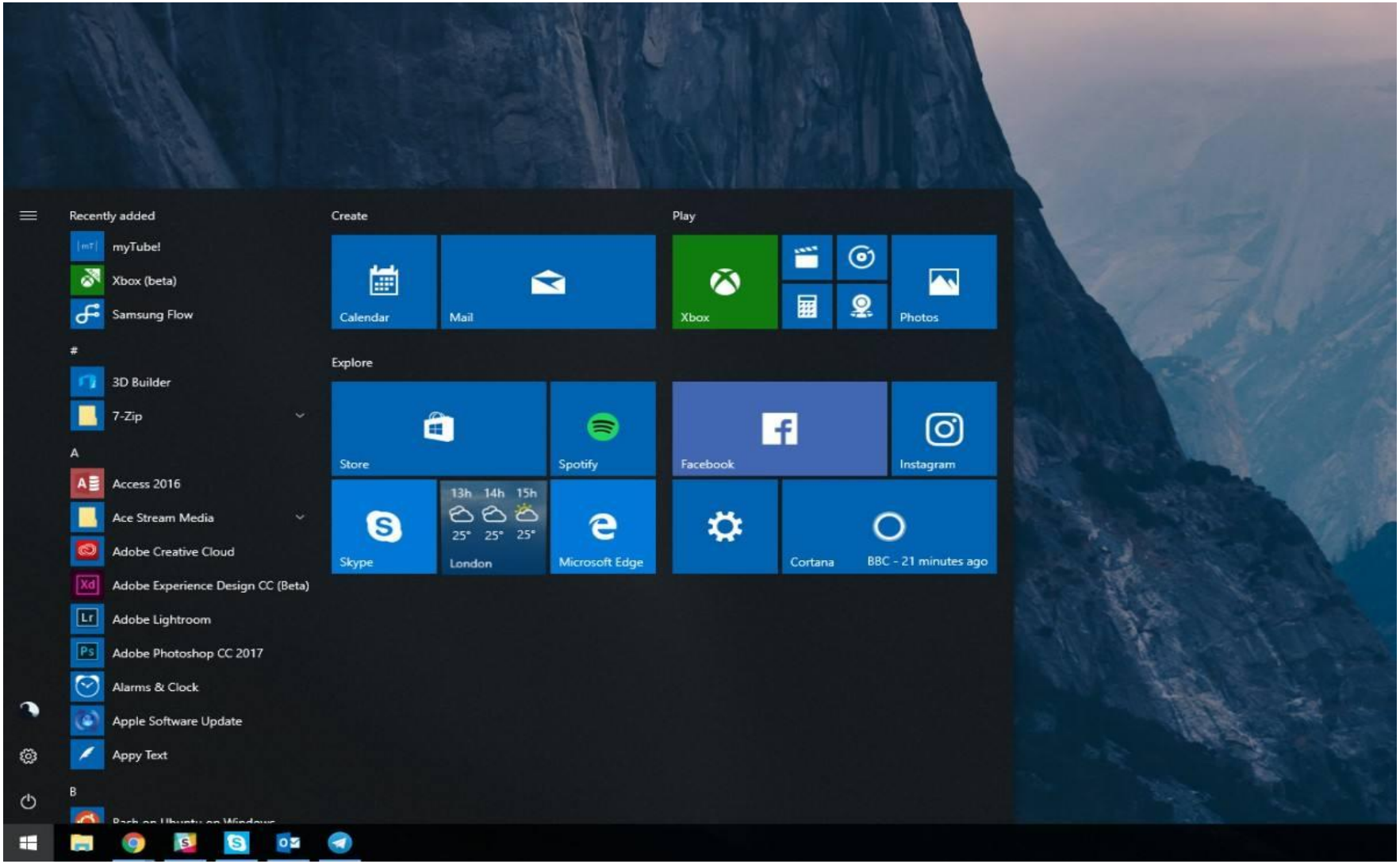
第1讲 Windows系统_编程模式_程序结构



Windows是一个具有可视化图形用户界面的多任务的操作系统，它为用户提供了风格统一的由窗口、菜单、工具栏等界面元素所构成的多任务环境。

一、Windows操作系统介绍

1 Windows操作系统的发展及简介

1.Microsoft公司早在1985年和1987年分别推出Windows 1.03版和Windows2.0版。

2.1990年5月推出Windows3.0 。

3.1991年推出Windows3.1 。

4.1995年推出新一代操作系统Windows95 ,是操作系统发展史上一个里程碑式的作品 。

2 Windows操作系统的特点

- 直观、高效、统一的面向对象的图形用户界面，易学易用
- 丰富的设备无关的图形操作
- 多任务多进程

3 Windows的内存管理

Windows系统采用虚拟内存，在操作系统中有两种存贮器，物理存贮器和虚拟存贮器 。

虚拟内存系统的工作过程如下:

1. 当建立一个新进程时，操作系统赋予每个进程2GB的虚拟地址（不是内存）。

2. 虚拟内存管理器把应用程序的代码映射到应用程序的虚拟地址空间的某个位置，并把当前所需要的代码加载到物理内存。

3. 如果应用程序使用了动态链接库(DLL)，则被映射到进程的虚拟地址空间，并在需要时加载到物理内存。

4. 程序的数据存储和栈操作等需要的空间在物理内存中分配，并映射到虚拟地址空间。

5. 应用程序使用虚拟地址空间中的地址开始执行，虚拟内存管理器把每个对内存的访问映射到物理内存的某个位置，从而访问数据。

二、Windows编程模式

由于Windows 操作系统完全不同于单任务的MS-DOS操作系统，Windows程序设计有着完全不同模式，具有图形用户界面、多任务、多窗口等特点。

Visual C++提供两套完整的Windows 程序开发系统：

- 直接使用Windows提供的API 函数；
- 采用面向对象的方式，使用微软基础类库MFC 。

1 Windows API简介

Windows API是Windows 系统和Windows 应用程序间的标准接口，为应用程序提供Windows支持的函数定义、参数定义和消息格式等。

标准Win32 API 函数可以分为以下几类：

1) 系统服务
2) 通用控件库
3) 图形设备接口
4) 网络服务
5) 用户接口
6) 系统Shell
7) Windows 系统信息

2 Windows程序设计的基本概念

1.窗口

-----窗口是Windows程序的基本操作单元，是应用程序与用户之间进行交互的接口，也是系统管理应用程序的基本单位。

编写应用程序其实就是创建一个或多个窗口，程序的执行过程是窗口内部、窗口与窗口之间以及窗口与系统之间进行数据交换与处理的过程。

在winuser.h中定义了代表窗口的WNDCLASS结构类型。

```
typedef struct tagWNDCLASS {
    UINT style;                         // 窗口风格
    WNDPROC lpfnWndProc;                //指向窗口处理函数的函数指针
    int cbClsExtra;                     //窗口结构中的预留字节数
    int cbWndExtra;                     // 本窗口创建的其它窗口结构中预留字节数
    HINSTANCE hInstance;                //注册该窗口类的实例句柄
    HICON hIcon;                        //代表该窗口类的图标句柄
    HCURSOR      hCursor;               // 该窗口客户区鼠标光标句柄
    HBRUSH       hbrBackGround;         //该窗口背景颜色句柄
    LPCSTR       lpszMenuName;          //指向窗口菜单名的字符指针
    LPCSTR       lpszClassName;         //指向窗口名的字符指针
  } WNDCLASS, *PWNDCLASS, NEAR *NPWNDCLASS, FAR *LPWNDCLASS;
```

2.句柄和Windows对象

Windows对象-----窗口、图标、光标、菜单及正在运行的应用程序实例等等。

句 柄-----系统用来唯一标识某个Windows对象的一个无符号整数，相当于Windows对象的名字，应用程序只有通过句柄才能使用Windows对象。

应用程序是通过恒定不变的句柄来访问程序中各个对象的。这是多任务操作系统对多个进程进行管理的基本手段。

Windows常用句柄类型如下所示：

句柄类型	说明	句柄类型	说明
HANDLE	一般类型句柄	HBRUSH	画刷类型句柄
HWND	窗口类型句柄	HDC	设备描述表类型句柄
HINSTANCE	程序实例句柄	HICON	图标类型句柄
HCURSOR	光标类型句柄	HMENU	菜单类型句柄
HFONT	字体类型句柄	HBITMAP	位图类型句柄
HPEN	画笔类型句柄	HFILE	文件类型句柄

3. 事件与消息

事件-----在Windows环境下，应用程序启动后，系统等待用户在图形用户界面内的输入选择，如鼠标按键、键盘按键、窗口被创建、关闭、改变大小、移动等。

只要有事件发生，系统即产生特定的消息，消息描述了事件的类别，包含了相关信息，Windows应用程序利用消息与系统及其它应用程序进行信息交换。

由于windows事件的发生的是随机的，程序的执行先后顺序也无法预测，系统采用消息队列来存放事件发生的消息，然后从消息队列中依次取出消息进行相应的处理。

在winuser.h中，消息结构的定义如下：

```
typedef struct tagMSG {
        HWND    hWnd;           //指定消息发向的窗口句柄
        UINT    message;        //标识消息的消息值
        WPARAM  wParam;         //消息参数
        LPARAM  lParam;         //消息参数
        DWORD   time;           //消息进入队列的时间
        POINT    pt;            //消息进入队列时鼠标指针的屏幕
    坐标
    }MSG,*PMSG,NEAR *NPMSG,  FAR * LPMSG;
```

下面介绍消息结构中各个成员的意义：

a）message是标识消息的消息值或消息名。每个消息都有唯一一个数值标识，常用不同前缀的符号常量以示区别。例如，WM_表示窗口消息 。

Windows常用的窗口消息和消息值定义，定义于winuser.h中：

```
#define   WM_CREATE       0X0001   //创建窗口产生的消息
#define   WM_DESTROY      0X0002   //撤销窗口产生的消息
#define   WM_PAINT        0X000F   //重画窗口产生的消息
#define   WM_CLOSE        0X0010   //关闭窗口产生的消息
#define   WM_CHAR         0X0102   //按下非系统键产生的字符消息
#define WM_USER           0X0400   //用户自定义消息
```

b) wParam和lParam都是32位消息参数，其数据类型在windef.h中定义如下：

```
typedef     UINT        WPARAM;
typedef     LONG        LPARAM;
```

c) pt表示消息进入消息队列时鼠标指针的屏幕坐标，POINT是定义在windef.h中的结构体，表示屏幕上一个点：

```
 typedef struct tagPOINT {
LONG      x;                  //表示点的屏幕横坐标
    LONG y;                   //表示点的屏幕纵坐标
  } POINT,  PPOINT,NEAR *NPPOINT,FAR *LPPOINT;
```

4. 进程与线程

32位Windows提供“进程-线程-窗口”的模式来管理运行的程序。

“程序”指存储在介质上、含有指令和数据的文件，是一个被动的对象；

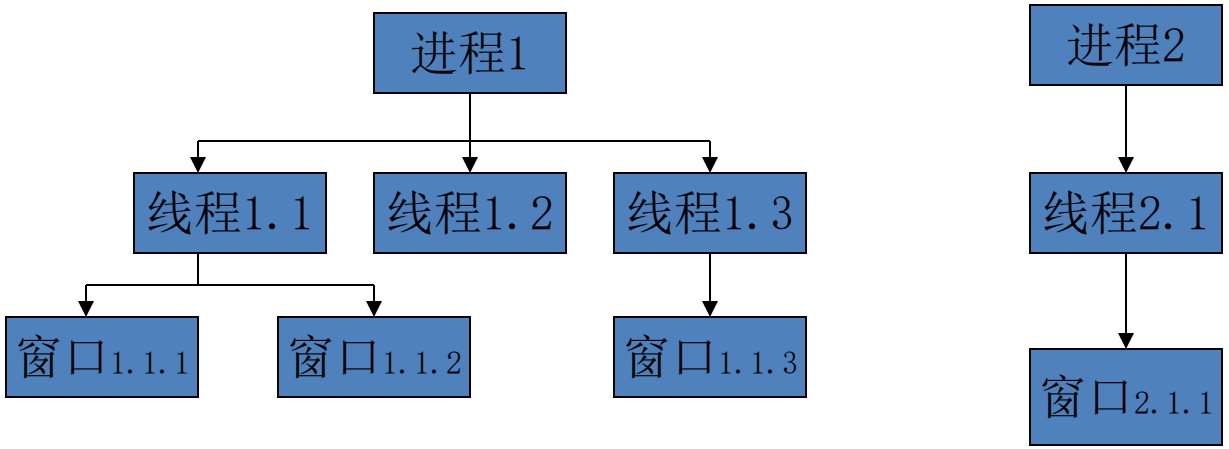
“进程”是一个被运行的程序的实例，代表了内存中正在运行的指令和系统资源，如地址空间、打开的文件等，是一个主动对象。

“进程”又分为“动态”和“静态”的两部分，其中“动态”部分形成“线程”概念。

进程并不执行代码，它只是线程的容器，拥有系统资源和私有资源。

当创建一个进程的同时也创建了第一个线程，称为“主线程”，线程是进程主动的关键。主线程的入口函数是WinMain。之后，主线程可以在进程中再创建一个或多个线程，即“多线程”，多个线程共享进程的静态部分。使用进程-线程模式，是因为创建一个线程比一个进程所需资源要少，且线程间共享进程地址空间，简化了进程间的通信，可提高程序执行效率。

窗口是应用程序和用户进行交互的接口，窗口总是由线程拥有，一个线程可以拥有一个或多个窗口，也可没有窗口。关系如下图所示：



5. 资源共享

不同之处：

DOS程序：在运行时独占系统的全部资源，包括显示器、内存等，直到程序结束时才释放资源。

Windows：它是一个多任务的操作系统，各个应用程序共享系统提供的资源。

Windows的基本模式是：

向Windows系统请求资源 ；

使用该资源；

释放该资源给Windows以供别的程序使用。

注意：

1. 最容易忽略的是第三步，如果忽略了这一步，会出现程序运行出现异常情况，或干扰其它程序正常运行，甚至造成立即死机，如设备上下文没有释放时。
2. CPU也是一种非常重要的资源，应用程序应避免长时间的占用CPU资源；如果确实需要这样做，也应当采取一些措施，以让程序能够响应用户的输入。
3. 主存也是一个共享资源，要防止同时运行的多个应用程序因协调不好而耗尽内存资源。

6. 数据类型

包括简单类型和结构类型，常用数据类型说明如下：

数据类型	说　　明
BYTE	8位无符号字符
BSTR	32位字符指针
COLORREF	32位整数，表示一个颜色
WORD	16位无符号整数
LONG	32位有符号整数
DWORD	32位无符号整数
UINT	32位无符号整数
BOOL	布尔值，值为TRUE或FALSE

三、Windows程序结构

1.Windows应用程序的构成

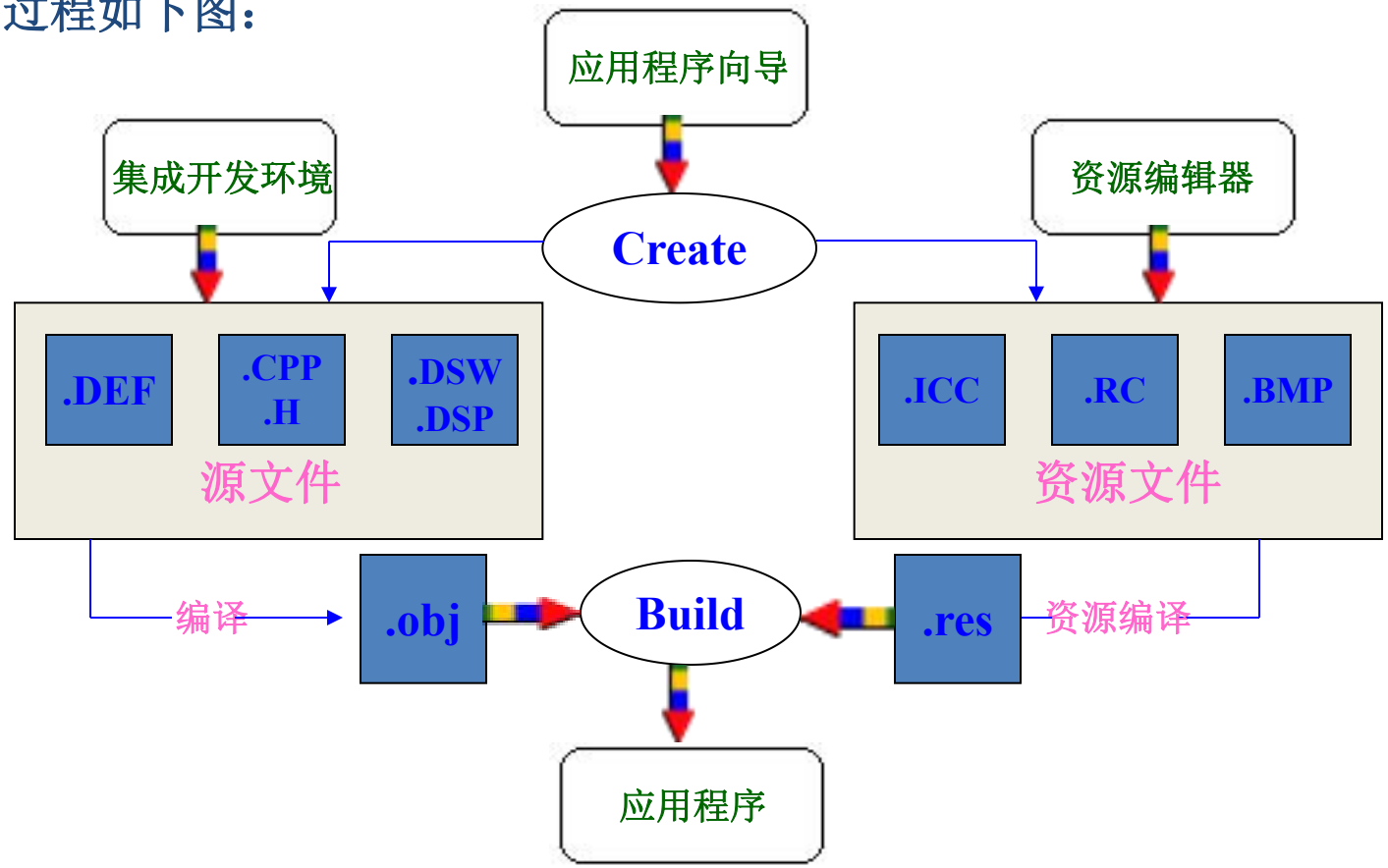
- **组成：**程序代码、用户界面资源和动态链接库库模块(dynamic-link library) 。
  - 程序代码主要由可执行代码构成
  - 资源是应用程序所能够使用的一类预定义工具，用户界面资源是应用程序定义的图形用户接口GUI (graphical user interface)，如菜单、对话框、按钮等。

- 常用资源如下：

资 源类型	说明
快捷键(Accelerator)	存贮击键和命令组合
位图 (Bitmap)	包括Windows匹配格式的图形
对话框(Dialogbox)	对话框控件细节、布局、属性
图标(Icon)	存贮图标用位图组
菜单(Menu)	菜单和菜单组文本和布局细节
字符串表(String table)	存字符串和相关标识符(ID)值
工具栏(Toolbar)	工具栏布局细节和画按钮所用的位图
版本消息 (Cursor)	程序状态消息，如程序名，作者，拷贝日期 版本号光标包括画光标用的特殊位图

Windows应用程序的创建与控制台应用类似，要经过编译、链接两个阶段，要增加资源编辑、编译过程。

过程如下图：



2. 应用程序的执行过程

(1). 操作系统接收命令后，创建新的进程和初始线程 (initial thread)。

(2). 应用代码加载内存。

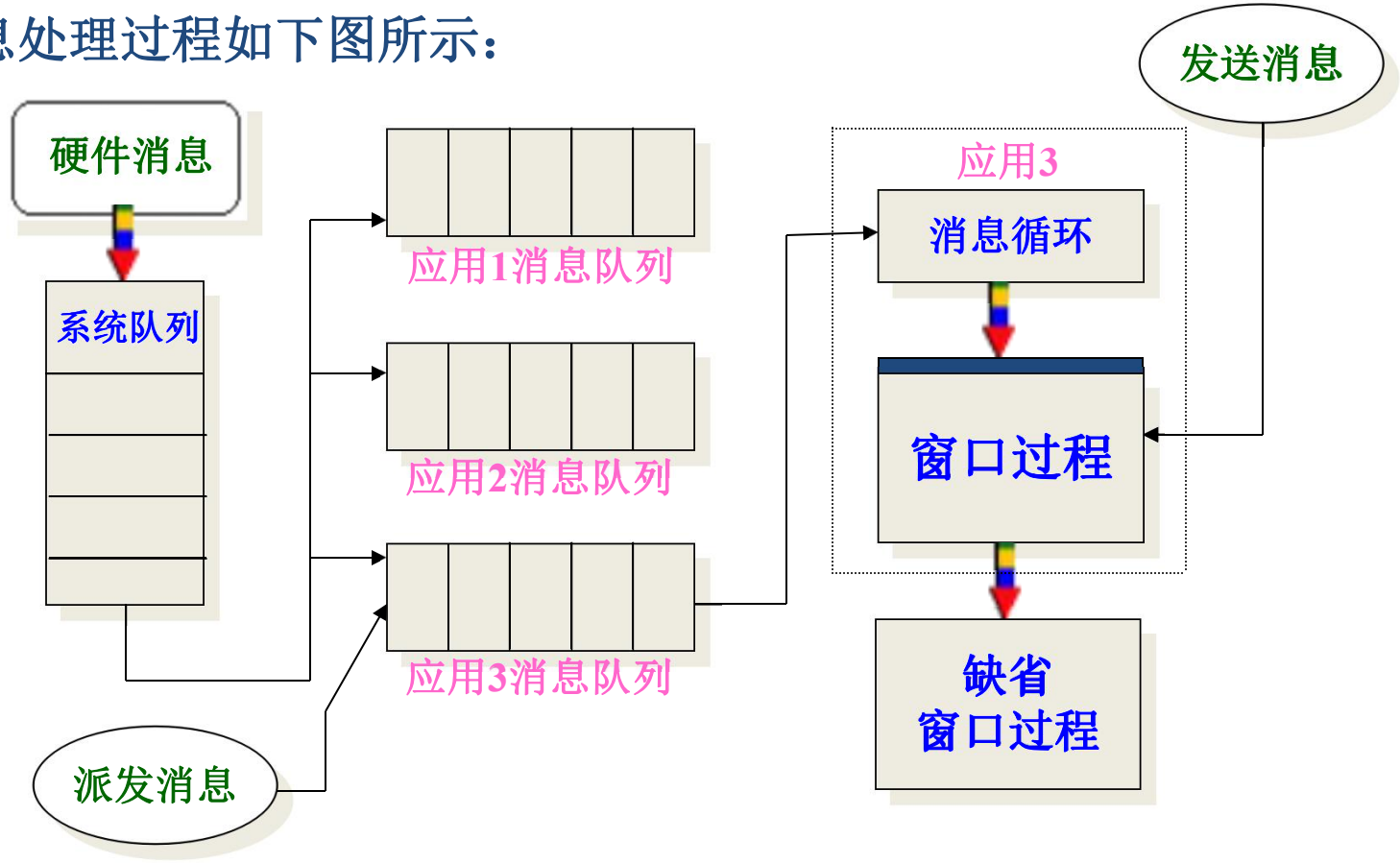
(3). 如应用了DLL，则加载DLL到内存。

(4). 为项目空间(如数据和堆栈)分配物理内存，并映射到虚拟地址空间。

(5). 应用程序开始执行。

3. 处理消息

消息处理过程如下图所示：



(1). 当Windows应用程序开始执行，操作系统在预定的入口WinMain与应用程序连接；

(2). 应用程序创立一个或更多的窗口，每个窗口（包括对话框）都有一个窗口过程函数，程序通过窗口接收用户的输入和决定窗口显示；

(3). 各种输入产生的消息首先进入系统队列，由系统根据产生消息的窗口将消息发至窗口所在应用程序的消息队列；

(4). 程序的消息循环代码段检索从消息队列中来的消息，并把它们返回Windows，由Windows发给适当的窗口过程函数进行处理，对于程序未响应的消息，发给缺省窗口过程函数进行处理。

3.MessageBox 是一个API函数，显示一个消息框,其原型为：

```
int WINAPI MessageBox(HWND hWnd,
LPCSTR lpszText,
LPCSTR lpszCaption,
UINT uType);
```

第一个参数指明此消息框的父窗口句柄， 为NULL则说明没有父窗口；

第二、三个参数类型均为字符型指针，分别指向消息框中要显示的字符串和消息框标题栏 显示的字符串；

第四个参数是一个无符号整数，表明消息框中显示的按钮和风格。

Windows已有定义：

```
#define MB_OK                       0x00000000L       // OK按钮
#define MB_OKCANCEL                 0x00000001L       // OK和CANCEL两个按钮
#define MB_ABORTRETYRIG NORE  0x00000002L       // ABORT、RETRY 、//IGNORE三个按钮
#define MB_YESNOCANCEL              0x00000003L       // YES、NO、CANCEL三//个按钮
```

- **uType参数**
- 即可取这些定义值或是它们按位’或’运算的组合。

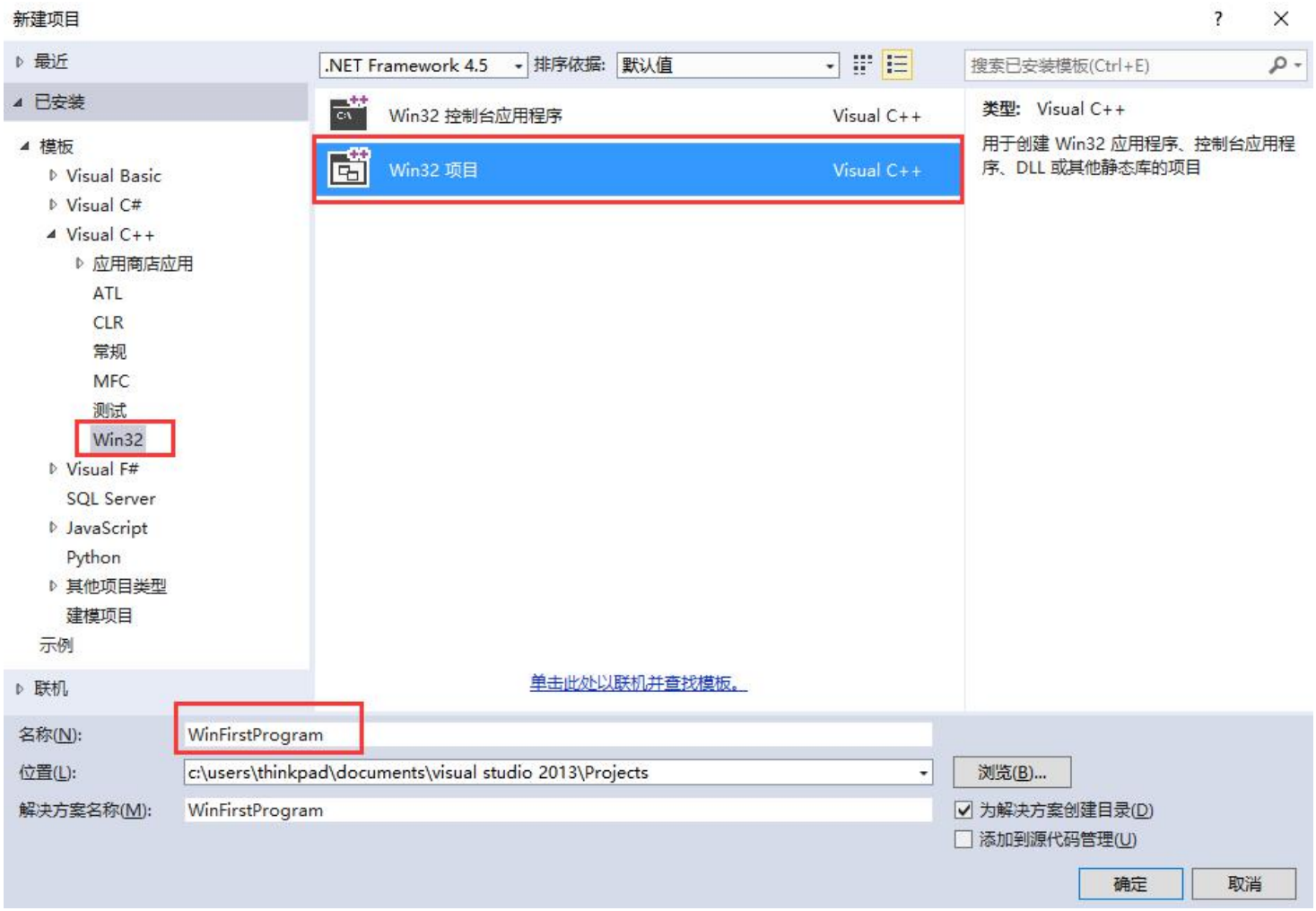
- **MessageBox返回所按下按钮的ID值。**

即Windows用一个无符号整数唯一表示某个资源(对话框、按钮、菜单等等)，并定义一个唯一的符号常量与之对应，称为资源的ID值。

windows内部定义的部分资源标识：

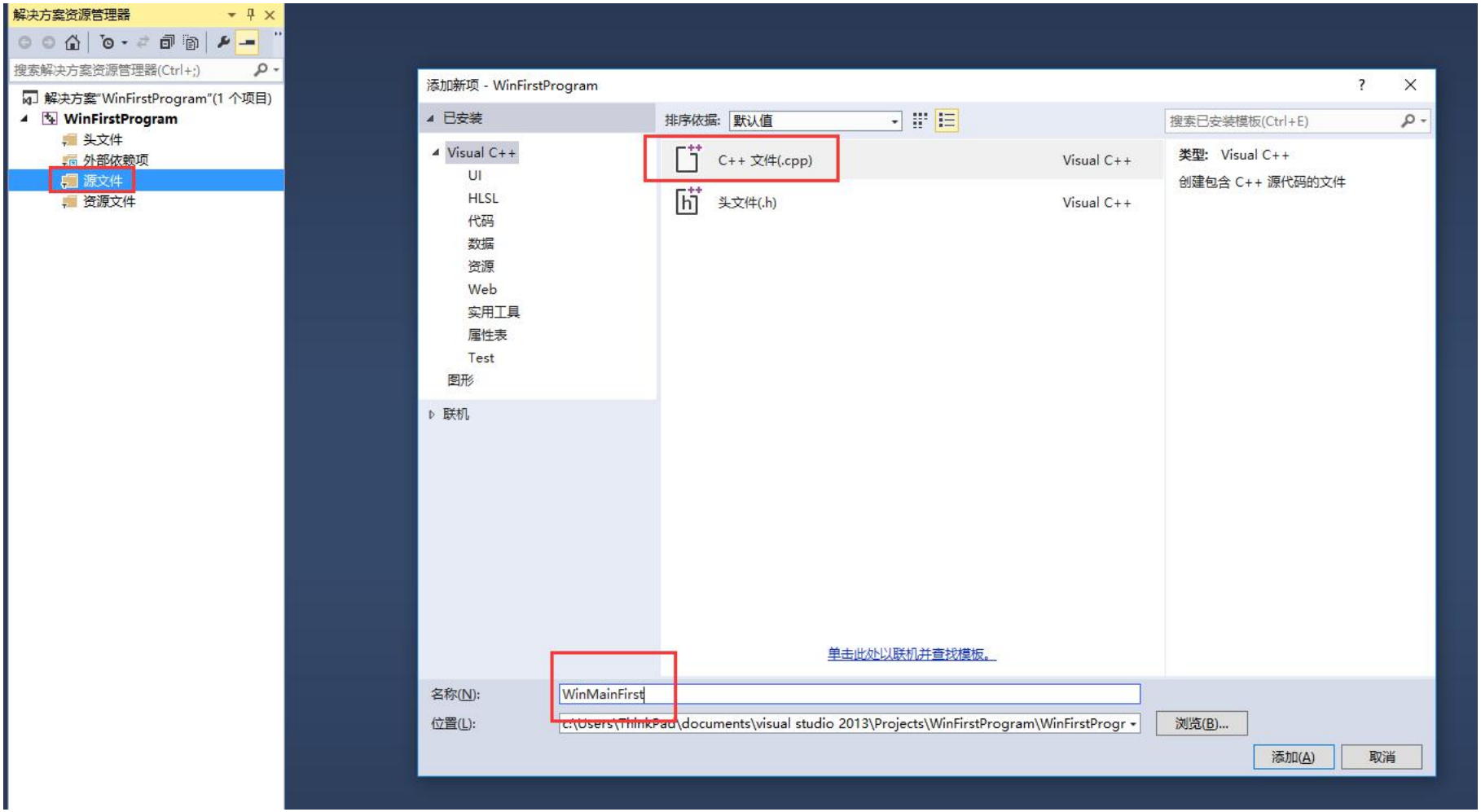
```
#define IDOK            1       // OK按钮ID值
#define IDCANCEL        2       // CANCEL按钮ID值
#define IDABORT         3       // ABORT 按钮ID值
#define IDRETRY         4       // RETRY按钮ID值
#define IDIGNORE        5       // IGNORE按钮ID值
#define IDYES           6       // YES按钮ID值
#define IDNO            7       // NO按钮ID值
```

Windows程序设计第一个案例实战1



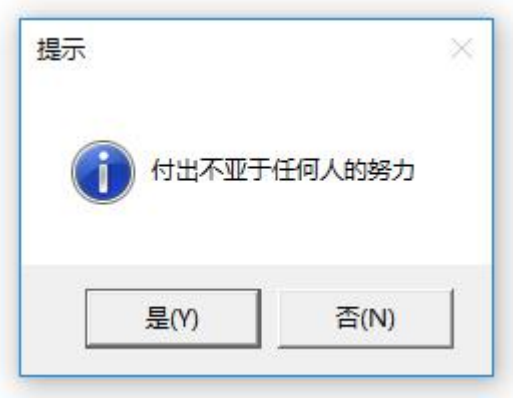
Windows程序设计第一个案例实战1

Windows程序设计第一个案例实战1



Windows程序设计第一个案例实战

```
1  #include <Windows.h>
2  #pragma comment(lib,"winmm.lib")
3
4  int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow)
5  {
6      PlaySound(L"郭少杰 - 一曲红尘.wav", NULL, SND_FILENAME | SND_ASYNC);    //播放音效
7
8      MessageBox(NULL, L"付出不亚于任何人的努力", L"提示", MB_YESNO|MB_ICONINFORMATION);//显示一个消息框
9
10     return 0;
11 }
12
```




梦想在这儿起飞！！！