

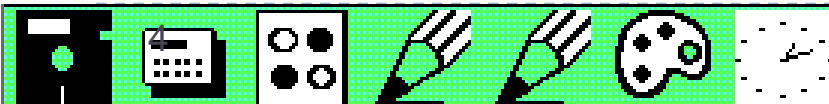
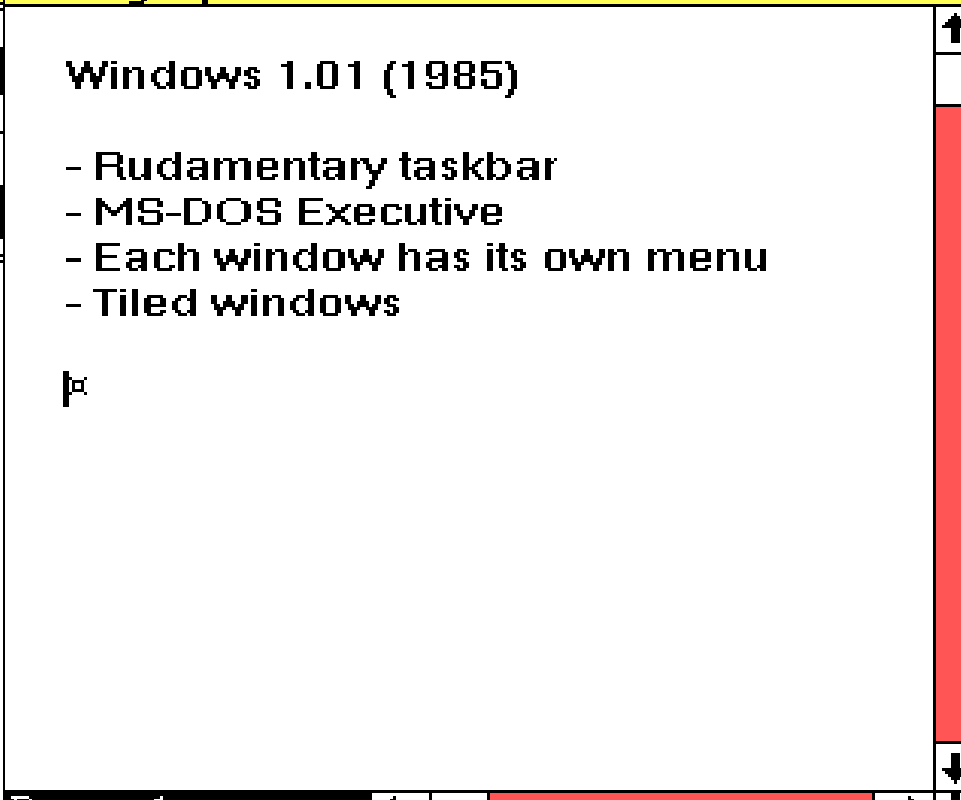
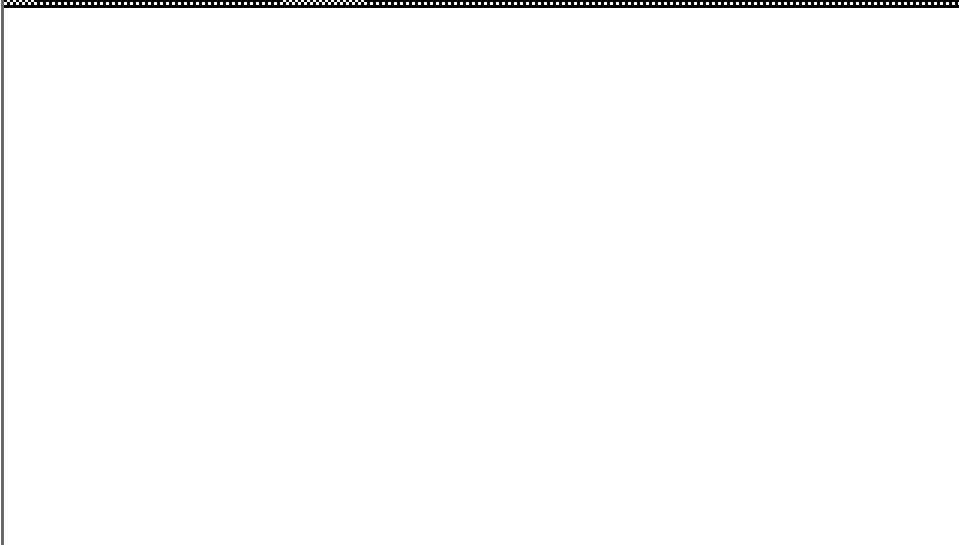
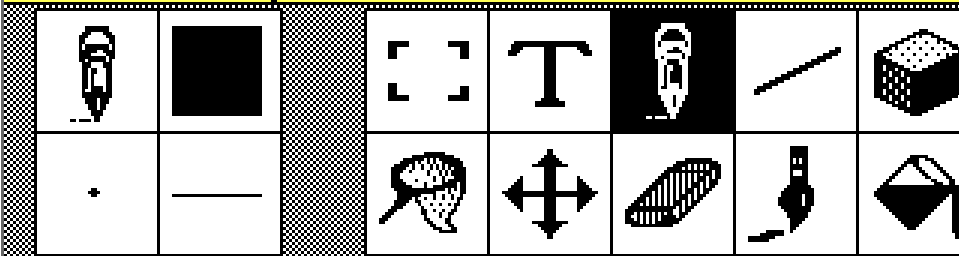
# **Tổng quan lập trình môi trường Windows**

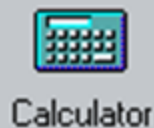
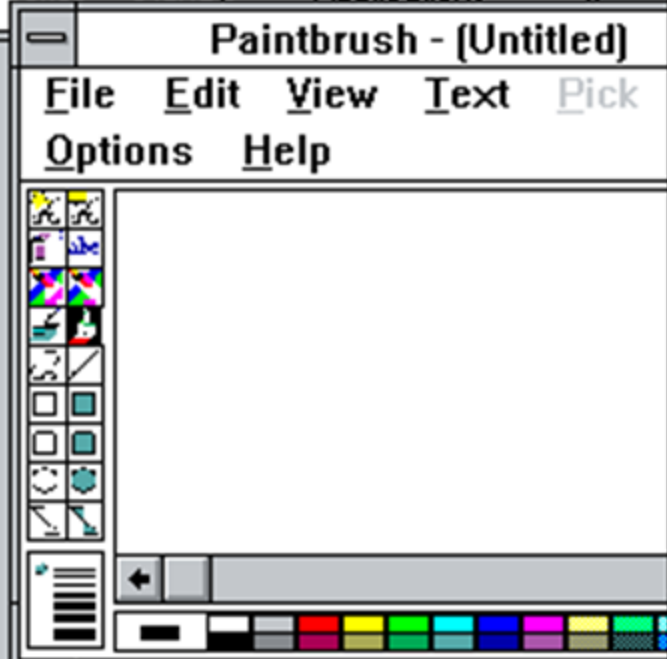
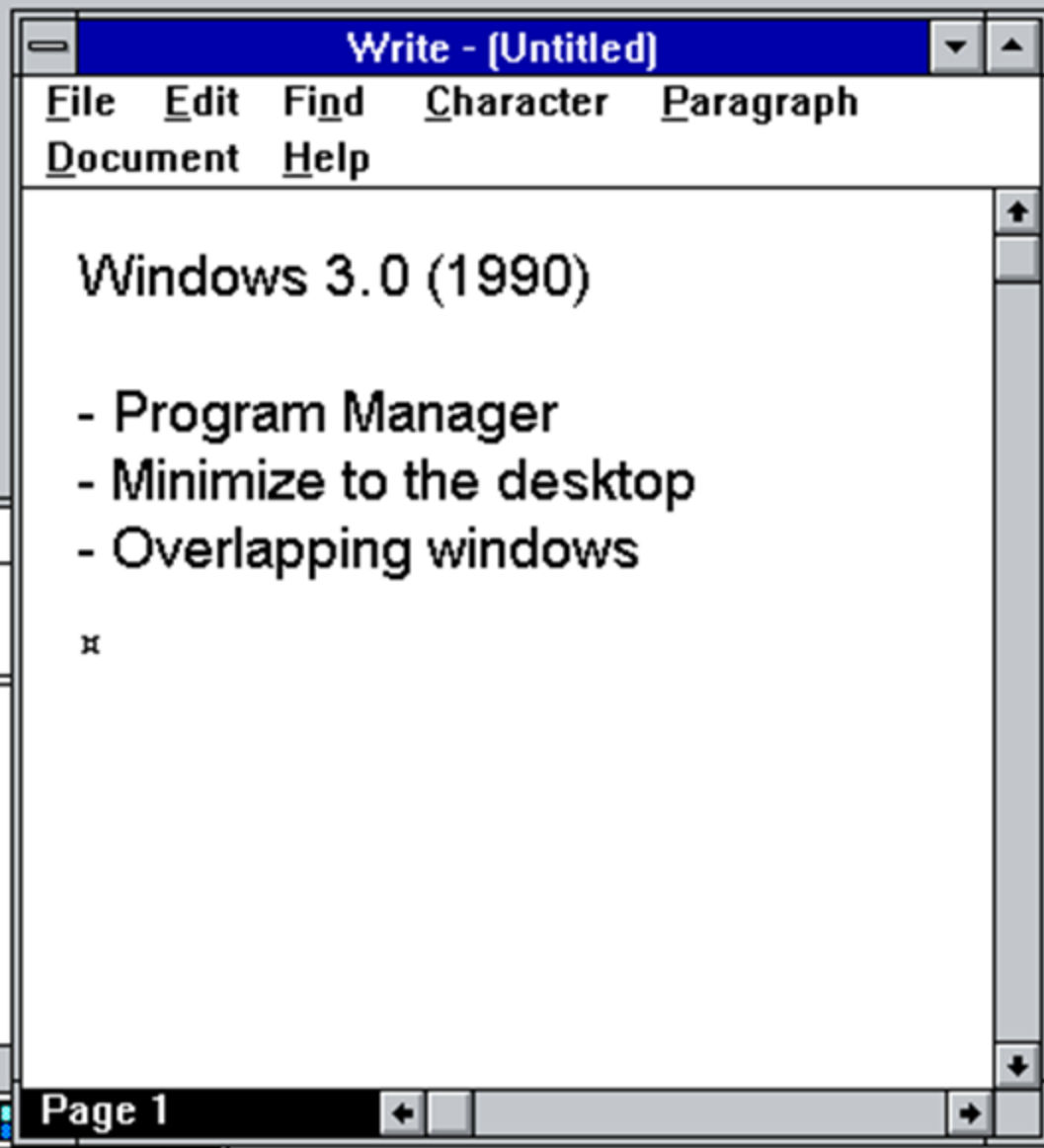
# Lịch sử Windows

	1990	1993	1994	1995	1996	1997
<b>Windows Desktop</b>	Windows 3x	Windows NT		Windows 95	Windows NT Workstation	
<b>Windows Server</b>		Windows NT Advanced Server 3.1	Windows NT Server 3.5	Windows NT Server 3.51	Windows NT Server 4.0	Windows NT Server Enterprise Edition
<b>Windows Cloud</b>						

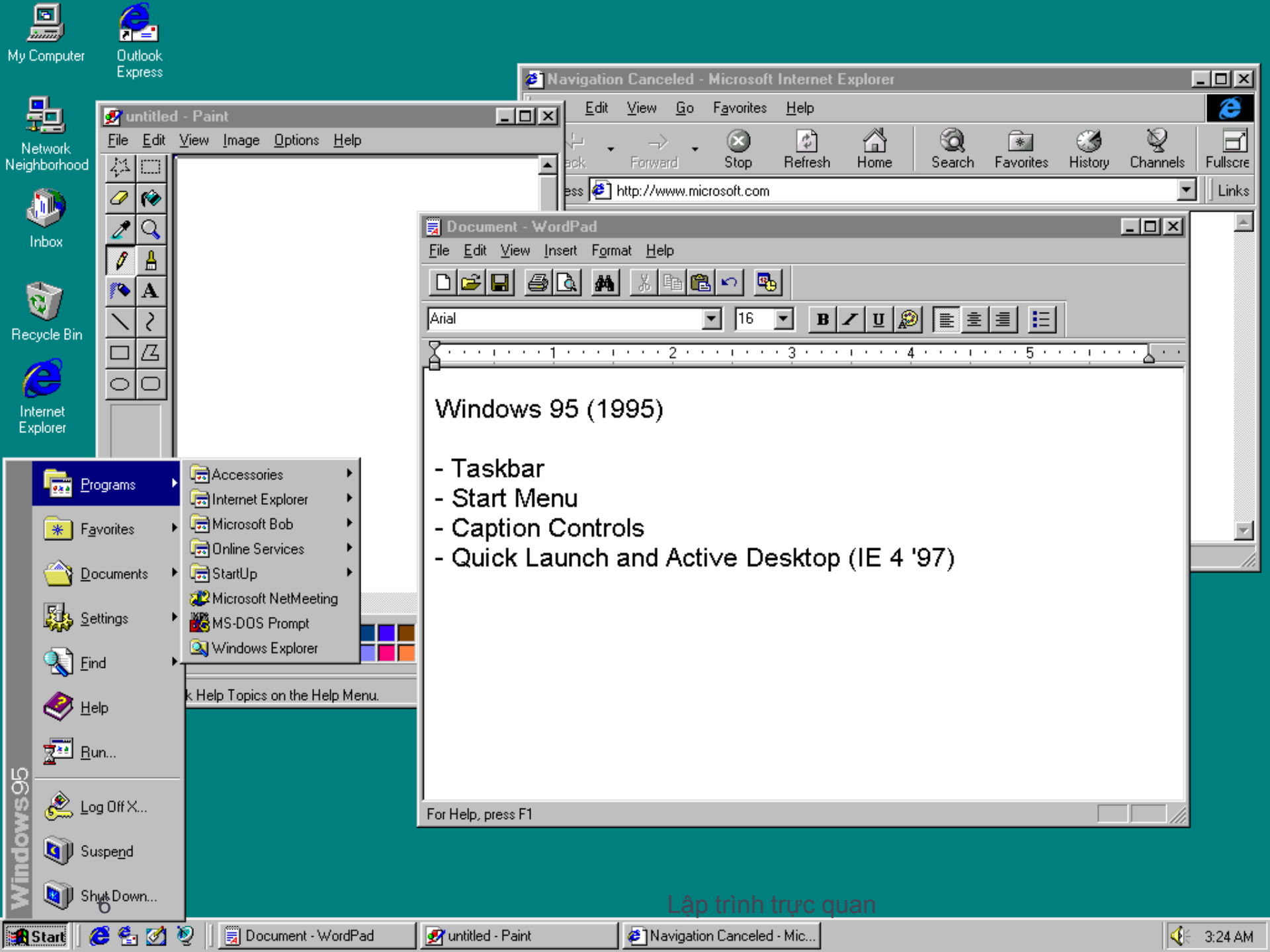
# Lịch sử Windows

	1998	2000	2001	2003	2006	2008	2009	2012
<b>Windows Desktop</b>	Windows 98	1. Windows ME 2. Windows 2000 Professional	Windows XP		Windows Vista		Windows 7	Windows 8
<b>Windows Server</b>	Windows NT Server 4.0 Terminal Server Edition	Windows 2000 Server Family		1. Windows Server 2003 2. Windows Server Systems		Windows Server 2008		
<b>Windows Cloud</b>						Windows Azure		



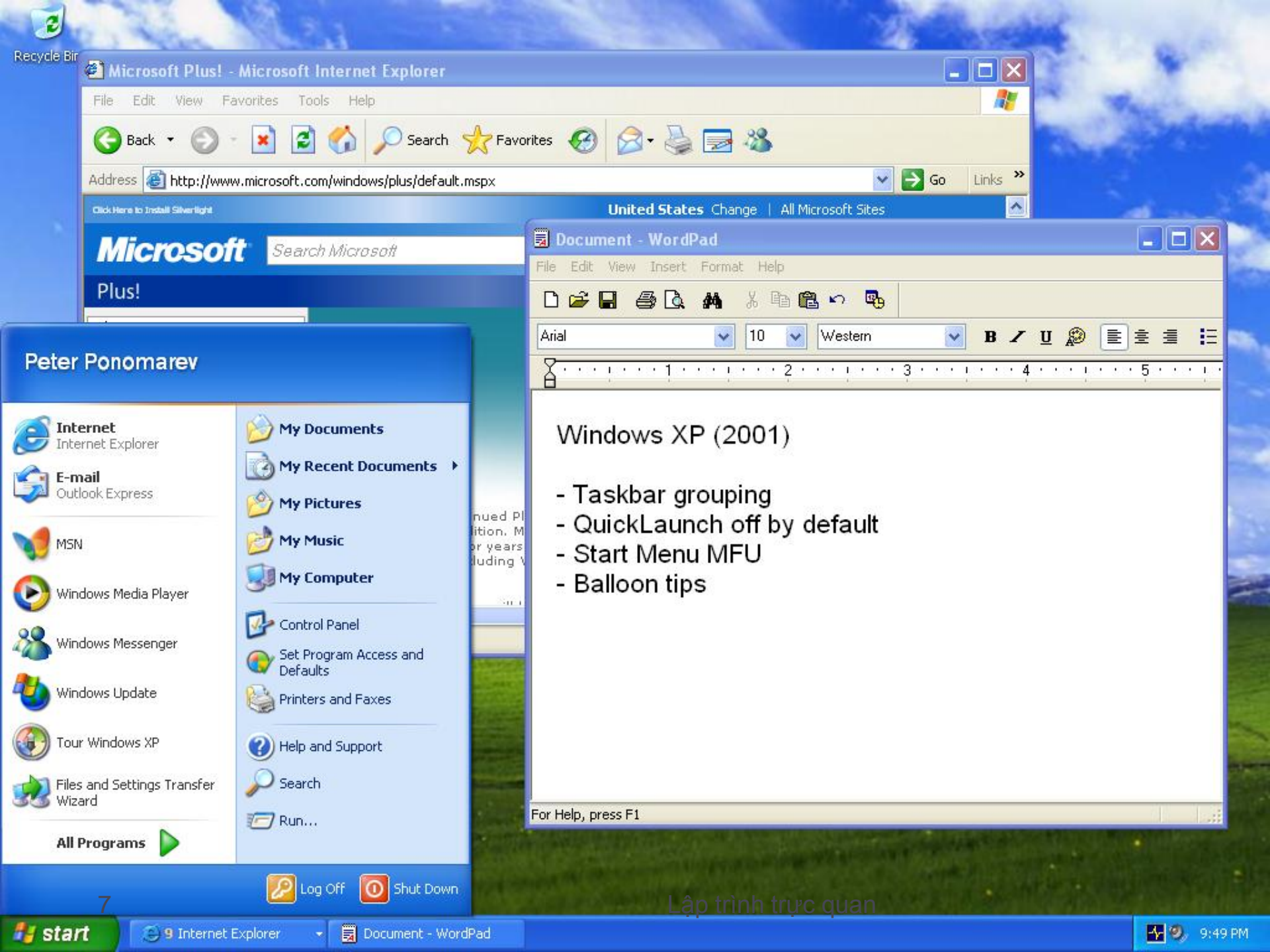


Lập trình trực quan



## Windows 95 (1995)

- Taskbar
- Start Menu
- Caption Controls
- Quick Launch and Active Desktop (IE 4 '97)



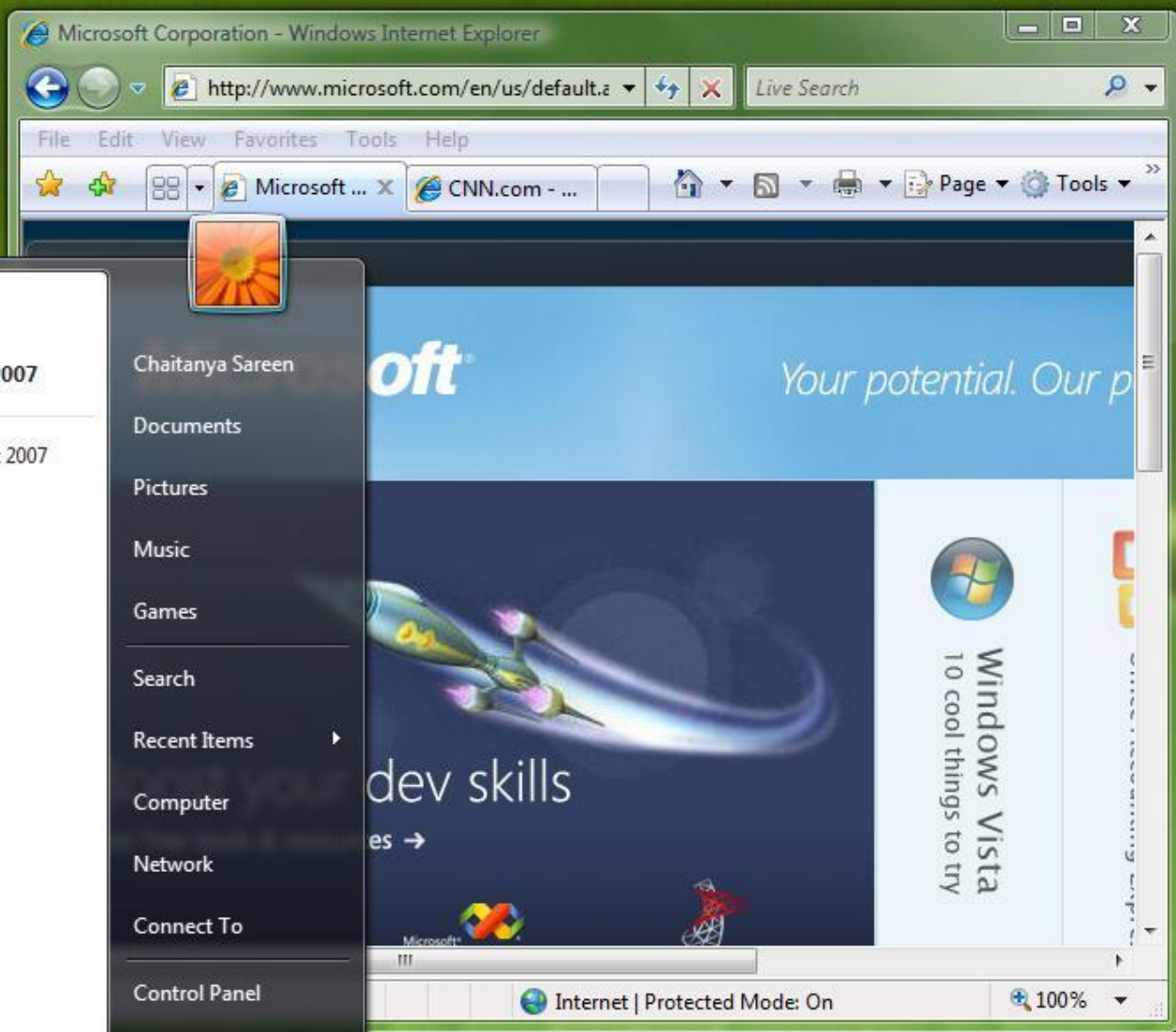
Peter Ponomarev

## Windows XP (2001)

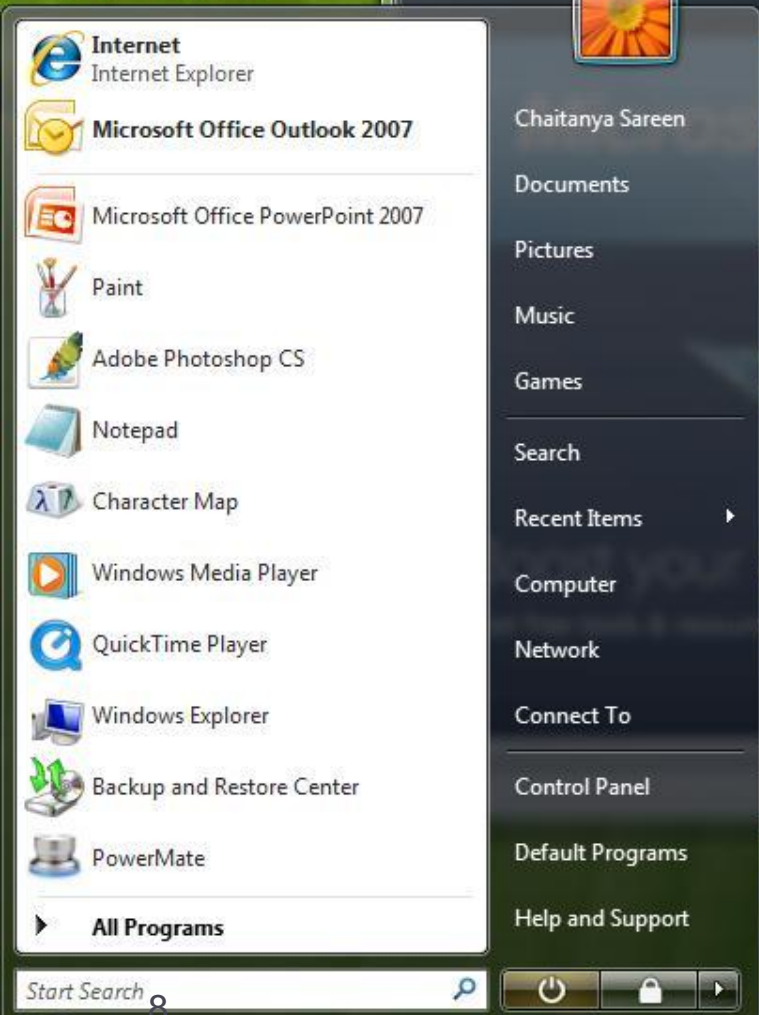
- Taskbar grouping
- QuickLaunch off by default
- Start Menu MFU
- Balloon tips

Lập trình trực quan





How many Outlook icons do you see?





# Start

Paul  
Thurrott



# Môi trường Windows

---

- ▶ Hệ điều hành 32/64 bit
- ▶ Giao diện người dùng kiểu đồ hoạ (GUI)
- ▶ Đa nhiệm
- ▶ Quản lý bộ nhớ
- ▶ Tư tưởng hướng đối tượng
- ▶ Giao diện đồ hoạ độc lập thiết bị
- ▶ Kiến trúc hướng sự kiện, dựa vào thông điệp

# Giao diện người dùng kiểu đồ hoạ (GUI)

---

- ▶ Visual Interface
- ▶ WYSIWYG
- ▶ Các ứng dụng Windows có giao diện thống nhất với cùng dáng vẽ:
  - ▶ 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 Windows 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

# 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 dùng ngôn ngữ 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
- BCL - .NET Framework Base Class Library
- ...

# 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

# Giao diện đồ họa

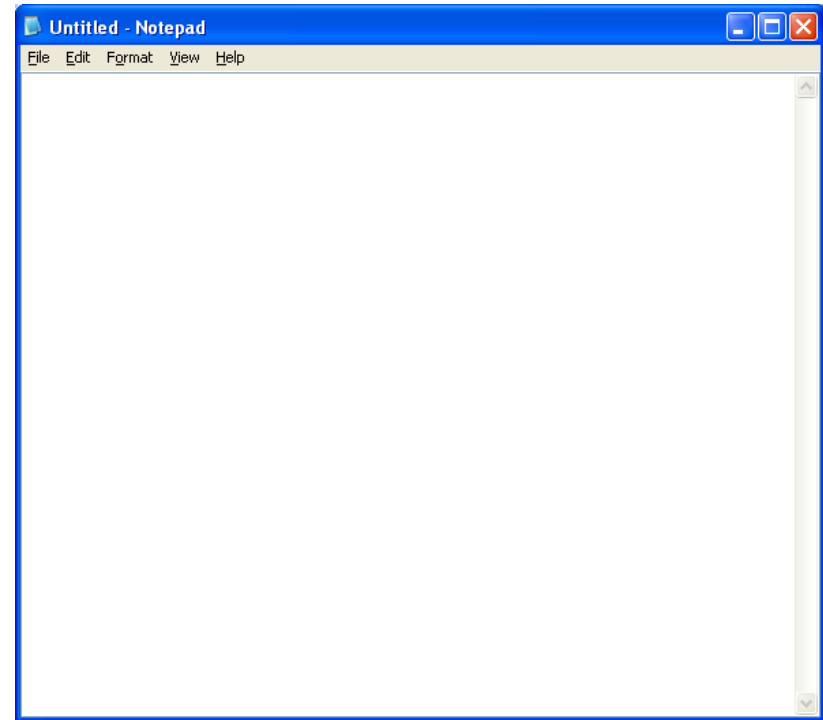
---

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

# SDI - Single Document Interface

---

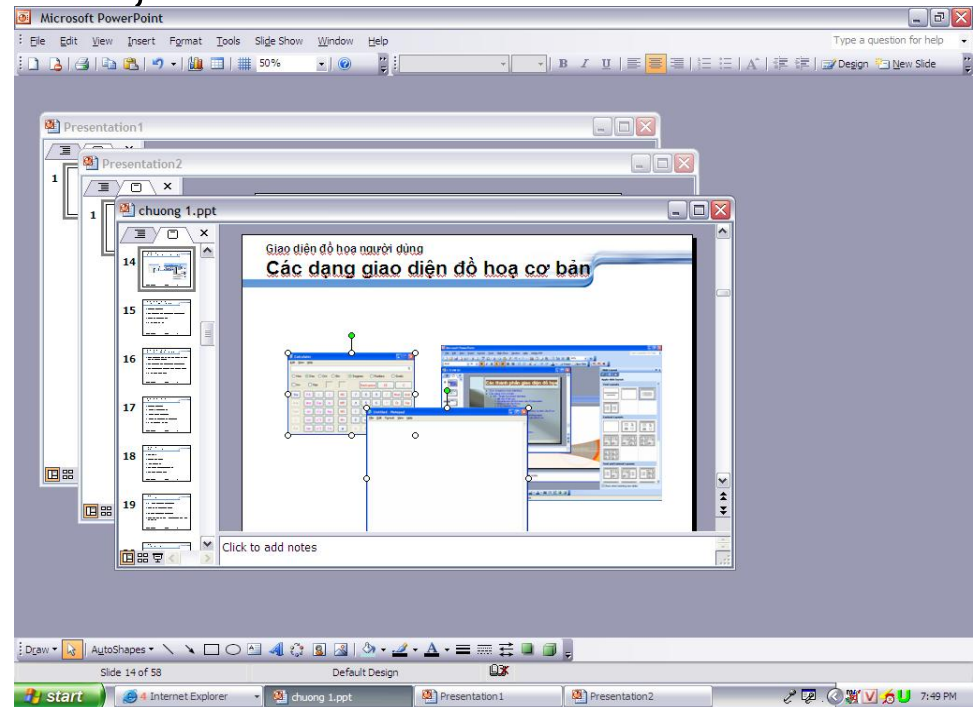
- ▶ Một cửa sổ làm việc
- ▶ Cho phép thay đổi kích thước cửa sổ (Resizable)
- ▶ Không có các cửa sổ con
- ▶ Ví dụ: Notepad, Paint,...





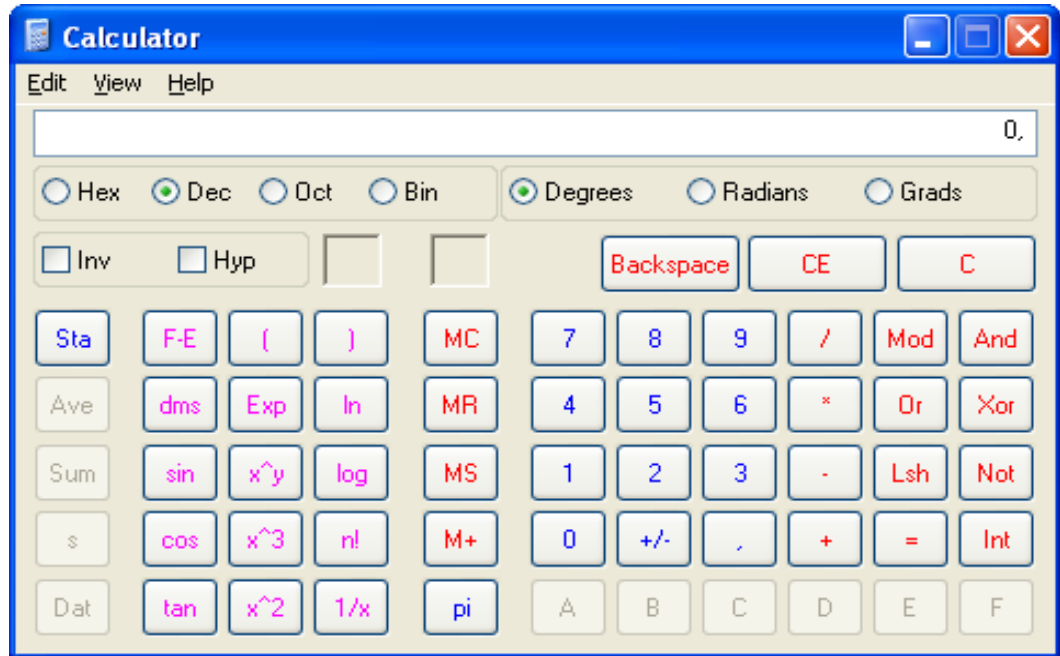
# MDI - Multiple Document Interface

- ▶ Một cửa sổ làm việc chính và nhiều cửa sổ con
- ▶ Cho phép thay đổi kích thước cửa sổ (Resizable)
- ▶ Cho phép Maximize/Minimize/Close các cửa sổ con
- ▶ Ví dụ: Word, Excel, VC++, ...



# Dialog

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

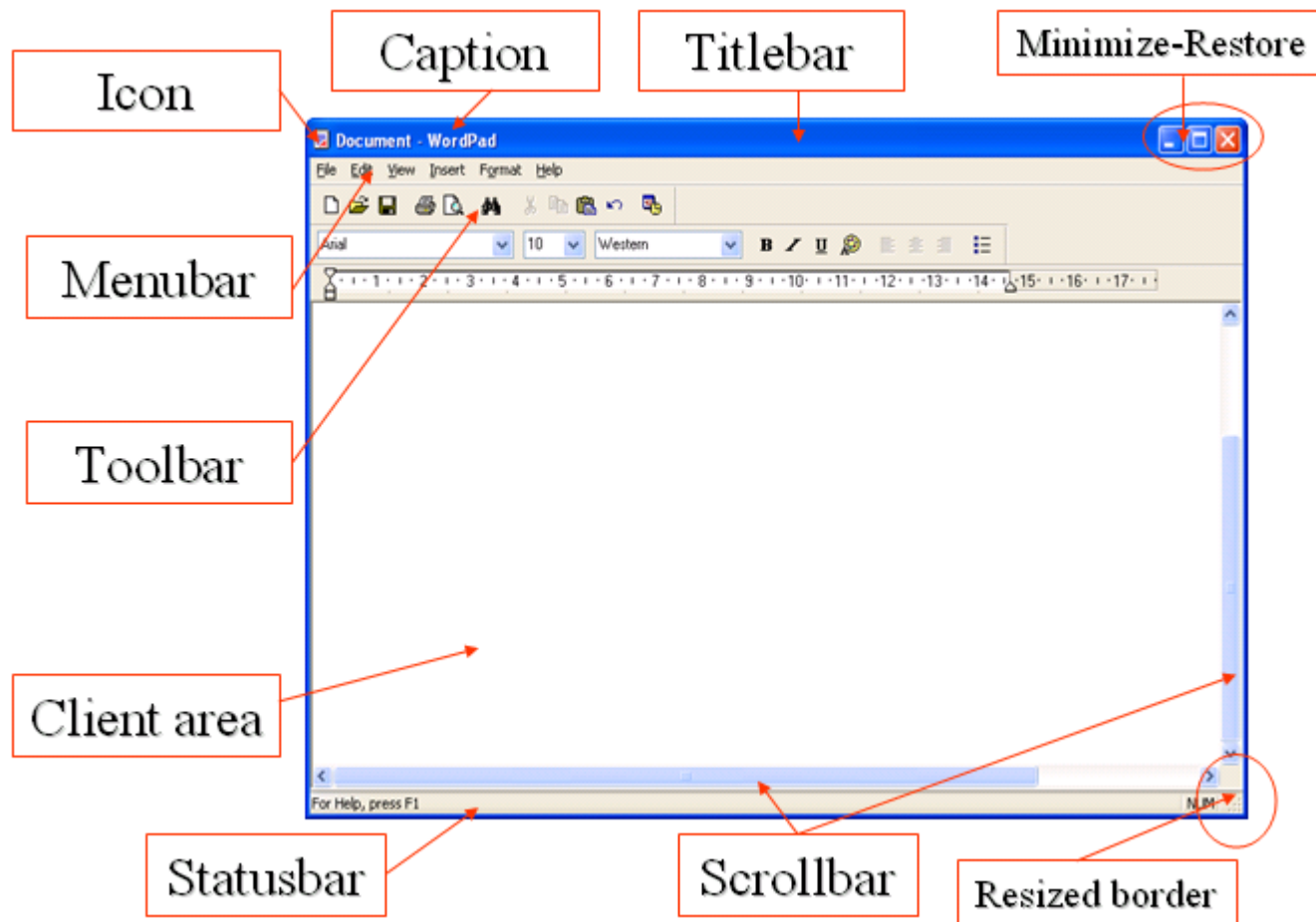


# Cửa sổ - Window

---

- ▶ 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

Edit box

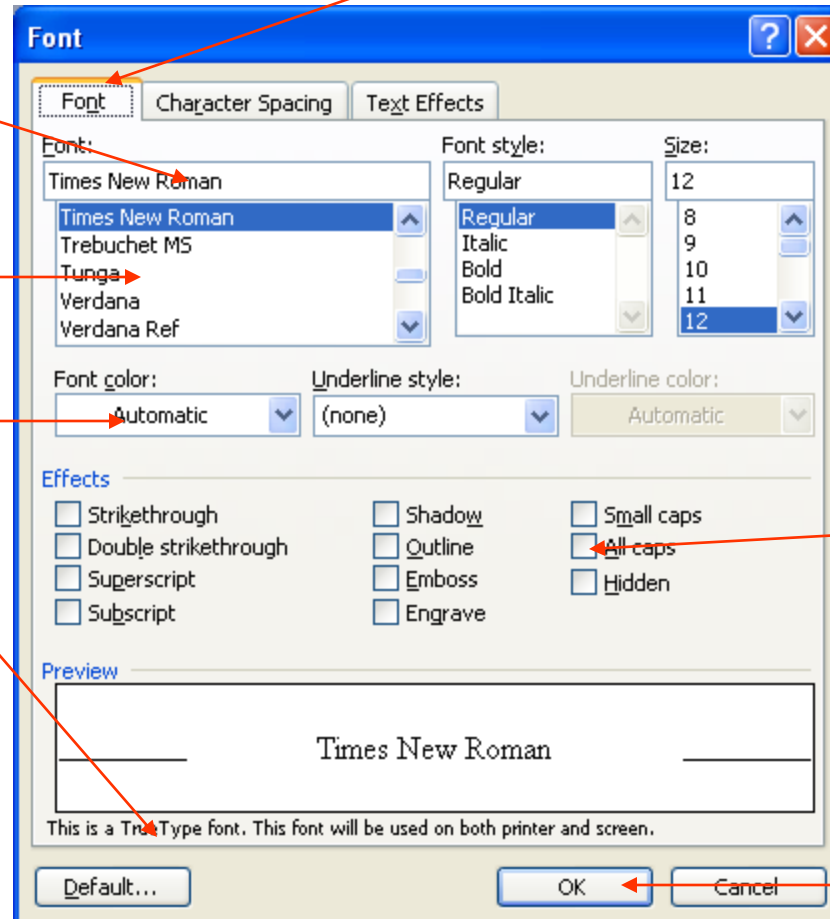
Listbox

Combobox

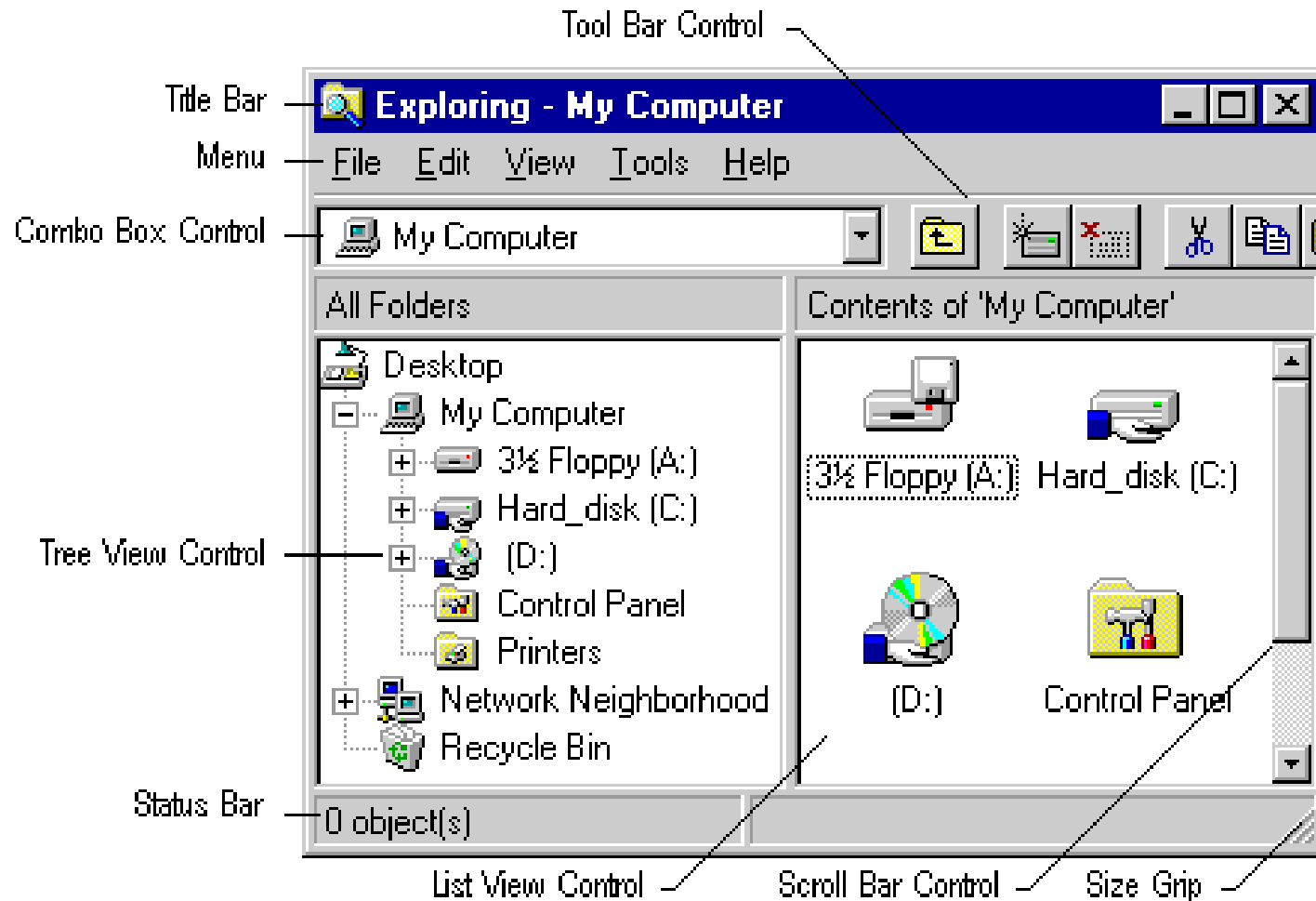
Static text

Check box

Button



# Các control thông dụng



# Lập trình hướng sự kiện

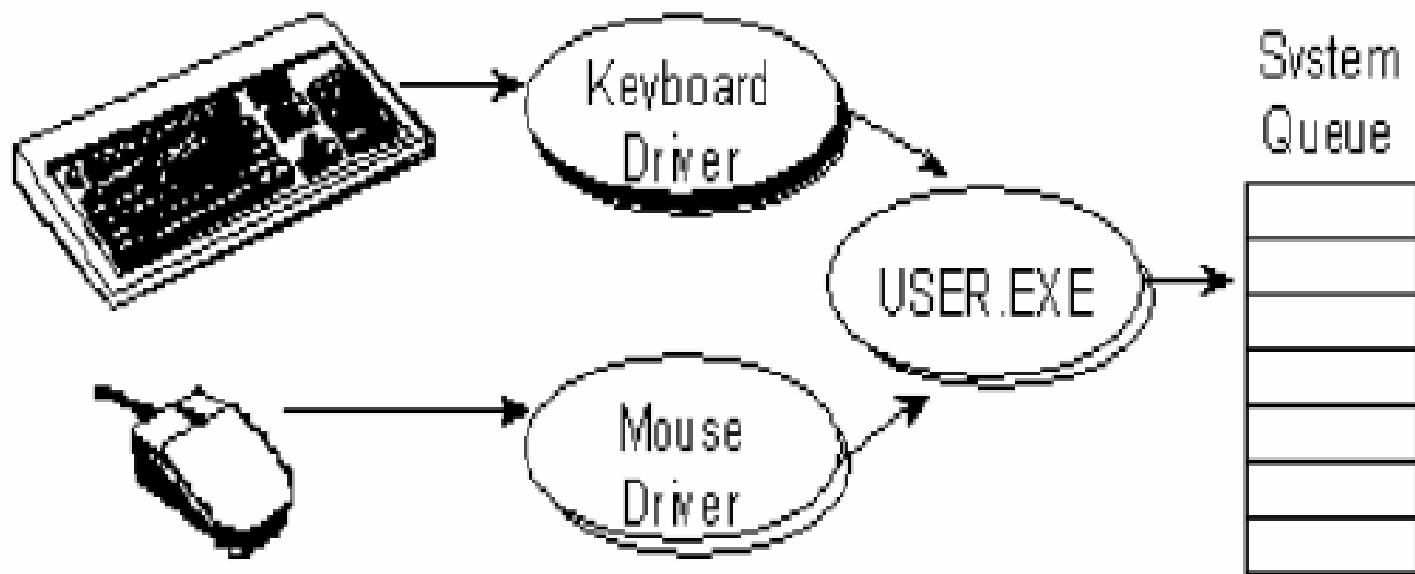
---

- ▶ Quy trình xử lý thông điệp
- ▶ Các loại thông điệp thường gặp
- ▶ Vòng lặp xử lý thông điệp



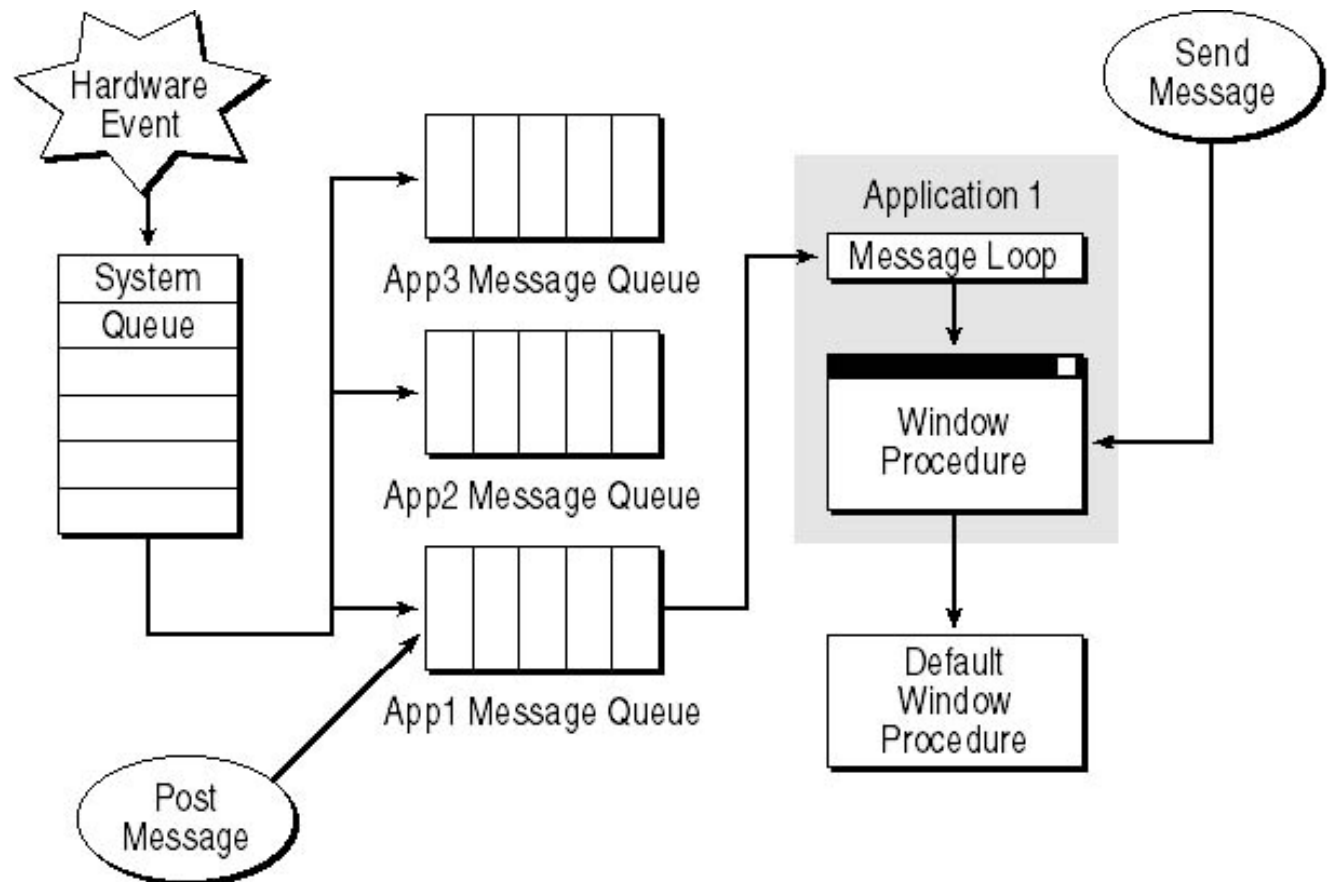
# Phát sinh các sự kiện và thông điệp

---



# Lập trình hướng sự kiện

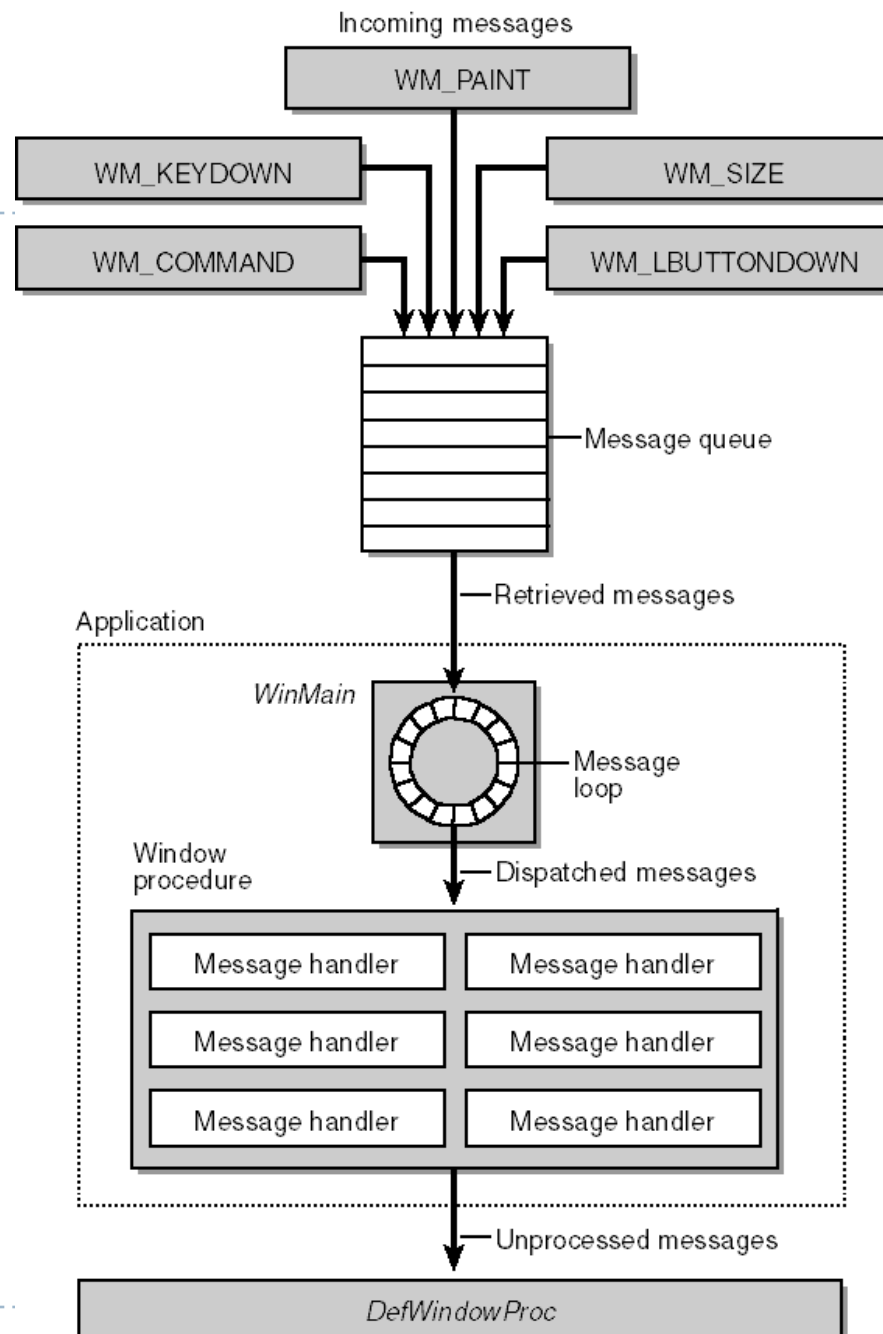
- Luồng điều khiển của chương trình được xác định dựa vào hành động của người dùng khi sử dụng chương trình



# 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



# Windows Application trên C

LRESULT CALLBACK WndProc (HWND, UINT, WPARAM, LPARAM) ;

```
int WINAPI WinMain (HINSTANCE hInstance, HINSTANCE hPrevInstance, PSTR szCmdLine, int iCmdShow)
```

```
{
```

```
    static TCHAR szAppName[] = TEXT ("...") ;
```

```
    HWND      hwnd ;
```

```
    MSG       msg ;
```

```
    WNDCLASS  wndclass ;
```

```
    wndclass.style          = CS_HREDRAW | CS_VREDRAW ;
```

```
    wndclass.lpfnWndProc    = WndProc ;
```

```
    ...
```

```
    wndclass.lpszClassName = szAppName ;
```

```
    if (!RegisterClass (&wndclass))
```

```
    {
```

```
        MessageBox (NULL, TEXT ("..."), ...) ;
```

```
        return 0 ;
```

```
    }
```

```
    hwnd = CreateWindow (szAppName,...,NULL) ;
```

```
    ShowWindow (hwnd, iCmdShow) ;
```

```
    UpdateWindow (hwnd) ;
```

```
    while (GetMessage (&msg, NULL, 0, 0))
```

```
    {
```

```
        TranslateMessage (&msg) ;
```

```
        DispatchMessage (&msg) ;
```

```
    }
```

```
    return msg.wParam ;
```

```
}
```


→ Khai báo cửa sổ

→ Đăng ký cửa sổ

→ Vòng lặp bắt thông điệp

# Windows Application trên C (Hàm xử lý thông điệp)


```
LRESULT CALLBACK WndProc (HWND hwnd, UINT message, WPARAM wParam, LPARAM lParam)
{
    HDC      hdc ;
    PAINTSTRUCT ps ;
    RECT      rect ;

    switch (message)
    {
    case WM_CREATE:  Thông điệp nhận được
        ...
        return 0 ;

    case WM_PAINT:
        ...
        return 0 ;

    case WM_DESTROY:
        ...
        return 0 ;

        ....
    }
    return DefWindowProc (hwnd, message, wParam, lParam) ;
}
```

 Xử lý thông điệp

# Một số thông điệp thường gặp

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



# Windows Application trên C#

## (Chương trình chính)

---

```
using System;
using System.Windows.Forms;
namespace NotepadForms
{
    public class Program
    {
        [STAThread]
        static void Main()
        {
            Form myForm = new MyForm(); —————> Khai báo cửa sổ

            Application.Run(new MyForm()); —————> Khởi động vòng lặp
                                           thông điệp
        }
    }
}
```

# Windows Application trên C# (Xử lý thông điệp)

```
public class MyForm : System.Windows.Forms.Form
{
    public MyForm()
    {
        this.Paint += new System.Windows.Forms.PaintEventHandler
            (this.MainForm_Paint);
    }

    private void MainForm_Paint (object sender,
                                System.Windows.Forms.PaintEventArgs evt)
    {
        ...
    }
}
```

**Gán sự kiện**

**Xử lý sự kiện**

# Tài nguyên - Resource

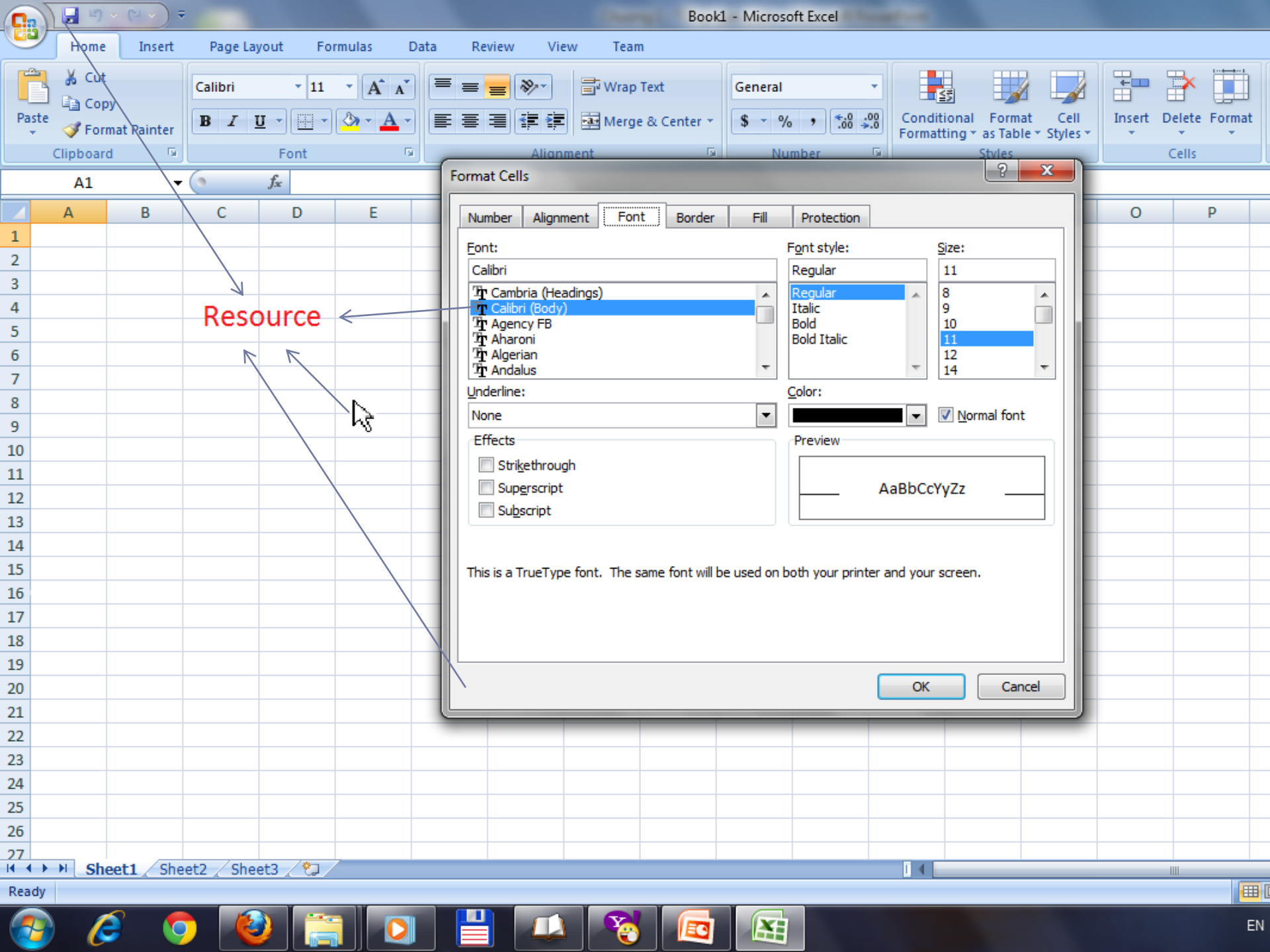
---

- ▶ 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”



# 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

---

- ▶ Tương tác giữa các thành phần trong và ngoài .NET Framework
- ▶ Common Language Runtime Engine
- ▶ Độc lập ngôn ngữ
- ▶ Thư viện lớp cơ sở
- ▶ Quản lý phiên bản
- ▶ Bảo mật
- ▶ Tính di động (portable)

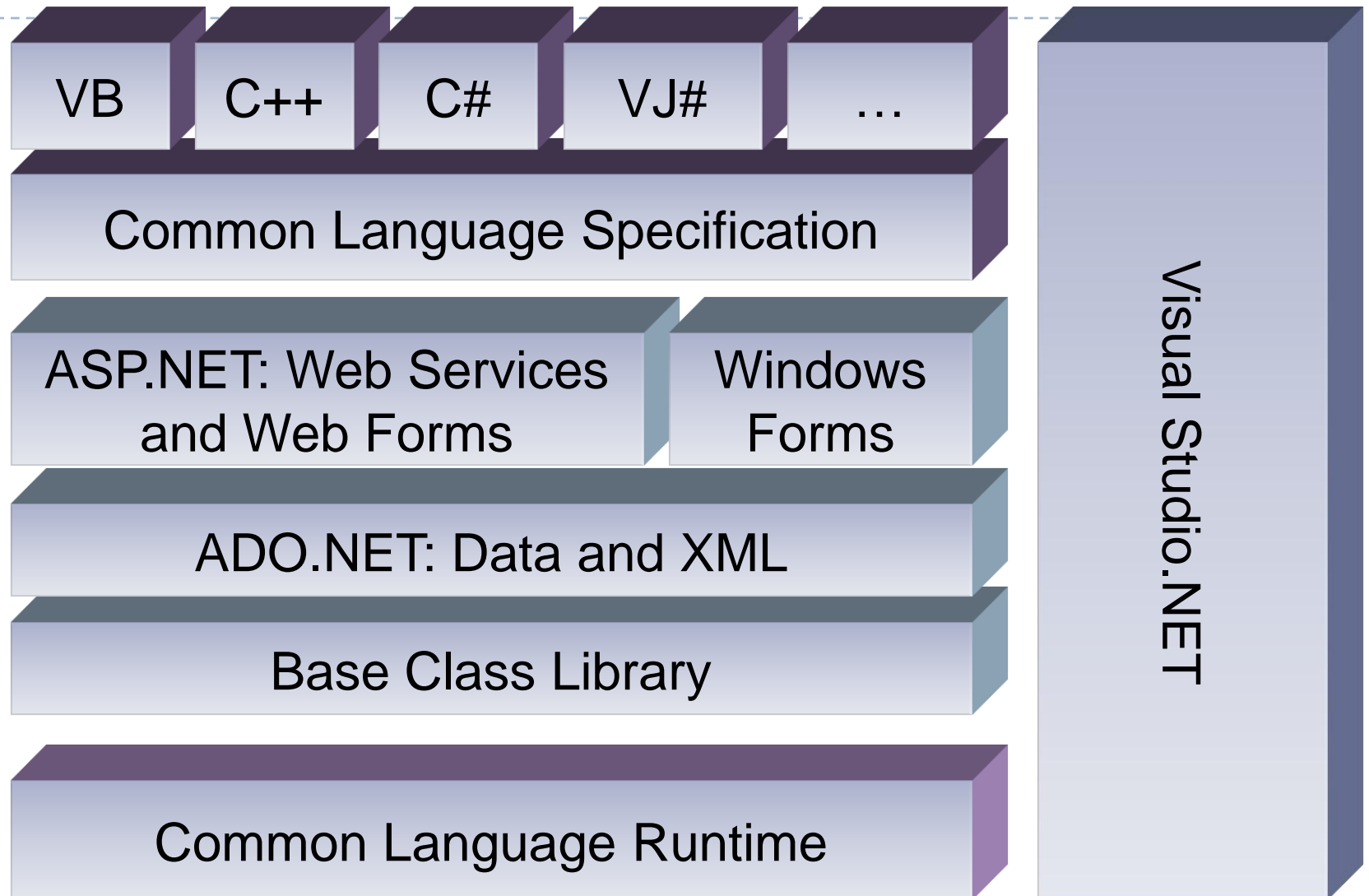
# Kiến trúc .NET Framework

---

- ▶ .NET framework bao gồm 4 thành phần chủ yếu
  - ▶ Common Language Runtime (CLR)
  - ▶ Các lớp cơ sở của .NET Framework
  - ▶ Data Management
  - ▶ 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

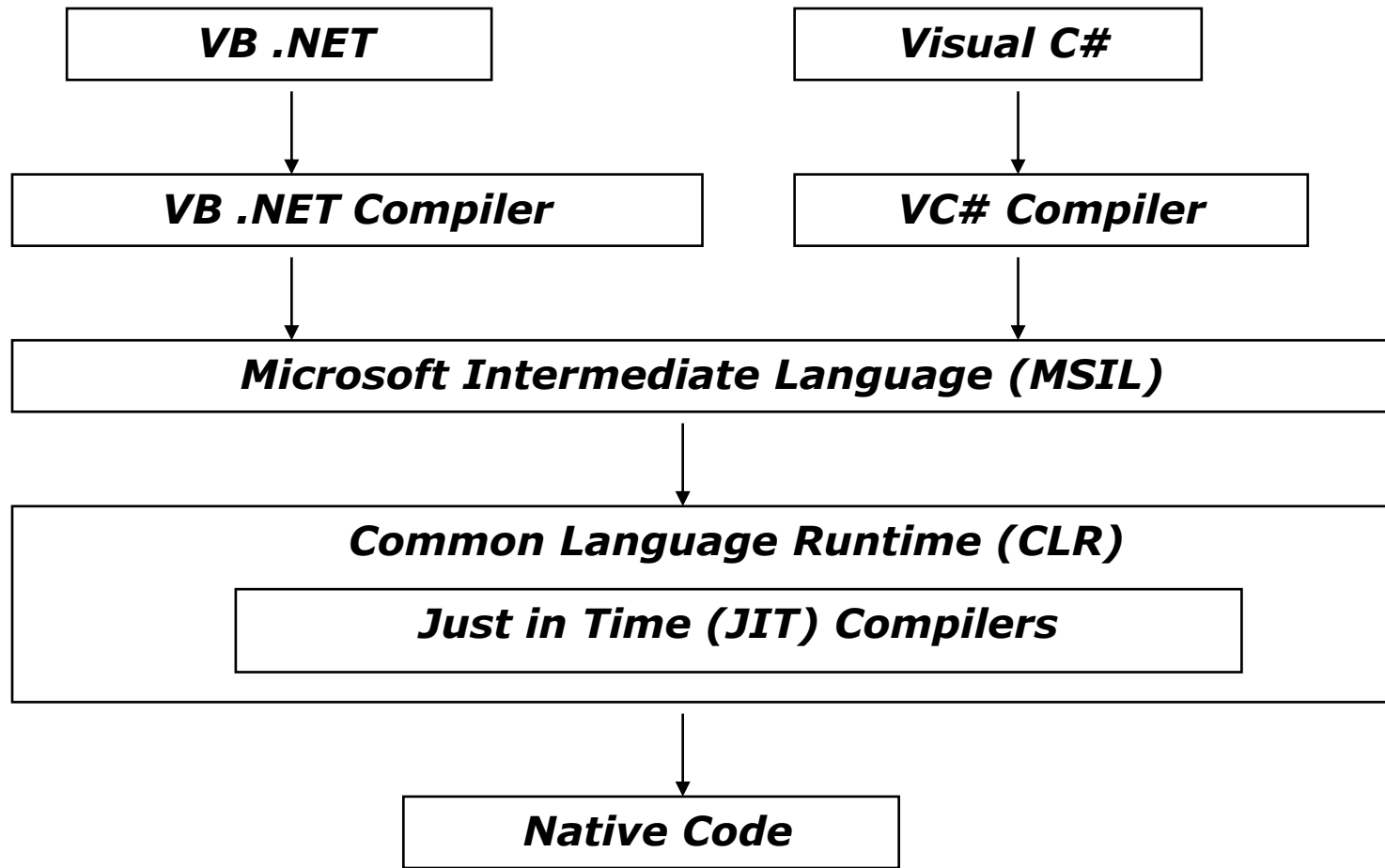
# Các đặc điểm của CLR

---

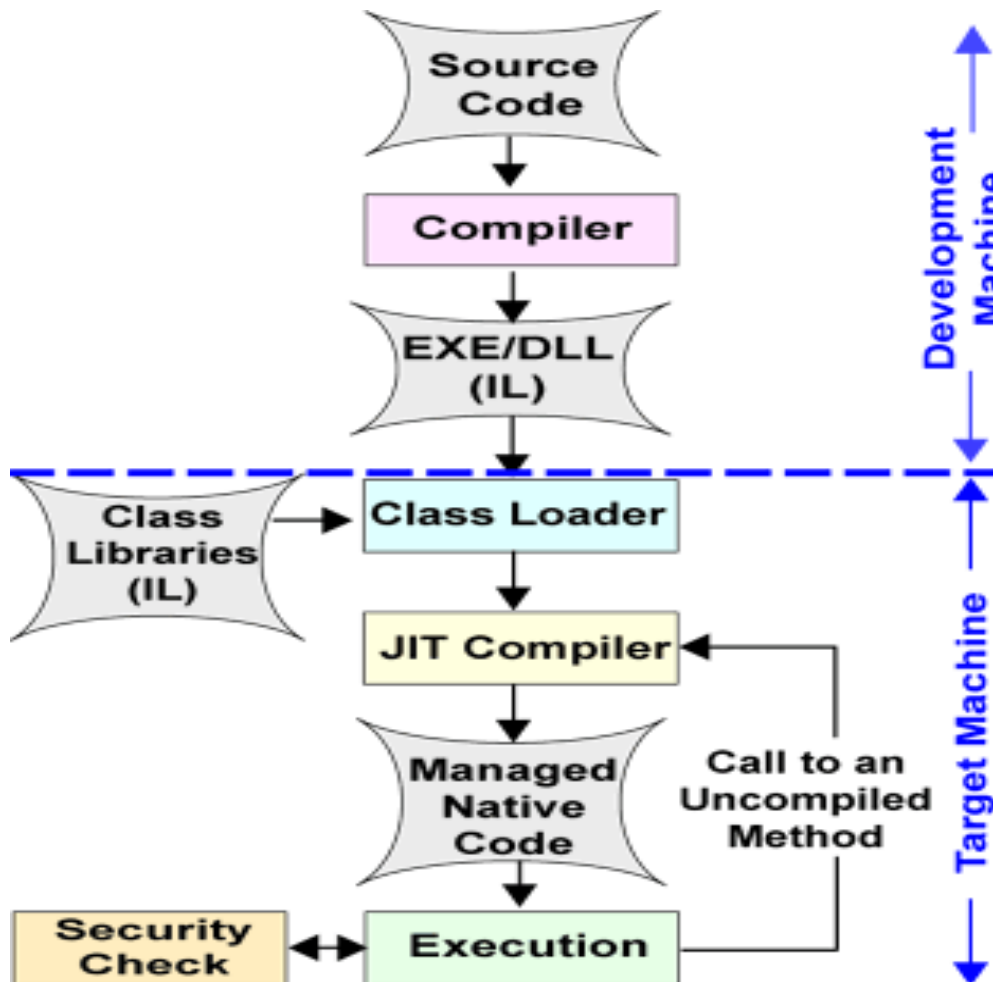
- ▶ Tự động quản lý bộ nhớ
- ▶ Hệ thống kiểu dữ liệu dùng chung giữa các ngôn ngữ lập trình
- ▶ Tính tương tác giữa các ngôn ngữ
- ▶ Độc lập với cấu trúc phần cứng bên dưới
- ▶ Cơ chế bảo mật hệ thống
- ▶ An toàn kiểu dữ liệu

# CLR

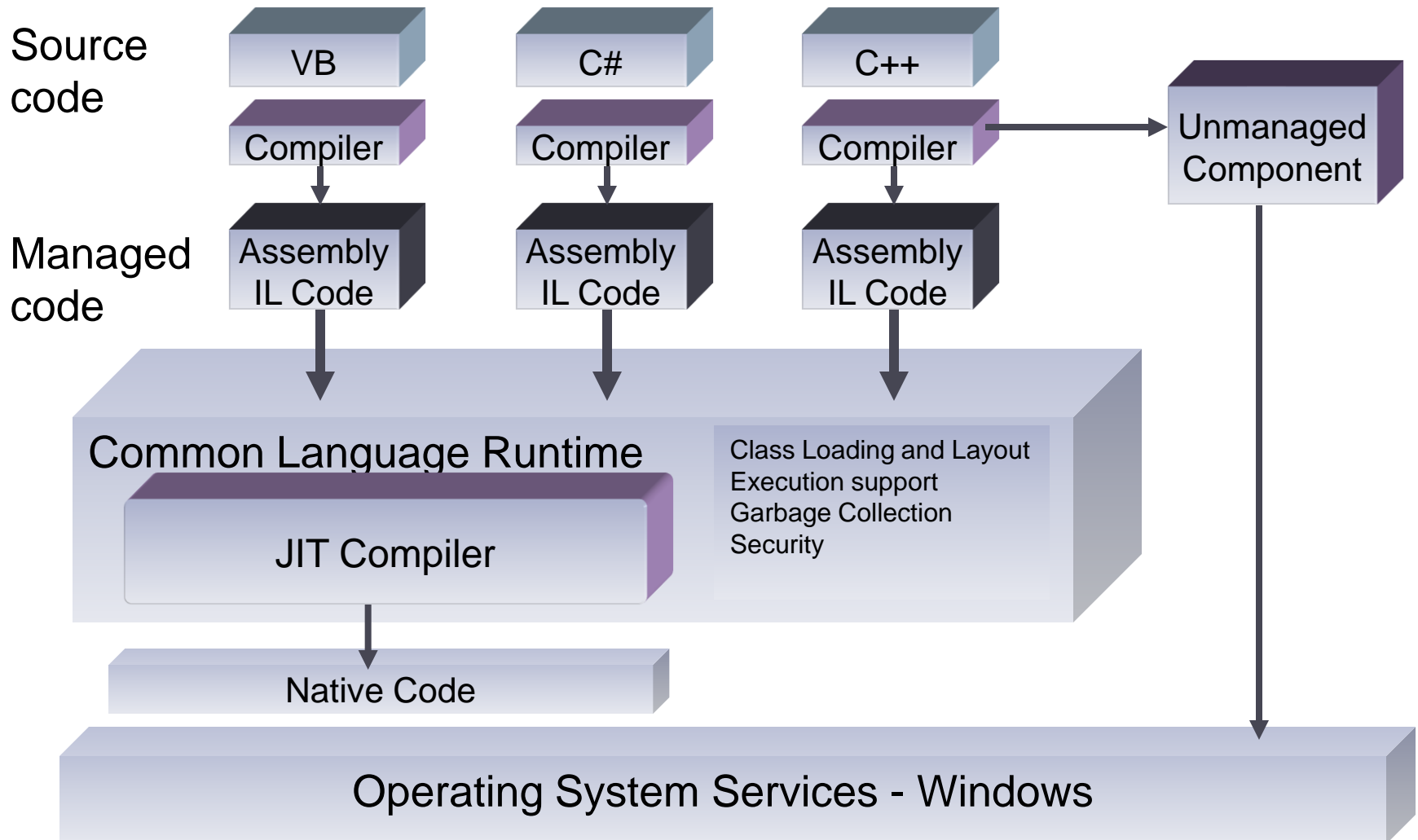
---



# Quá trình biên dịch



# CLR Execution Model



# 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

---



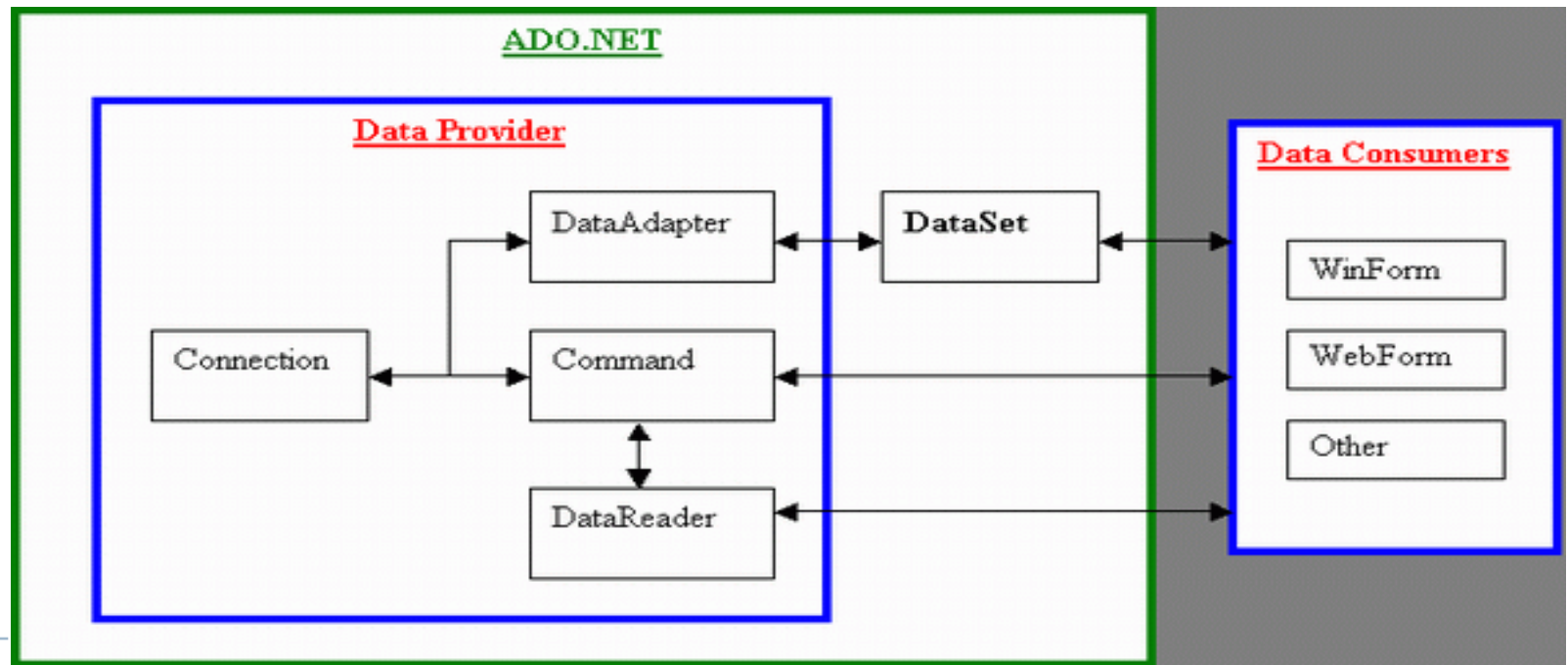
# Các lớp cơ sở của .NET framework

---

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

# ADO.NET

- ▶ Là các lớp truy cập dữ liệu cho .NET Framework
- ▶ Được thiết kế cho truy cập dữ liệu hiệu quả cao
- ▶ Hỗ trợ XML và disconnected record sets



# 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

# Common Language Specification

---

- ▶ Hệ thống kiểu dữ liệu chung (Common Type System – CTS)
- ▶ Metadata
- ▶ Đặc tả ngôn ngữ chung (Common Language Specification – CLS)
- ▶ Hệ thống thi hành ảo (Virtual Execution System – VLS)

# Common Language Specification

---

## ▶ Common Type System:

- ▶ Tích hợp cross-language, sử dụng kiểu an toàn và thi hành code hiệu năng cao
- ▶ Mô hình hướng đối tượng cho nhiều ngôn ngữ
- ▶ Các luật giúp sử dụng các đối tượng ở những ngôn ngữ khác nhau
- ▶ Các luật giúp các đối tượng viết bởi các ngôn ngữ khác nhau tương tác với nhau
- ▶ Các luật cho phạm vi của các kiểu
- ▶ Các luật quản lý thừa kế kiểu, phương thức ảo và thời gian sống của đối tượng

# 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

File Edit View Refactor Website Build Debug Tools Test Window Help

Debug .NET debug

Process: [4996] WebDev.WebServer Thread: [3372] RSSCache

Default.aspx App\_Code/RssCacheDependency.cs Default.aspx.cs

RssCacheDependency CheckDependencyCallback(object sender)

```

28 public XmlDocument RetrieveRSS(string URL)
29 {
30     XmlDocument retVal = new XmlDocument();
31     retVal.Load(URL);
32     return retVal;
33 }
34
35 public void CheckDependencyCallback(object sender)
36 {
37     RssCacheDependency CacheDepends = sender as RssCacheDependency;
38     XmlDocument NewRSS = RetrieveRSS(RSSUrl);
39     if (NewRSS.OuterXml != RSS.OuterXml)
40     {
41         CacheDepends.NotifyDependencyChanged(CacheDepends, EventArgs.Empty);
42     }
43 }
44

```

Thread(s) at Location:  
[9988] <No Name>

Khung soạn thảo

Quản lý solution

Properties

Threads

ID	Category	Name	Location	Suspen
7852	Main Thread	Main Thread		0
4176	Worker Thread	<No Name>		0
11164	Worker Thread	Worker Thread		0
3372	Worker Thread	RSSCache	RssCacheDependency.Ch	0
10332	Worker Thread	<No Name>		0
9988	Worker Thread	<No Name>	RssCacheDependency.Re	0
4696	Worker Thread	<No Name>		0

Output

'WebDev.WebServer.EXE' (Manage  
'WebDev.WebServer.EXE' (Manage  
'WebDev.WebServer.EXE' (Manage  
'WebDev.WebServer.EXE' (Manage  
'WebDev.WebServer.EXE' (Manage  
'WebDev.WebServer.EXE' (Manage  
'WebDev.WebServer.EXE' (Manage  
The thread 0x172c has exited v  
The thread '<No Name>' (0x233c  
The thread '<No Name>' (0x225c

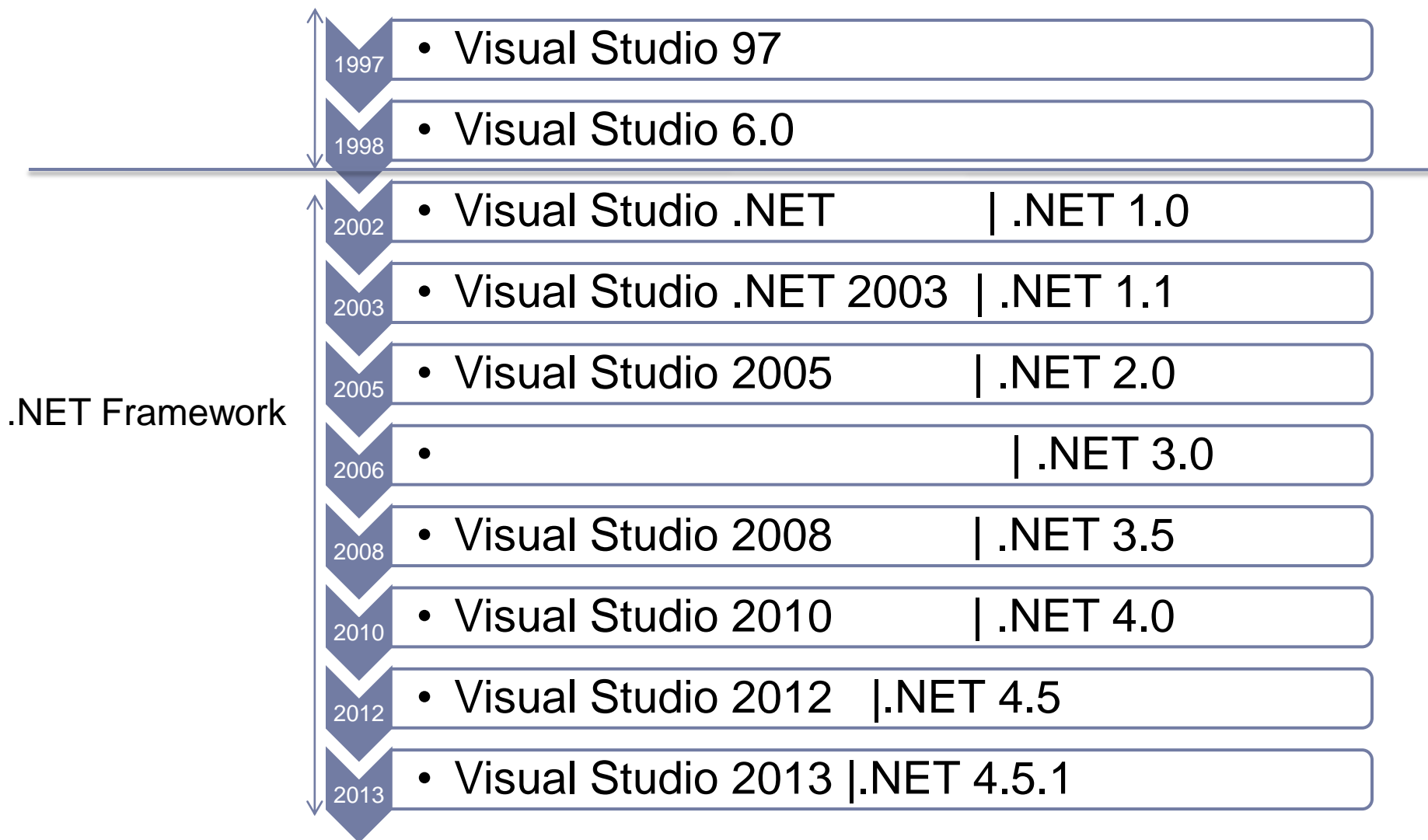
Quản lý tiến trình

Xuất kết quả

Error List Find Results 1 Locals Watch 1 Threads

Ready Ln 43 Col 6 Ch 6 INS

# Lịch sử Visual Studio



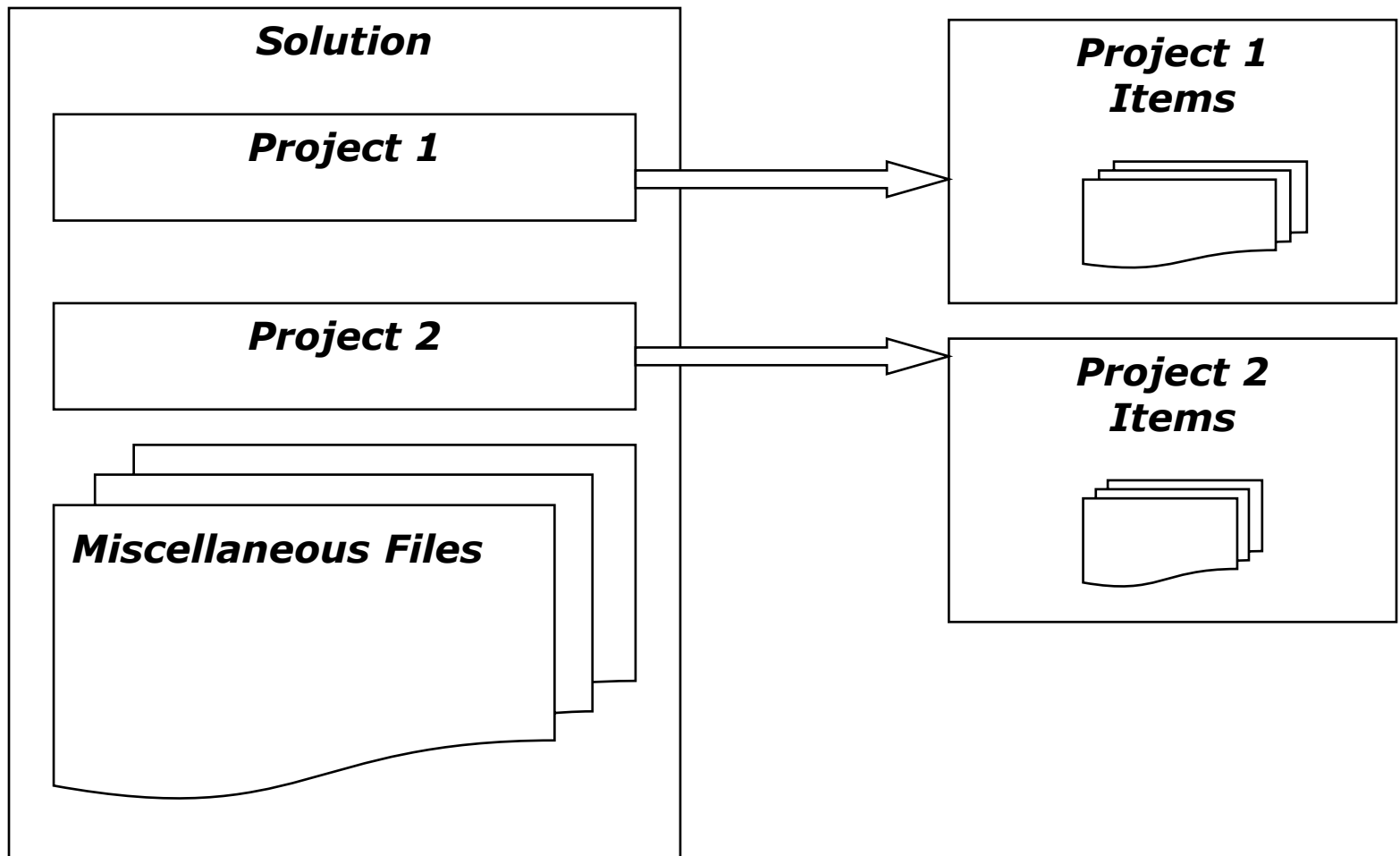
# Dự án và giải pháp

---

- ▶ 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