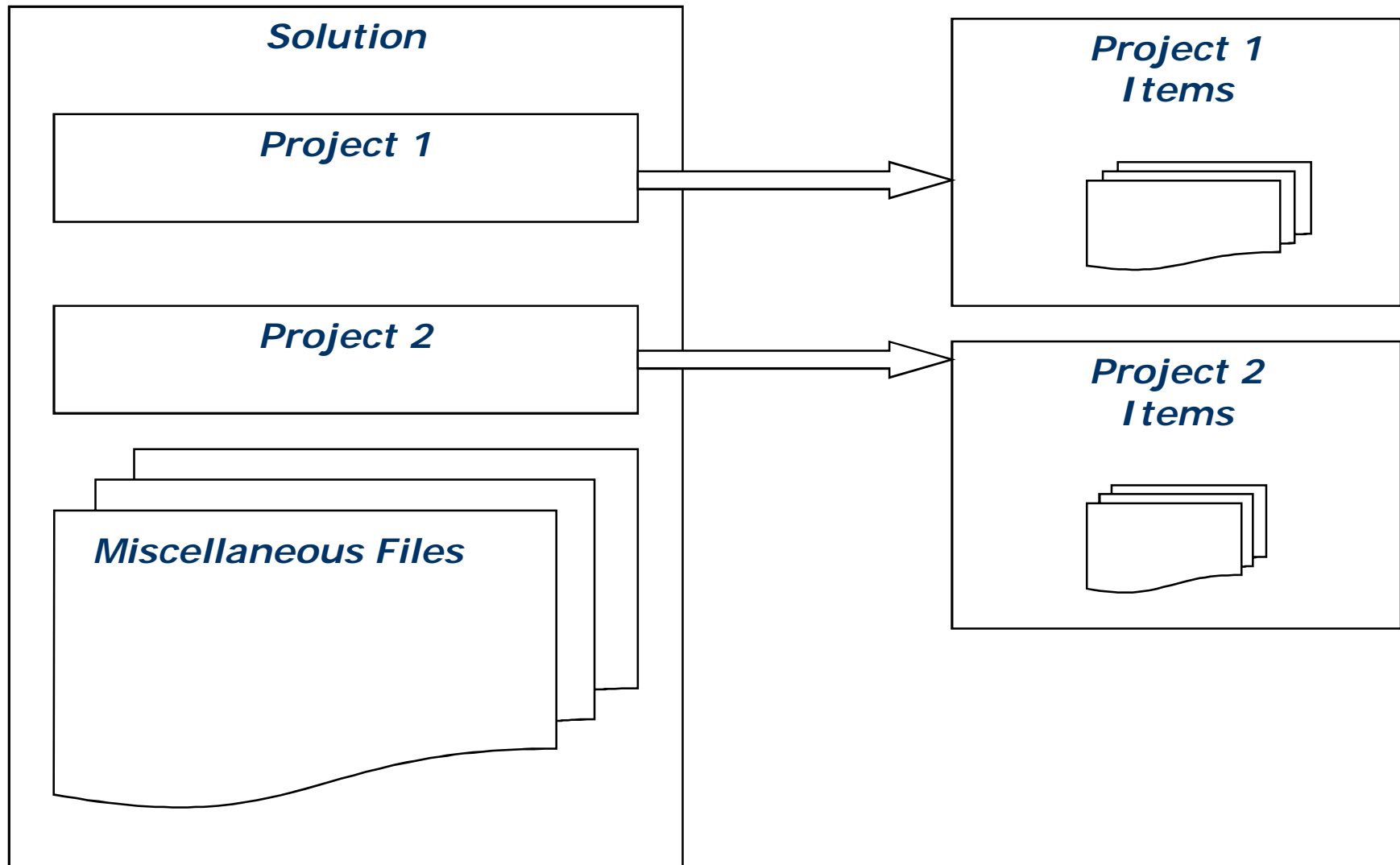
Visual studio.NET IDE

- ✓ Visual Studio .NET **I**ntegrated **D**evelopment **E**nvironment (IDE) cung cấp một giao diện dùng chung để phát triển nhiều loại dự án khác nhau.
- ✓ Cung cấp môi trường để thiết kế, viết code, kiểm tra và sửa lỗi khi phát triển ứng dụng
- ✓ Các thành phần của Visual Studio .NET IDE
 - § Dự án và giải pháp
 - § Giao diện của các thành phần

Dự án và giải pháp

- ✓ Trong visual studio .NET ứng dụng được tạo thành từ nhiều phần tử ví dụ như file, thư mục
- ✓ Để tổ chức các phần tử trên .NET cung cấp dự án và giải pháp
- ✓ Dự án (project): cho phép xây dựng, dịch và sửa lỗi các thành phần tạo nên ứng dụng. Dự án được dịch ra thành file exe hoặc dll
- ✓ Giải pháp (Solution): Bao gồm một hay nhiều project

Dự án và project



Q & A

LOGO

Thank You !

www.themegallery.com