

武汉大学

# Windows 原理与应用

## 1.Windows 操作系统概述

计算机学院

《Windows 原理与应用》课程组

# 内容提要



操作系统及Windows发展



Windows编程工具



VS中的Windows 应用程序  
类型



实例

# 1.1 操作系统的发展

- ▶ 操作系统的用户界面（User Interface, UI），从简单原始的字符界面，发展到直观易用的图形界面，再到人性化的多媒体界面，目前正向智能感知、无处不在的普适计算（Pervasive Computing）的方向发展

OS: 无（控制面板）→ 批处理（电传打字机）→ 多任务（控制台/GUI）→ 分布式（网络/人性

UI: 无界面 → 字符界面 → 图形界面 → 多媒体界面 → 普适计算

OS 与 UI 实例: Unix / DOS / Linux → Mac OS / X-Window / Windows / KDE / GNOME

→ iOS / Android / Windows Phone / BlackBerry OS

## 1.2 网络操作系统的发展

在计算机网络上配置网络操作系统NOS (Network Operating System)，是为了管理网络中的共享资源，实现用户通信以及方便用户使用网络，因而网络操作系统是作为网络用户与网络系统之间的接口。

以推出的时间来说，UNIX为最早，Netware为第二，Windows NT最晚。

# 1.3 网络操作系统的特性(1)

## □ 客户/服务器模式

- ▶ 客户/服务器 (Client/Server) 模式是近年来流行的应用模式，它把应用划分为客户端和服务端，客户端把服务请求提交给服务器，服务器负责处理请求，并把处理的结果返回至客户端。

## □ 32位/64位操作系统

- ▶ 采用32位/64位内核进行系统调度和内存管理，支持32位/64位设备驱动器，使得操作系统和设备间的通信更为迅速。

## □ 抢先式多任务

- ▶ 网络操作系统一般采用微内核类型结构设计，微内核始终保持对系统的控制，并给应用程序分配时间段使其运行，在指定的时间结束时，微内核抢先运行进程并将控制移交给下一个进程。

# 1.3 网络操作系统的特性(2)

## □ 支持多种文件系统

- ▶ 有些网络操作系统还支持多文件系统，以实现系统升级的平滑过渡和良好的兼容性。

## □ 高可靠性

- ▶ 网络操作系统是运行在网络核心设备（如服务器）上的指挥管理网络的软件，它必须具有高可靠性，保证系统可以365天24小时不间断工作，并提供完整的服务。

## □ 安全性

- ▶ 为了保证系统、系统资源的安全性、可用性，网络操作系统往往集成用户权限管理、资源管理等功能，定义各种用户对某个资源存取权限，且使用用户标识SID唯一区别用户。

## □ 容错性

- ▶ 网络操作系统应能提供多级系统容错能力，包括日志式的容错特征列表、可恢复文件系统、磁盘镜像、磁盘扇区备用以及对不间断电源（UPS）的支持。

# 1.3 网络操作系统的特性(3)

## □ 开放性

- ▶ 网络操作系统必须支持标准化的通信协议（如TCP/IP、NetBEUI等）和应用协议（如HTTP、SMTP、SNMP等），支持与多种客户端操作系统平台的连接。

## □ 可移植性

- ▶ 网络操作系统一般都支持广泛的硬件产品，往往还支持多处理机技术。这样使得系统就有了很好的伸缩性。

## □ 图形化界面(GUI)

- ▶ 网络操作系统良好的图形界面可以简化用户的管理，为用户提供直观、美观、便捷的操作接口。

## □ Internet 支持

- ▶ 各品牌网络操作系统都集成了许多标准化应用，例如Web服务、FTP服务、网络管理服务等等的支持，甚至E-mail（如Linux的Sendmail）也集成在操作系统中。

# 1.3 网络操作系统的功能(1)

网络操作系统功能通常包括：处理机管理、存储器管理、设备管理、文件系统管理以及为了方便用户使用操作系统向用户提供的用户接口，网络环境下的通信、网络资源管理、网络应用等特定功能。此外还有：

## 1. 网络通信

这是网络最基本的功能，其任务是在源主机和目标主机之间，实现无差错的数据传输。

## 2. 资源管理

对网络中的共享资源（硬件和软件）实施有效的管理、协调诸用户对共享资源的使用、保证数据的安全性和一致性。



# 1.3 网络操作系统的功能(2)

## 3. 网络服务

电子邮件服务

文件传输

存取和管理服务

共享硬盘服务

共享打印服务

## 4. 网络管理

网络管理最主要的任务是安全管理，一般这是通过“存取控制”来确保存取数据的安全性；以及通过“容错技术”来保证系统故障时数据的安全性。

## 5. 互操作能力

所谓互操作，在客户/服务器模式的LAN环境下，是指连接在服务器上的多种客户机和主机，不仅能与服务器通信，而且还能以透明的方式访问服务器上的文件系统。

# 1.4 网络操作系统的分类

- ▶ UNIX

- ▶ Sun Solaris
- ▶ HP-UX
- ▶ IBM AIX

- ▶ Linux

- ▶ Debian
- ▶ Redhat
- ▶ CentOS
- ▶ ...

- ▶ Windows NT / WindowsXP / Windows7 / Windows 8 / Windows 10

- ▶ Windows Server 2000 / Server 2003 / Server2008 / Server 2012 / Server 2016...

- ▶ Netware

# 1.5 Windows 家族

Windows 家族				
早期版本	For DOS	▪ Windows 1.0 ( 1985 )	▪ Windows 2.0 ( 1987 )	▪ Windows 2.1 ( 1988 )
		▪ windows 3.0 ( 1990 )	▪ windows 3.1 ( 1992 )	▪ Windows 3.2 ( 1994 )
NT系列	Win 9x	▪ Windows 95 ( 1995 )	▪ Windows 98 ( 1998 )	▪ Windows 98 SE ( 1999 )
		▪ Windows Me ( 2000 )		
	早期版本	▪ Windows NT 3.1 ( 1993 )	▪ Windows NT 3.5 ( 1994 )	▪ Windows NT 3.51 ( 1995 )
		▪ Windows NT 4.0 ( 1996 )	▪ Windows 2000 ( 2000 )	
NT系列	客户端	▪ windows xp ( 2001 )	▪ Windows Vista ( 2005 )	▪ Windows 7 ( 2009 )
		▪ Windows 8 ( 2011 )		
	服务器	▪ Windows Server 2003 ( 2003 )	▪ Windows Server 2008 ( 2008 )	
		▪ Windows Home Server ( 2008 )	▪ Windows HPC Server 2008 ( 2010 )	
NT系列		▪ Windows Small Business Server ( 2011 )	▪ Windows Essential Business Server	
	特别版本	▪ Windows PE	▪ Windows Azure	
NT系列		▪ Windows Fundamentals for Legacy PCs		
	嵌入式系统	▪ Windows CE	▪ Windows Mobile	▪ Windows Phone 7 ( 2010 )

# 1.5 Windows的发展(1)

- 早期的Windows并不是一个独立的操作系统，而只是DOS的一个补丁程序。
- 最初的Windows 1.0（于1985年11月20日推出）和2.0，运行于16位的286 PC机上，又慢又差，被业界所唾弃。
- 随着386 CPU的推出和微软公司的不断改进，1990年推出的（16位）Windows 3.0获得了巨大的成功，后来推出的带浏览器IE的（32位）Windows 95/98也得到广泛好评。
- 1993年微软开发出基于微内核结构的32位独立操作系统Windows NT（New Technology，新技术），2000年和2001年分别推出基于NT的Windows 2000和Windows XP（eXPerience，体验），大受好评，被广泛使用。

# 1.5 Windows的发展(2)

- 2006年底，微软推出（同样基于NT）的Windows Vista（远景/街景/展望）内含.NET框架和DirectX模块、还引入了全新的WPF图形子系统，具有华丽的Aero用户界面。
- 但由于Vista对硬件要求高、兼容性差、且速度慢，惨招失败。微软痛定思痛，对Vista进行了大量的改进，于2009年秋推出了运行效率高、对硬件要求低、而且兼容性好的Windows 7，获得了一致好评。
- 2012年推出Windows 8引入Metro（地铁）卡片界面，支持ARM CPU和多点触控，同时支持PC机、平板电脑和智能手机，意欲改变Windows操作系统在平板电脑和智能手机领域的落后状态。

# 1.5 Windows的发展(3)

- ▶ 继续推出Windows 8.1、Windows 10 和 Windows Server服务器企业级操作系统
- ▶ DOS->图形化GUI->图形化GDI+>WPF
- ▶ 16位->32位->64位

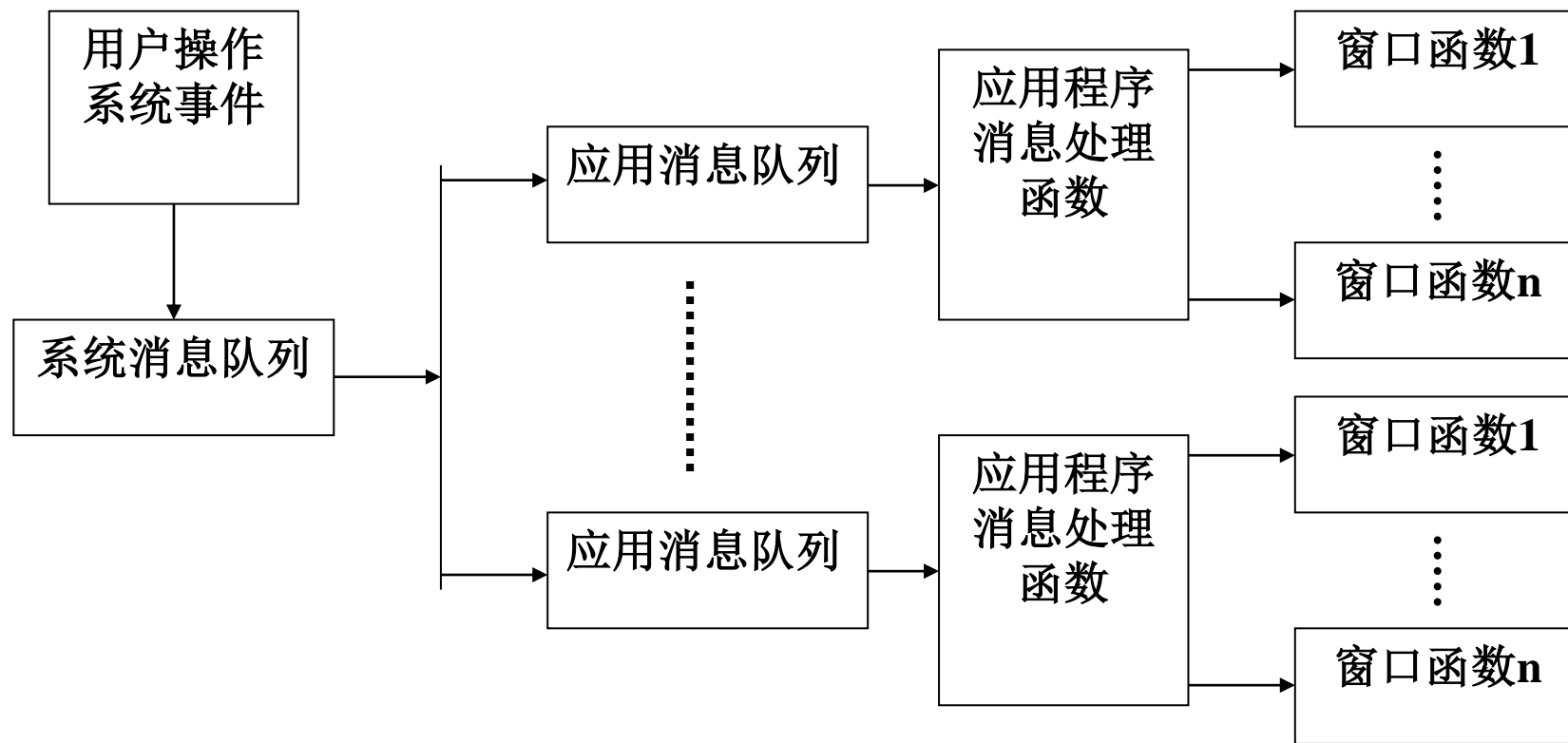
# 1.6 Windows 操作系统的主要特点(1)

## ➤ 面向对象

在Windows中，窗口、菜单、事件皆是对象，而对话框与各种控件只是一些特殊的窗口，所以对界面元素的操作和消息/事件的处理都涉及到对象。这些对象的属性和操作，由API的数据结构和函数（或由MFC和.NET框架中的类）提供。

# 1.6 Windows 操作系统的主要特点(2)

## ➤ 消息/事件驱动





# 1.6 Windows 操作系统的主要特点(3)

## ➤ 资源共享与数据交换

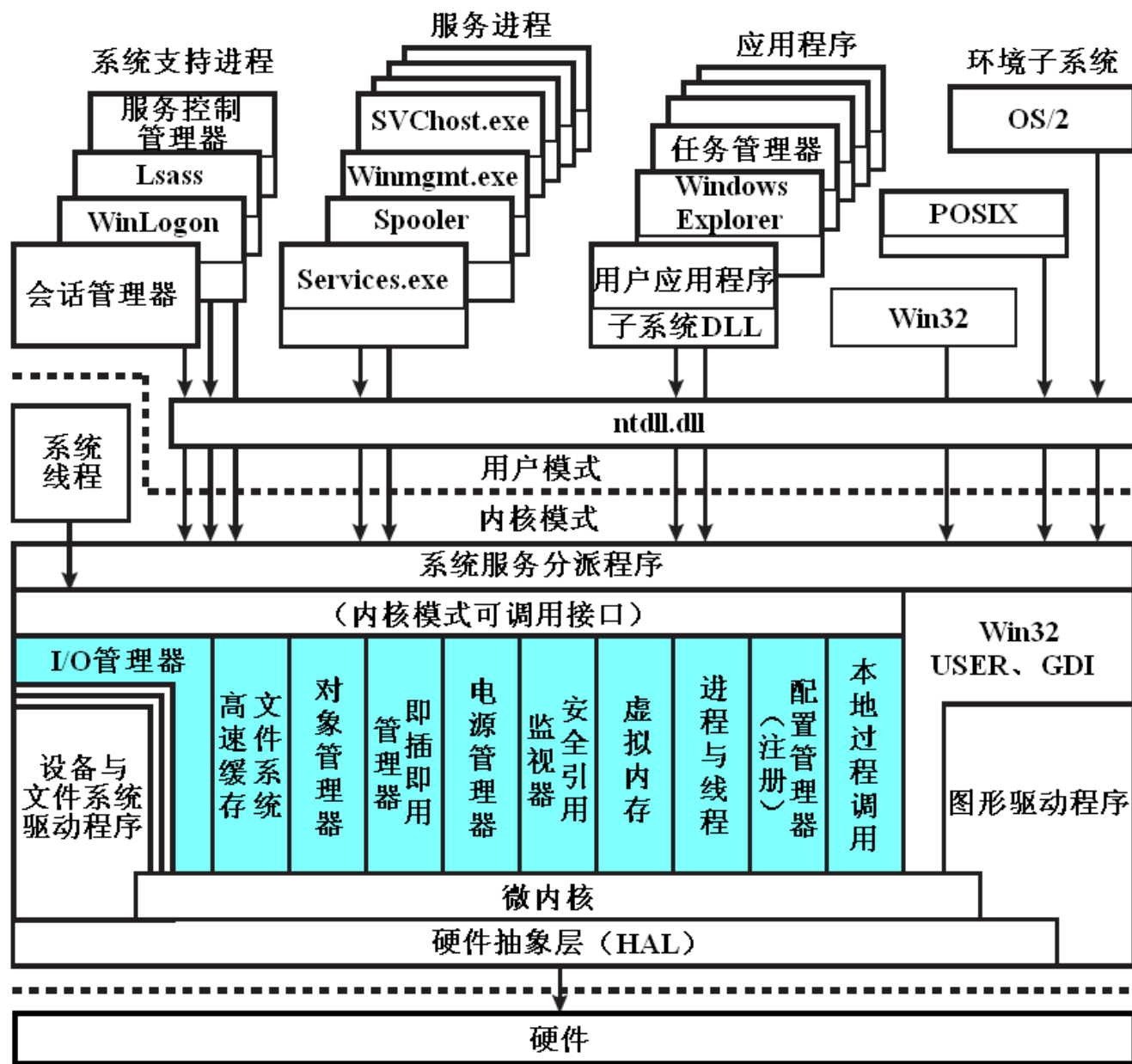
标准的MS-DOS程序在运行时，可独占计算机的所有资源。但由于Windows是一种（抢先式）多任务操作系统，所以Windows应用程序必须和正在运行的其他程序共享这些资源。因此在进行Windows程序设计时，必须时刻记住资源共享这一多任务OS的特点，以避免耗尽资源而造成系统死机。

# 1.6 Windows 操作系统的主要特点(4)

## ➤ 与设备无关的GDI

Windows为应用程序提供了丰富的与设备无关API，免去了为不同的I/O设备编写软件的不同版本之烦恼。如利用与设备无关的GDI，应用程序使用同一函数，可在不同的显卡、打印机和显示器上输出同一个图形。

# 1.7 Windows 操作系统的构成



# 1.8 Windows 操作系统的图形子系统

Windows的图形子系统经历了GDI、GDI+ 和WPF三个发展阶段。

## ➤ GDI

传统的GDI (Graphics Device Interface, 图形设备接口) 是微软随Windows 1.0 (1985年11月) 推出的Windows基本图形子系统, 用于与显卡和显示设备无关的图形界面编程, 支持WYSIWYG (What You See Is What You Get, 所见即所得)。

## ➤ GDI+

新式的GDI+是随Windows XP (2001年10月) 推出的GDI的改进版, 增加了 $\alpha$ 混色、渐变画刷、样条曲线、矩阵变换、图像处理、持久路径等新功能, GDI+也是.NET 框架中Windows窗体使用的图形子系统。

## ➤ WPF

随Windows Vista和.NET框架3.0 (2006年11月) 微软又推出了全新的Windows 图形子系统WPF (Windows Presentation Foundation, 视窗展示基础), 它统一了桌面和浏览器等客户端应用程序的图形界面, 基于DirectX和.NET框架, 采用声明式XAML进行界面设计, 是Windows的下一代GUI显示系统。

# 1.9 Windows 资源工具包

资源工具包Microsoft Windows Resouce Kit是Windows 的附件。它包含丰富的操作系统扩展工具，使用这些工具可以用来管理和调整优化操作系统。

Windows Resouce Kit提供了许多额外工具，它们可以完成多种调整任务。

Resource Kit实际上是Microsoft为管理员提供的一套额外的工具集，包括了超过400个各种工具、vbs、dll、msc，涵盖了管理TCP/IP、网络、注册表、安全、远程管理、配置、Batch文件、以及操作系统的其他方面。可以让管理员更容易的管理一个网络系统。这里我们介绍几个重要的管理工具。

## Appsec.exe (Application Security)

Appsec.exe 是一个基于GUI的应用程序，它允许管理员在一个多用户环境下限制普通用户访问一组网络上经预订的应用程序。启用这种应用程序安全性，将会导致系统拒绝普通用户执行或使用一个未经许可的应用程序。

## Delprof.exe (User Profile Deletion Utility)

删除用户的Profile工具，当然相同操作可以在System Properties里面完成。而这个工具可以在本地或是远程运行，且可以指定多长时间不使用就删除，在处理比较多的用户的时候十分方便。

其命令格式：

```
delprof /q /i /p /c:\\computername /days /?
```

其中：

/q：指后台执行，不用确认。

/i：忽略错误继续删除。

/p：在删除前提示确认。

/days：指定多少天不使用的Profile就删除，days用整数。

## Instsrv.exe (Service Installer)

Instsrv.exe 一个可以用来安装或者是删除可执行服务 (.exe) 命令行工具，当然也可以给服务分配名字，其语法如下：

```
instsrv ServiceName PathToExecuteable -a accountname -p  
accountpassword
```

或者 instsrv Servicename remove

ServiceName: 用来显示的服务名字。

PathToExecuteable: 要安装服务文件的全路径，包括盘符在内。

-a accountname: 指定服务运行的帐户。

-p accountpassword: 帐户的密码。

servicename remove: 指定想移出的服务。



## Inuse.exe (File-In-Use Replace Utility)

一个用来替换被锁定的操作系统文件命令行工具，比如一些重要的DLL文件。这些文件会在下一次启动的时候被替换，这种替换会在下一次启动的Autochk完成之后，创建Paging Files之前。

语法： `insuse -? source destinaion /y`

source：指明想用来替换的文件，可以使用完整的物理路径或者是UNC路径。

Destination：指明被替换的文件。

/y：表示确定。

## Netsvc.exe (Command-line Service Controller)

一个用来控制服务的命令行工具。用这个工具可以进行远程启动、停止、暂停、继续服务或者查询一个服务的状态。

该命令语法：

```
netsvc command servicename \\computername /? | /help
```

command: 可以是以下几个命令：

/list: 列出已经安装的服务，这个时候不使用 servicename。

/query: 查询一个服务的状态。

/start, /stop, /pause, /continue : 启动、停止、暂停、继续。

## Regback.exe (Registry Backup)

注册表备份工具，允许用户在系统运行期间对注册表进行备份。

语法：

```
regback destination_dir filename hivetype hivename more | /?
```

destination\_dir: 备份文件的位置。

filename: 要创立的文件。

hivetype: machine或者是users。

hivename: HKLM或者HKLU。

## Srvany.exe (Applications as Services Utility)

可以让Windows应用程序像一个服务那样运行。

需要对注册表进行如下操作：

(1) 在

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\MyService\下 添加一个Parameters子键，这里的MyService是刚刚指定的Srvany的服务名字。

(2) 在这个subkey底下建立一个应用程序入口，使用REG\_SZ类型。

(3) 指定程序的全路径，包括扩展名，比如Application:  
REG\_SZ: C:\Tools\srv.exe。

## Srvany.exe (Applications as Services Utility)

(4) 指定相应的启动参数。在Parameters底下接着创建一个AppParameters条目，也是REG\_SZ类型。比如：

AppParameters: REG\_SZ: C:\tmp\example, 当然因为srvany已经被安装为一个服务了，用户也可以在Services面板里指定。C:\Tools\srv.exe C:\\temp\example。注意，这里使用\\来表示一个\。

(5) 指定环境变量。添加一个AppEnvironment, REG\_MULTI\_SZ类型，它是Services面板里面的Dependencies项。可以根据需要指定，没有就不需要添加。

(6) 指定Working Directory。也是在Parameters里面建立一个AppDirectory使用REG\_SZ类型。比如：AppDirectory: REG\_SZ: C:\Tmp 当然也可以在Services里面 /D c:\\tmp D:\\Tools\\Vi.exe c:\\tmp\\example。

## Ptree.exe ( Process Tree )

允许用户在本地或者是远程查询进程树。

其命令行格式为：

```
ptree -c computer -k | -kt process -? | /?
```

-c computer: 管理远程主机的时候才需要用的这个参数，如果是本机，直接ptree命令即可。

-k process: 停止后面指定的进程，这里的process可以是进程的名字也可以是进程的PID。

# 1.10 Windows 编程工具(1)

- 早期的语言工具的主体基于命令行的编译器（如MS C），后来才出现直观易用的集成开发环境（如Turbo Pascal）和基于软件模块拖放组合的可视编程工具（如Visual Basic和Delphi）。
- 常用的高级编程语言（如Fortran、Basic、Pascal、C），其中主流的是OOP语言（如C++、Java、C#等）。随着Web网络的兴起和普及，多种简单实用的脚本语言（如JavaScript、PHP、Perl、Ruby等）也开始流行起来。

## 1.10 Windows 编程工具(2)

- 最早的Windows系统是用汇编、C和Pascal语言编写的（现在则主要用C++），Windows程序员最初都是用C语言（主要是微软无IDE的MS C，）直接调用Windows SDK（Software Development Kit，软件开发工具包）中的Windows API（Application Program Interface，应用程序接口）来编写Windows应用程序。
- 1991年5月，Tripod（三脚架）公司的Alan Cooper按微软公司的要求，将Basic语言嵌入他自己开发的可视编程环境“Ruby”中，为Windows 3.0推出了世界上第一个流行的可视编程软件Visual Basic for Windows 1.0。



# 1.10 Windows 编程工具(3)

- Borland公司将买下White Water的 C++ Framework（框架）改名为OWL（Object Windows Library，对象视窗库，一种Windows API的C++类库），于1992年6月推出了以OWL为核心的Borland C++ 3.1，支持Windows的OOP编程，且并带有直观易用的IDE，受到Windows程序员的热情追捧。
- 由于微软公司利用自己在PC机操作系统上的垄断优势，将它的新推出的OLE和COM等组件技术，集成到它自己的开发工具中，而对Borlan等其他公司保密。因此，Visual C++和Visual Basic才得以取代Borland C++在Windows编程中的霸主地位。

## 1.10 Windows 编程工具(4)

- Borland公司并不甘心，Anders Hejlsberg于1995年开发出了嵌入Pascal语言、基于组件的可视编程工具Delphi，它比VC更好用、比VB的功能更强大、长于数据库连接，受到了Windows程序员的热烈欢迎，可与微软的VB和VC分庭抗礼。
- 1995年5月Sun公司推出了由James Gosling等人开发的一种跨平台、面向对象、适应网络环境的全新编程语言Java，彻底改变了Windows开发工具的格局

# 1.10 Windows 编程工具(5)

- Borland公司并不甘心，Anders Hejlsberg于1995年开发出了嵌入Pascal语言、基于组件的可视编程工具Delphi，它比VC更好用、比VB的功能更强大、长于数据库连接，受到了Windows程序员的热烈欢迎，可与微软的VB和VC分庭抗礼。
- 1995年5月Sun公司推出了由James Gosling等人开发的一种跨平台、面向对象、适应网络环境的全新编程语言Java，彻底改变了Windows开发工具的格局

## 1.10 Windows 编程工具(6)

- 为了安全和跨平台，Java采用了字节码和虚拟机，程序须经过解释才能执行，大大降低了Java程序的运行效率。
- 2002年2月，微软公司推出了由Anders Hejlsberg领导开发的跨语言、跨平台、面向组件的运行环境——.NET框架和简单现代、面向对象、类型安全、运行高效的编程语言——C#。
- 到此，Anders Hejlsberg才真正实现了他的设计和梦想，微软公司的开发平台和编程工具也因此发展到了一个全新的阶段。

# 1.11 Windows 编程工具-.NET 框架

- .NET 框架 (.NET Framework) 是微软公司为了与 Sun/Oracle 公司的 Java (EE) 竞争，于 2000 年 6 月提出来的。一种新的跨语言、跨平台、面向组件的操作系统环境，适用于 Web 服务 (Web services) 和因特网 (Internet) 分布式应用程序的生成、部署和运行。.NET 框架也是 Windows Vista、Windows 7 和 Windows 8 等新版 Windows 操作系统的核心部件
- .NET Framework 4.5/4.6...

# 1.11 Windows 编程工具-Visual Studio

Visual Studio (VS, 可视工作室) 是基于.NET框架软件的开发平台, 也是.NET的主要组成部分。  
Visual Studio采用了支持可视编程的集成开发环境。

- ▶ 1993年2月 Visual Workbench (可视工作台) 1.0。
- ▶ 1993年12月 Visual Workbench 1.5。
- ▶ 1994年9月 Visual Workbench 2.0。
- ▶ 1995年7月 Developer Studio (开发者工作室) 4.0。
- ▶ 1997年1月 Visual Studio (可视工作室) 5.0 (97)。
- ▶ 1998年6月 **Visual Studio 6.0** (98)。
- ▶ 2002年1月 Visual Studio.NET (7.0)。
- ▶ 2003年5月20日 Visual Studio.NET 2003 (7.1)。
- ▶ 2005年11月7日 Visual Studio 2005 (8.0)。
- ▶ 2007年11月16日 Visual Studio 2008 (9.0)。
- ▶ 2008年8月11日 Visual Studio 2008 SP1 (9.0 SP)。
- ▶ 2010年4月12日 Visual Studio 2010 (10.0), 中文版于2010年6月1日推出。
- ▶ 2011年3月3日 Visual Studio 2010 SP1 (10.0 SP)。
- ▶ 2012年9月12日 Visual Studio 2012 (11.0支持Metro界面编程)。
- ▶ 2013年11月13日, 微软发布 Visual Studio 2013。
- ▶ 2014年11月, 微软发布 Visual Studio 2015。
- ▶ 2017年3月, 微软发布 Visual Studio 2017。

## 1.12 Windows 编程工具的选择（仅供参考）

- 小型应用软件：Visual Basic。
- 大型/系统软件：Visual C++。
- .NET软件：Visual C#。
- 网络/安全软件：Java。
- Web/XML软件：Java / Visual C#。
- 数据库通用软件：Delphi / PowerBuilder。

# Windows 编程工具的选择（仅供参考）

- ▶ 在 Visual Studio 提供的各种语言工具中，只有用 Visual C++ 才能编写传统的 Windows 应用程序。另外，VC 也是 VS 中唯一的一种可以同时[混合]编写非托管（API 与 MFC/ATL）程序和托管（.NET）程序的工具，
- ▶ VS 中的其他语言工具（如 C#、VB 和 F# 等）则只能编写 .NET 环境下的托管程序



# Windows编程工具的选择（仅供参考）

传统应用程序		托管应用程序		Java应用程序
Delphi、 C++Builder、 Powerbuilder 、 ...	Visual C++ (含MC++或C++/CLI)		Visual C#、 Visual Basic、 Visual F#、 JScript、 IronPython、 IronRuby、 C# Builder、 .....	Java
	C++类库 (MFC/ATL)	.NET框架类库 (FCL)		JFC/EJB
		.NET运行环境 (CLR)		JVM
	应用程序接口 (Win32/64 API)			
操作系统 (Windows)				
计算机硬件 (PC机)				

41

# Visual Studio 中的 Windows 应用程序类型

应用程序类型与开发语言有一定的关系

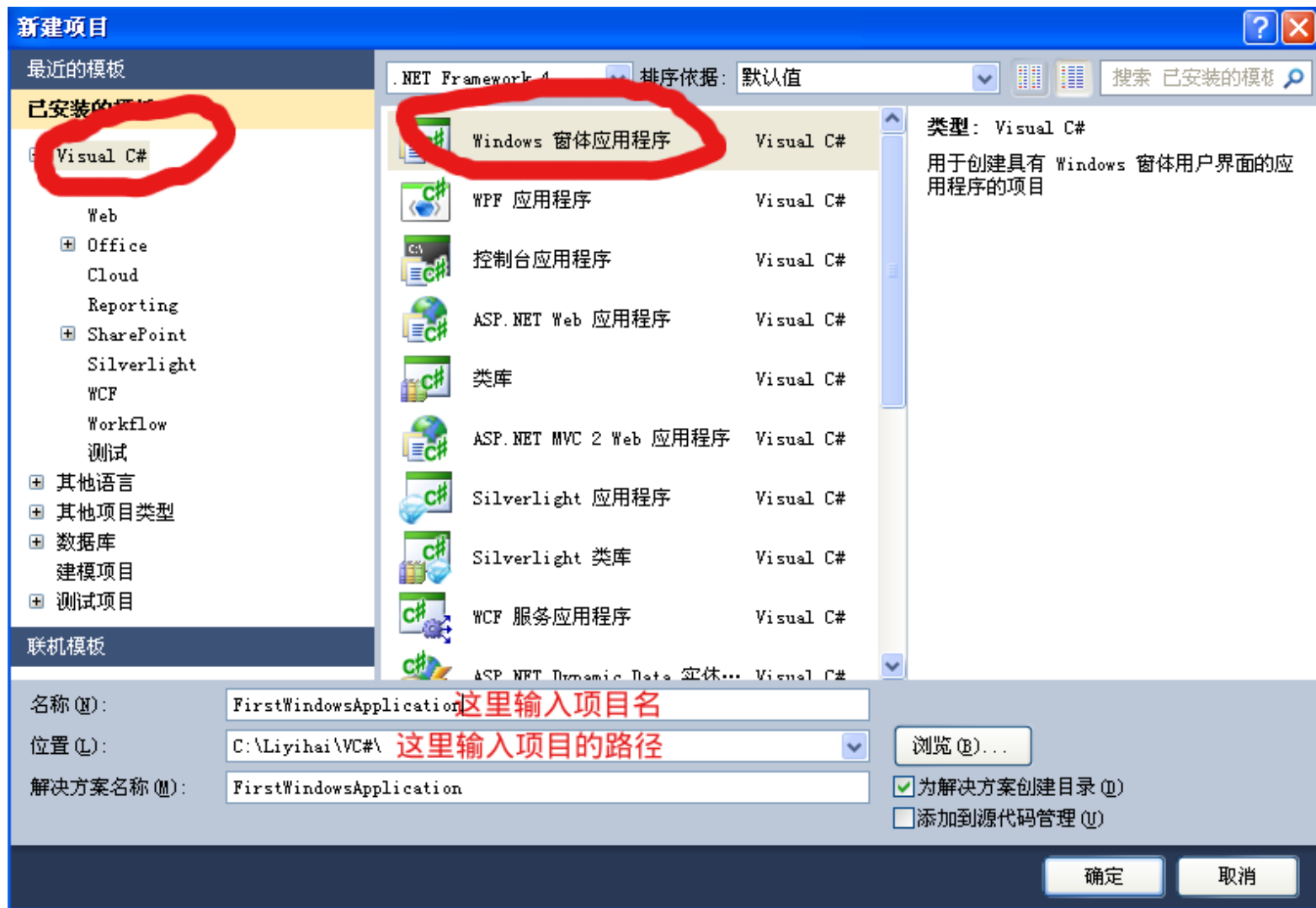
## ▶ VC++

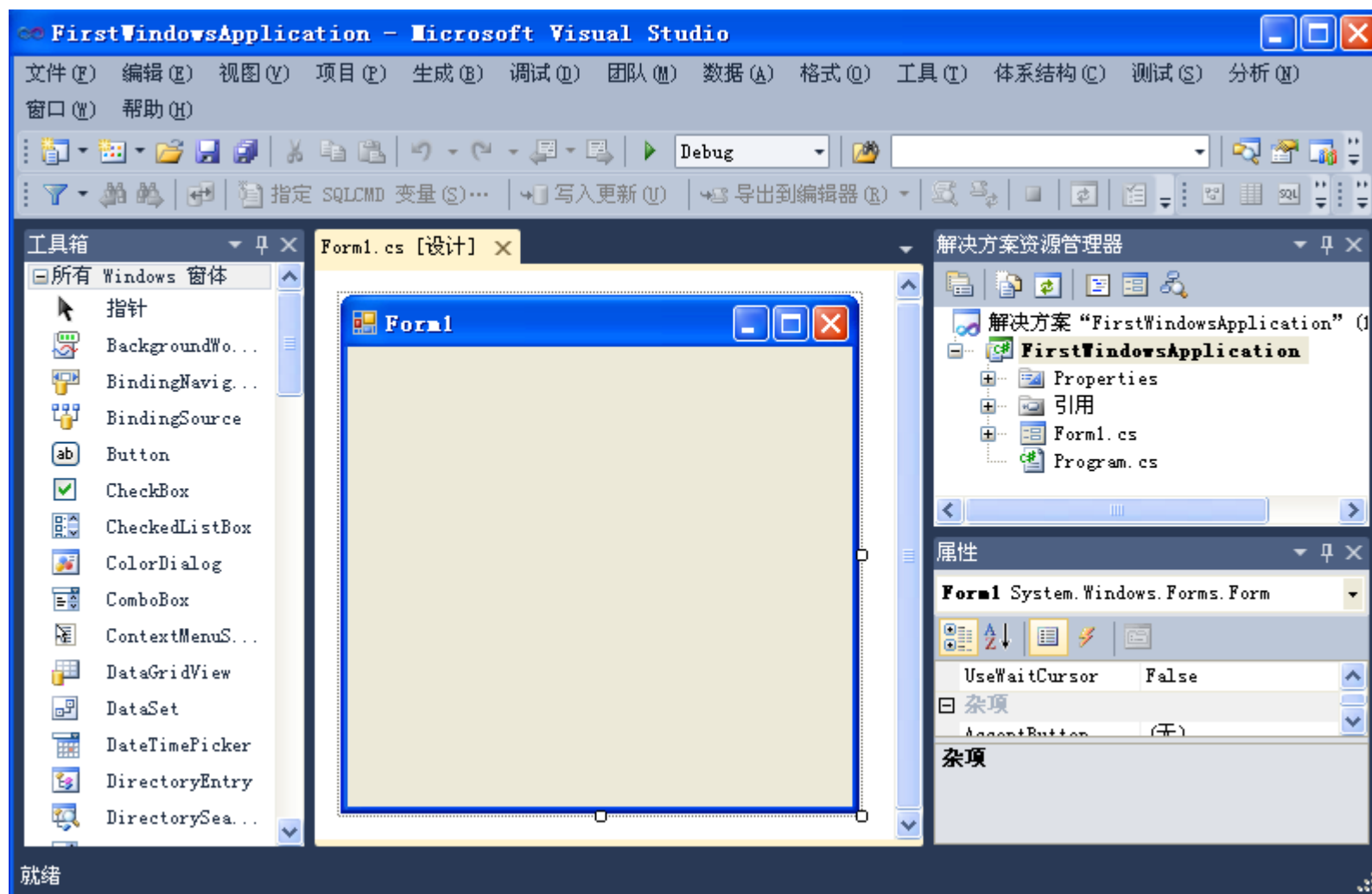
- ▶ 基于控制台的应用程序
- ▶ 基于对话框的应用程序
- ▶ 单文档应用程序
- ▶ 多文档应用程序
- ▶ 基于html的应用程序

## ▶ C#

- ▶ 控制台应用程序
- ▶ Windows窗体应用程序
- ▶ WPF应用程序
- ▶ ASP.NET Web应用程序
- ▶ WCF服务应用程序
- ▶ .....

# Windows窗体应用程序

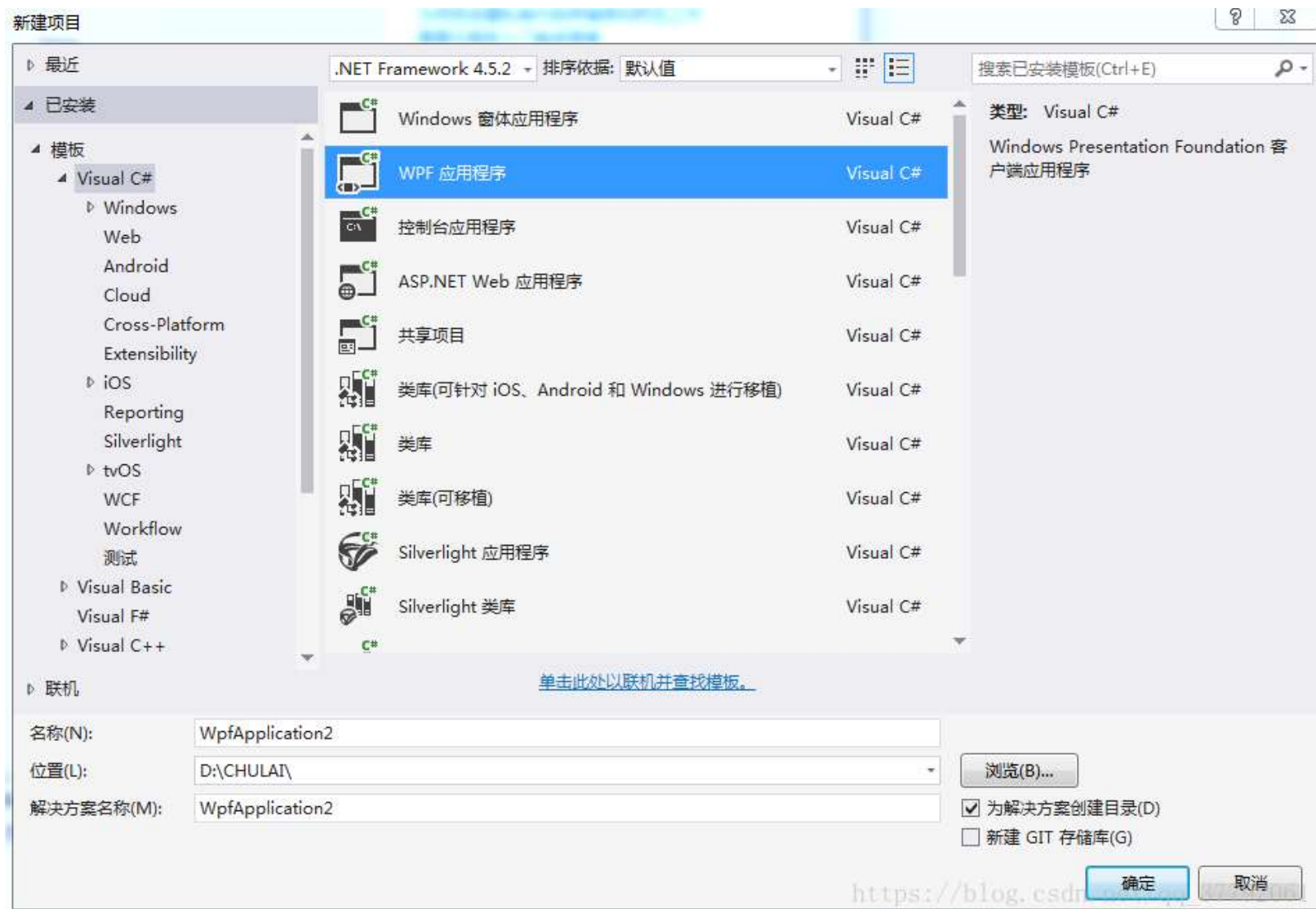


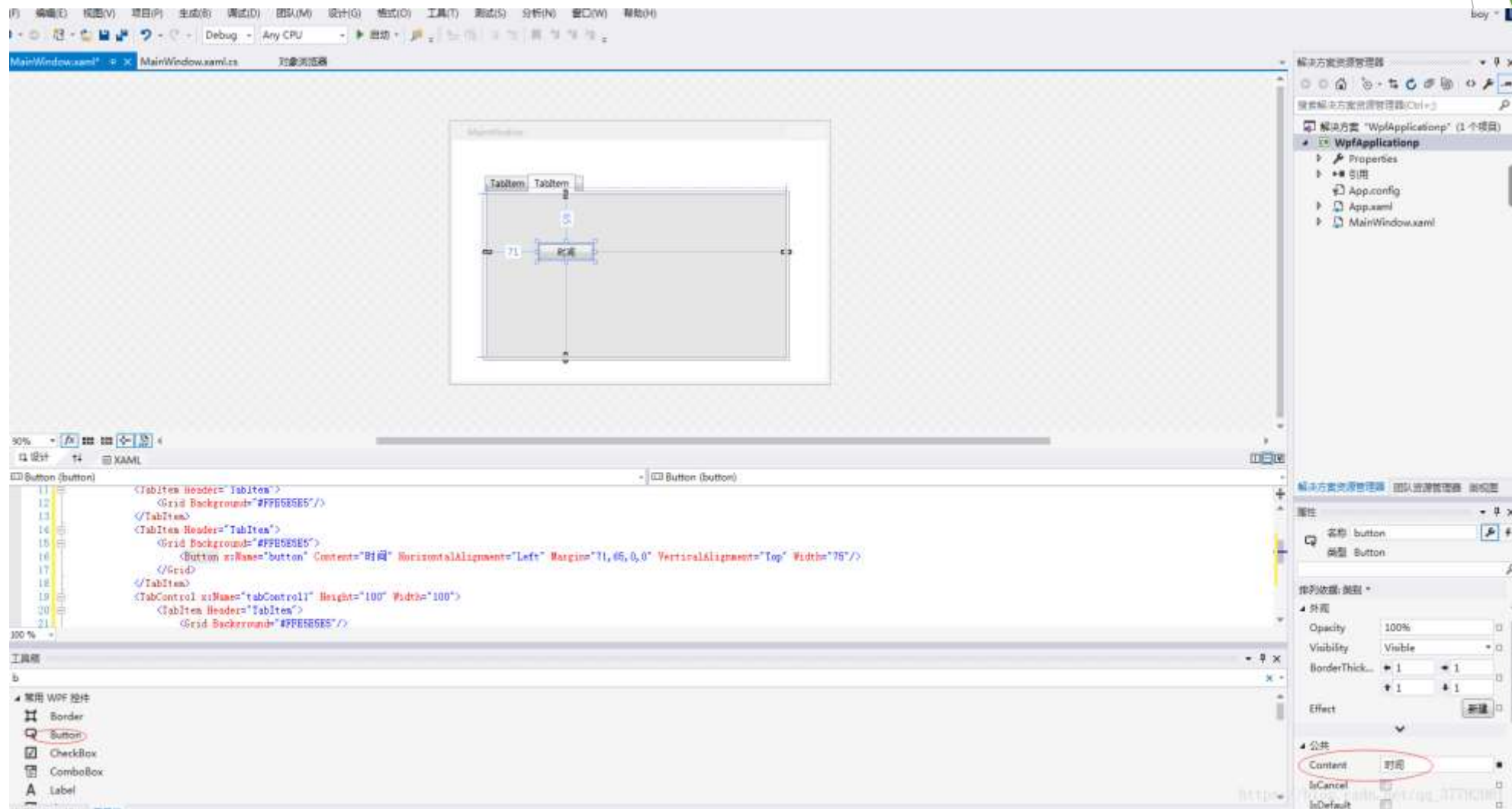


# WPF 应用程序

程序界面：基于XML的XAML语言定制；

程序逻辑：C#语言实现





# 上机练习作业

- ▶ 熟悉Winform窗体应用程序创建流程
  - ▶ 掌握基本控件如button, label, textbox, listbox, dataGridView, combobox的用法
- ▶ 熟悉WPF窗体应用程序创建流程
  - ▶ 掌握基本控件如button, label, textbox, listbox, dataGrid, combobox的用法