

这些道理没人告诉  
你！-如鹏网精华电  
子书

rupeng.com



## 《C 语言也能干大事》第一节

## 板书:

## 1、

C 语言不是只能开发黑底白字的控制台界面程序。学习本课的基础：本课程不讲 C 语言基础知识，C 语言考及格，学的过程中配合复习，有不懂的回去翻书。

## 2、

学完本课程以后同学们做的一些作品赏析：计算器、俄罗斯方块、音乐播放器、学生管理系统（<http://www.rupeng.com/forum/thread-2487-1-1.html>），邮件客户端（<http://www.rupeng.com/forum/thread-3697-1-1-uid7.html>），Web 论坛（<http://www.rupeng.com/forum/thread-3991-1-1-uid7.html>）可以开发游戏、网络程序、网站程序等。C 语言什么都可以开发

## 3、

准备工作，安装 VC6（在网上搜“Visual C++ 6”，找合适的地址下载）。VC6 在 Vista、Windows 下启动可能有兼容性提示。

## 3、开始做第一个 C 语言的程序

【文件】→【新建】，打开【工程】选项页，选择【Win32 Application】（不要选错了），在“位置”中修改工程保存的路径。（以前用 Turbo C 的同学会问“什么叫工程？”。在以前大家用 TurboC 写的程序一般也就是一个文件，但是用 C 语言写大程序的时候不可能把所有的代码都写在一个文件中，肯定要写很多文件。“工程（Project）”就是这些文件的一个集合。）在“工程名”中写入你为这个工程取的名字，最好有一定意义。下一步【一个简单的 Windows 程序】

## 4、向 Windows 的世界问好

```
MessageBox(NULL,TEXT("世界你好"),TEXT("问好"),MB_OK);
```

VC6 在 Vista、Windows 下编译调试时可能有兼容性提示。

## 5、学本课程常见问题

### (1) 为什么不用 TurboC?

Turbo C (Win-TC) 太老了, 界面还是 DOS 界面, 库也是 DOS 的库, 没有网络开发包、没有 Windows 界面开发包。TC 只能用来学 C 的基础, 要进行实际开发必须使用先进一点的工具。还是让 TurboC 进博物馆吧。

### (2) 为什么不用更高版本的 VC?

我们不使用高版本的 VC 主要是如下考虑: VC6 和高版本的 VC 没有本质区别 (VC6 是 1998 年的东西, 大部分人、公司还是用 VC6, 市面上的书也都是用 VC6), 只不过是升级了一下 C++ 的版本、STL 的版本而已, 对咱们这次学习没有什么影响; VC6 的操作方式和 VC2008 非常相似, 用熟了 VC6 再学高版本的 VC 只用半小时而已; 高版本的 VC 体积太大, 要五六百兆, 而上面给出的 VC6 只有 100MB。VC2008 下建立工程。

### (3) 学 C 语言还是学其他语言, 用其他语言实现岂不是更好?

用最熟悉的语言深入编程的各个领域 (网络、数据库、图形学、内核等), 一通百通, 一切语言都是纸老虎。参考资料如鹏网首页。

## 6、深入研究 MessageBox

### (1) `MessageBox(NULL,TEXT("世界你好"),TEXT("问好"),MB_OK);`

第一个参数暂时不讲; 第二个是正文; 第三个是标题 (Caption); 第四个是类型。

MB\_OK 就是表示有一个 OK 按钮【确定】)

TEXT 是一个宏, 当字符串中有中文的时候最好用 TEXT 来包围这个字符串, 虽然

不使用 TEXT 在 VC6 中没问题，但是在 VC7 中有问题，而且微软也建议使用 TEXT 宏，因此在涉及到中文的场合要使用它。他的作用就是把中文转化成不会乱码的格式。（暂时这么认为）。\_T("问好"), 其实\_T 只是 TEXT 的一个缩写而已。而且\_T 在有的低版本里不识别。TEXT()低版本也识别。暂时不用关心 TEXT 的细节。

换个样子：

```
MessageBox(NULL, TEXT("世界你好"), TEXT("你好"),MB_OKCANCEL);
```

可从设置希望在对话框中显示的按钮：

```
#define MB_OK  
0x00000000L  
  
#define MB_OKCANCEL  
0x00000001L  
  
#define MB_ABORTRETRYIGNORE  
0x00000002L  
  
#define MB_YESNOCANCEL  
0x00000003L  
  
#define MB_YESNO  
0x00000004L  
  
#define MB_RETRYCANCEL  
0x00000005L
```

也可以设置对话框中显示的图标：

```
#define MB_ICONHAND  
0x00000010L
```

```
#define MB_ICONQUESTION  
0x00000020L
```

```
#define MB_ICONEXCLAMATION  
0x00000030L
```

```
#define MB_ICONASTERISK  
0x00000040L
```

比如： `MessageBox(NULL, TEXT("世界你好"), TEXT("你好"),  
MB_ICONQUESTION);`

(2) 可是如果我想显示“确定、取消”按钮的时候同时使用问号图标呢？

```
MessageBox(NULL, TEXT("世界你好"), TEXT("你好"), MB_OKCANCEL |  
MB_ICONQUESTION);
```

```
MessageBox(NULL, TEXT("世界你好"), TEXT("你好"),  
MB_ABORTRETRYIGNORE | MB_ICONEXCLAMATION);
```

(3) 思考：

到这一步如果你只是在背诵“第一组和第二组中间放一个竖线就可以实现两种效果的组合形式”，那么你未来只会成为一个代码工人。我们要思考一下其背后的原理。

“|”是什么意思？“|”是位运算里的“或”运算，只有对应的两个二进位有一位为 1 时，结果位才为 1，否则为 0。MB\_OK、MB\_OKCANCEL 等的低四位不同，但是高位永远为 0；MB\_ICONHAND、MB\_ICONQUESTION 等的低 5 至第 8 位不同，而其他位永远为 0。这样“MB\_OK、MB\_OKCANCEL”组的数值与“MB\_ICONHAND、MB\_ICONQUESTION”组的数值进行或运算后能分别保留各自的部分，也就是在结果值中同时体现两组的取值。

这种风格叫“掩码”，在 Windows 编程中这种用法会经常用到。

#### (4) 返回值

MessageBox 是有返回值的，返回值为用户点击的按钮：

```
#define IDOK
1

#define IDCANCEL
2

#define IDABORT
3

#define IDRETRY
4

#define IDIGNORE
5

#define IDYES
6

#define IDNO
7
```

代码：

```
int ret = MessageBox(NULL, TEXT("你是人吗？"), TEXT("火星人"),
```

```
MB_YESNO | MB_ICONQUESTION);
```

```
if(ret==IDYES)
```

```
{  
  
    MessageBox(NULL, TEXT("火星人你好"), TEXT("问好"), MB_OK);  
  
}  
  
else  
  
{  
  
    MessageBox(NULL, TEXT("欢迎回家来"), TEXT("问好"), MB_OK);  
  
}
```

6、课后作业：自己动手写恶搞程序。

运行以后弹出询问对话框（有【是】、【否】两个按钮以及问号图标）“你是好人吗？”，如果点击【是】，则弹出对话框（只有一个【确定】按钮以及一个警告图标）“你看你就不像好人，点击【确定】开始格式化 C 盘！”；如果点击【否】，则弹出对话框（有【重试】、【取消】两个按钮）“尝试把你变成好人失败，是否重试？”。

别把教学视频当电影看！！！多练！！！多写

## 《C 语言也能干大事》第二节

板书：

运行以后弹出询问对话框（有【是】、【否】两个按钮以及问号图标）“你是好人吗？”，如果点击【是】，则弹出对话框（只有一个【确定】按钮以及一个警告图标）“你看你就不像好人，点击【确定】开始格式化 C 盘！”；如果点击【否】，则弹出对话框（有【重试】、【取消】两个按钮）“尝试把你变成好人失败，是否重试？”。

### 1、怎么生成 exe 程序。怎么把做好的 exe 程序发给别人？

在我的电脑里直接进入工程文件夹。工程目录下生成的 debug/release 文件就是生成好的可执行文件，发给别人就可以。Debug 和 Release 版本区别。怎么生成 Release 版本。

#### 1、对话框程序

VC6 下安装 Win32 DialogBased Application

<http://down1.rupeng.com/download/software/Win32DlgBasedAppWizard.zip>

VC2008 下安装 Win32 DialogBased Application

<http://down1.rupeng.com/download/software/DlgBasedAppWizardVS2008.zip>

如鹏的 Win32 DialogBased Application 没有做神奇的事情，只是简化了创建的过程，大家也可以手工创建，创建的代码可以在没有安装 Win32 DialogBased Application Wizard 的机器上用。演示在 2008 上从 win32application 创建一个 Win32 DialogBased Application。

在资源编辑器中编辑对话框界面并且拖放控件上去。试试常用的那些控件

什么是属性。

作业：开发一个注册界面。

#### 2、常见问题



(1) 开发 Windows 程序有什么前途？不要急于求成，用自己最熟悉的语言探索编程的各个方面，内核级开发、数据库、网络、动态网站开发等。学《C 语言也能干大事》的目的并不是直接找开发 Windows 程序的工作，只要锻炼了动手能力和一通百通的数据库、网络、动态网站开发、面向对象等技能，学什么东西、找什么工作都很轻松。

(2) 学 MFC 不是更好？MFC 乱七八糟的东西太多，隐藏了太多，隐藏不仅没有有利于初学者学习，反而使得更复杂。MFC 只是一个稍微封装的 SDK，学会了这里的课程如果又需要再去学 MFC 的话会发现 90%是相同的。

<http://www.rupeng.com/forum/thread-338-1-1.html>

3、对话框中图片资源的应用在资源视图中，选择“引入”，然后文件过滤器选择“\*. \*”，找到 bmp 图片文件。选中图片空间，要点击边缘。拖一个 IMAGE 控件过来，并且修改 IMAGE 控件的 Type 属性为 BitMap，在控件的 Image 属性中选择刚才添加的 BITMAP 资源。jpg 的格式怎么转换为 bmp 格式的呢？用画图工具（mspaint）。图片被编译到了程序中，不再依赖于源文件

#### 4、菜单资源

制作好菜单资源后，在对话框的 Menu 属性中选择刚才制作好的对话框。

子菜单项（选中【弹出】）、快捷键（在要担当快捷键的字母前加&）、分隔符

5、只有控件、菜单，可以点击控件、菜单以后的动作怎么做呢？就要写代码了。后面会讲

### 《C 语言也能干大事》第三节

板书：

1、怎么修改程序图标，引入一个 ico 文件（可以用一些工具制作 ico 图标，网上搜“制作 ico”），然后命名 ID 为 1，重新构建。

2、在编写控制台程序的时候一切流程都是有先后关系、并行的，而且所有函数都是由我们来调用的，比如下面的实例性代码：

```
printf("确定请输入 y，取消输入 n");

char c = getchar();//scanf

if(c=='y')

{

///

}

else if(c=='n')

{

///

}

Void func1()

{

MessageBox(NULL, TEXT("按钮 1 被按下了")...);

}

Void func2()

{
```

```
MessageBox(NULL, TEXT(“按钮 2 被按下了”)….);  
  
}
```

我们可以用 `getchar` 来等待用户输入一个值。但是到了 Windows 编程中就不一样了，同一时刻用户即可能点击【OK】按钮，又可能点击【Cancel】按钮，又可能在文本框中输入几个字，还可能在窗口上双击几下，这样就无法同时等待用户的这些动作。为了解决这个问题，Windows 引入了消息机制（也可以叫做回调机制或者事件机制）。在程序启动的时候把函数 `func1` 要响应【OK】按钮 1 的点击动作、函数 `func2` 要响应【Cancel】按钮的点击动作、函数 `func3` 要响应窗口的双击动作等等这些信息告诉 Windows，然后当用户执行相应操作的时候 Windows 就会来主动调用你注册的函数，主动通知你。不再是程序调用操作系统的函数，而是操作系统反过来调用你的函数。Don't call me ,I'll call you!（也被人称为“好莱坞法则”）。

2、关于上面的这个问题要慢慢来理解，下面就来通过第一个例子来初步理解这个 Don't call me ,I'll call you!

//HANDLE\_MSG 的意思就是告诉 Windows，当发生 WM\_COMMAND 消息的时候，请

//通知 Main\_OnCommand 函数

//id 这个参数是 Windows 给咱们“送”过来的

看 Main\_OnCommand 方法，初探 windows 的消息机制。

```
void Main_OnCommand(HWND hwnd, int id, HWND hwndCtl, UINT codeNotify)
```

```
{

switch(id)

{

case IDC_OK:

{

MessageBox (hwnd, TEXT ("欢迎访问如鹏网 www.RuPeng.com 大学生计算机学习
社区"), TEXT ("问好"), MB_OK);

}

break;

default:

break;

}

}
```

不同的控件要有不同的 id，控件的名字，在人类社会中名字有可能重名，但是在 windows 中 id 不能重复

按钮被按下的时候 Main\_OnCommand 方法被调用，hwnd 是对话框句柄（什么是句柄后面讲，通俗的说就是通过它能够操纵对话框），id 是控件的 id，后两个参数暂时不关心。

Main\_OnCommand 方法中根据 id，也就是被点击按钮的名字来决定不同的动作。

Hwnd 是当前窗口的句柄，句柄就是把柄！操作一个东西，要先抓住他的把柄！通过句柄就可以操作这个对象。

//不要忘了 break//不要忘了 case 中用大括号

3、定制自己的对话框，不同的按钮不同的问好

然后重新添加一个自己的按钮，在属性中修改 Name 属性为 IDC\_BTNHELLO。

控件名字的潜规则：所有控件的 Name 都以“IDC\_”开头，然后后边跟着控件类型的简称（按钮简称 BTN、文本框简称 EDT 等等），最后才是控件的真正的名字。修改按钮的 Caption 属性（也就是按钮上显示的文字）为“问好”。

编辑 Main\_OnCommand 方法：

```
void Main_OnCommand(HWND hwnd, int id, HWND hwndCtl, UINT codeNotify)

{

switch(id)

{
```

.....

```
case IDC_BTNHELLO:
```

```
{
```

```
    MessageBox(NULL, TEXT("世界你好"), TEXT("问好"), MB_OK);
```

```
}
```

```
break;//不要忘记 break
```

```
default:break;
```

```
}
```

```
}
```

运行程序，点击“问好”按钮，咱们第一个响应按钮动作的程序就做好了。

#### 4、自己动手写计算器 1.0 版

##### (1)、得到控件中用户输入的文本

```
GetDlgItemText(hwnd,IDC_EDTNAME,str,sizeof(str));
```

第一个参数是对话框的句柄；第二参数就是控件的 id (name)；第三个参数就是字符串数组的指针；

疑问:为什么不能像 `MessageBox` 一样把字符串做为返回值返回呢???

这就涉及到 C 语言中函数返回指针的问题了

在函数内部返回函数内构建的指针有可能出现数据混乱.

当 `f1` 函数在执行的时候, 这段内存是被占用的, 一旦函数返回, 那么内存就标记为可以被其他人、函数占用。

如果你把这段内存中的指针返回了, 那么一旦其他地方用了这段内存, 那么你引用的就是错误的数据了。

用 `GetWindowText`、`GetDlgItemText` 的时候为什么要传数组名呢? 复习: 初学者可以认为数组名就是指针, 函数只有得到指针, 才能直接操作数组中的数据。

(2) c 语言中字符串连接: `wsprintf(str, "...", a1, a2);`

还有 `sprintf`, 但是在编写 windows 程序的时候最好使用 `wsprintf`。

```
//GetDlgItemText(hwnd, IDC_EDITNAME, str, sizeof(str));
```

```
//第一参数是窗口的句柄, 第二个参数是要取得内容的控件的 ID
```

```
//第三个参数就是字符串指针, 第四个参数就是指针指向区域的大小
```

```
TCHAR name[256]; //为什么不用 strlen, sizeof。
```

//如鹏网站内搜索搜“strlen, sizeof”区别

//Windows 编程中一般都用 TCHAR 代替 char。TCHAR 有时候就是 char。

//解决 Unicode、中文编码问题。

GetDlgItemText(hwnd, IDC\_EDITNAME, name, sizeof(name));

//printf 是打印到屏幕上

//printf("%s 你好", name);

//sprintf 是打印到字符串中

//相当于很多 C#、Java 语言里的字符串相加 name+"你好"

//sprintf(字符串, "...", a1, a2, a3...);

TCHAR msg[256];

//sprintf(msg, "%s 你好", name);

//wsprintf, C 语言开发 Windows 程序的时候用 wsprintf 代替 sprintf



//解决中文等问题

```
wsprintf(msg, "%s 你好", name);
```

```
MessageBox(hwnd, msg, msg, MB_OK);
```

### (3) 向用户问好

```
TCHAR name[256];
```

```
GetDlgItemText(hwnd, IDC_EDTNAME, name, sizeof(name)/sizeof(TCHAR));
```

```
TCHAR msg[256];
```

```
wsprintf (msg, "%s 你好", name);
```

```
MessageBox(hwnd, msg, TEXT("问好"), MB_OK);
```

### (4)

C 语言里边字符串转换为数字: atoi: ascii to int

(5)

编写 windows 程序的时候最好用 TCHAR 来代替 char，可以避免中文的问题

```
case IDC_BTNHELLO:
```

```
TCHAR name[256];
```

```
GetDlgItemText(hwnd, IDC_EDTNAME, name, sizeof(name)/sizeof(TCHAR));
```

```
int i = atoi(name);
```

```
int j=i*2;
```

```
TCHAR result[256];
```

```
itoa(j, result, 10);
```

```
MessageBox(hwnd, result, TEXT("问好"), MB_OK);
```

(6) Get、Set: 配对的, 设置对话框的值用 SetDlgItemText:

代码:

```
case IDC_BTNADD:
```

```
//取第一个文本框的字符串, 然后得到整数表示
```

```
// 取第二个文本框的字符串, 然后得到整数表示
```

```
//计算两个整数的和
```

```
//把和重新转换成字符串, 然后 Messagebox 出来。
```

```
TCHAR cNumber1[256];
```

```
GetDlgItemText(hwnd, IDC_EDTNUM1, cNumber1, sizeof(cNumber1)/sizeof(TCHAR));
```

```
int n1 = atoi(cNumber1);
```

```
TCHAR cNumber2[256];
```

```
GetDlgItemText(hwnd, IDC_EDTNUM2, cNumber2, sizeof(cNumber2)/sizeof(TCHAR));
```

```
int n2 = atoi(cNumber2);
```

```
int n3 = n1+n2;
```

```
TCHAR cResult[256];
```

```
itoa(n3, cResult, 10);
```

```
SetDlgItemText(hwnd, IDC_EDTRESULT, cResult);
```

```
break;
```

```
//int i1;
```

```
//int i2;
```

//不能梦想着直接定义两个 int 类型的参数，就让 windows 告诉你两个数的值

```
TCHAR str1[256];
```

```
TCHAR str2[256];
```

```
GetDlgItemText(hwnd, IDC_EDIT1, str1, sizeof(str1));
```

```
GetDlgItemText(hwnd, IDC_EDIT2, str2, sizeof(str2));
```

//两个字符串不能直接相加，要先转换成 int 类型。

//C 标准库函数：atoi, a:ascii

to i:int 理解了 atoi 的简写不用死记硬背了

```
int i1=atoi(str1);
```

```
int i2=atoi(str2);
```

```
int i3=i1+i2;
```

```
//printf("%i", i3);
```

```
TCHAR str3[256];
```

```
itoa(i3, str3, 2); //字符串转换为整数，第三个参数就是进制。
```

```
//wsprintf(str3, "%i", i3);
```

```
SetDlgItemText(hwnd, IDC_EDIT3, str3);
```

(7) 代码中有 `UINT`、`TCHAR`、`LRESULT`、`HWND` 之类的新的数据类型，其实它们只是一些类型的别名而已，可以通过宏定义看出来。但是考虑到可移植性，尽量不要使用它们的真实类型。`Typedef` C 语言中声明类型别名。

(8) 作业：做一个面积计算器，用户输入半径，在另外一个文本框中显示面积。

(10) 作业：用户输入十进制整数，在另外一个文本框中显示整数的二进制形式。做一个十进制到二进制的转换小工具。

(10) 作业：计算两个数的差。

(11) 思考：这个计算两个数的和程序有什么缺陷？没有阻止用户输入非数字。怎么解决。

```
{
```

```
TCHAR str1[256];
```

```
TCHAR str2[256];
```

```
GetDlgItemText(hwnd, IDC_EDIT1, str1, sizeof(str1));

GetDlgItemText(hwnd, IDC_EDIT2, str2, sizeof(str2));

int i1=atoi(str1);

int i2=atoi(str2);

TCHAR str1temp[256];

TCHAR str2temp[256];

wsprintf(str1temp, "%i", i1);

wsprintf(str2temp, "%i", i2);

if(strcmp(str1, str1temp)!=0)

{

    MessageBox(hwnd, TEXT("第一个数不是合法的数!"),

        TEXT("错误提示"), MB_OK|MB_ICONERROR);

    return;

}

if(strcmp(str2, str2temp)!=0)

{

    MessageBox(hwnd, TEXT("第二个数不是合法的数!"),

        TEXT("错误提示"), MB_OK|MB_ICONERROR);
```

```
        return;

    }

    int i3=i1+i2;

    TCHAR str3[256];

    wsprintf(str3,"%i",i3);

    SetDlgItemText(hwnd, IDC_EDIT3, str3);

}
```

多练手，光听视频什么用都没有，浪费时间，必须练手，再简单的代码也要自己写，不要眼高手低!!! 你没有多牛 X。

#### **如何提取函数：**

```
/*BOOL Main_OnInitDialog(HWND hwnd, HWND hwndFocus, LPARAM lParam)

{

    return TRUE;

}*/

BOOL IsInt(TCHAR* str)

{

    int i=atoi(str);

    TCHAR strtemp[256];
```



```
        wsprintf(strtemp, "%i", i);

        if(strcmp(str, strtemp) != 0)

        {

            return FALSE;

        }

        else

        {

            return TRUE;

        }

    }

}

/*void Main_OnCommand(HWND hwnd, int id, HWND hwndCtl, UINT
codeNotify)

{

    switch(id)

    {

        case IDC_OK:*/

        {

            TCHAR str1[256];
```

```
TCHAR str2[256];

GetDlgItemText(hwnd, IDC_EDIT1, str1, sizeof(str1));

GetDlgItemText(hwnd, IDC_EDIT2, str2, sizeof(str2));

if(IsInt(str1)==FALSE)

{

    MessageBox(hwnd, TEXT("第一个数不是合法的数!"),

        TEXT("错误提示"), MB_OK|MB_ICONERROR);

    return;

}

if(IsInt(str2)==FALSE)

{

    MessageBox(hwnd, TEXT("第二个数不是合法的数!"),

        TEXT("错误提示"), MB_OK|MB_ICONERROR);

    return;

}

int i1=atoi(str1);

int i2=atoi(str2);

int i3=i1+i2;

TCHAR str3[256];
```

```
wsprintf(str3, "%i", i3);  
  
SetDlgItemText(hwnd, IDC_EDIT3, str3);  
  
}
```

#### 《C 语言也能干大事》第四节

板书:

上节课作业

(8) 作业: 做一个面积计算器, 用户输入半径, 在另外一个文本框中显示面积。

(10) 作业: 用户输入十进制整数, 在另外一个文本框中显示整数的二进制形式。做一个十进制到二进制的转换小工具。

(10) 作业: 计算两个数的差。不要总认为看很多书、学高深技术、看很多视频就是牛人, 能把一个最简单的程序快速写出来的才是牛人!!!

(11) 思考: 这个计算两个数的和程序有什么缺陷? 没有阻止用户输入非数字。怎么解决。

调试的技巧, 断点: 执行到断点的时候程序暂停住。

#### 《C 语言也能干大事》第五节

板书:

上节课的作业你自己写了吗? 比我快吗? 如果没有写的话别继续往下看, 先回去自己写。

实现“自己动手写计算器 2.0 版”, 允许选择运算符号。

ComboBox 的操作

(1) 添加项:

填入 Combo 最简单的方法是借助 `ComboBox_AddString` “函数”（其实是宏）：

```
ComboBox_AddString(hwndCtrl,str)
```

每个控件也有自己的句柄。句柄是 **WINDOWS** 用来标识被应用程序所建立或使用的对象的唯一整数，**WINDOWS** 使用各种各样的句柄标识诸如应用程序实例、窗口、图片、图标等等。控件句柄和控件 `id` 的区别：控件句柄在每次运行的时候都是不一样的，而控件 `id` 则是确定的。`ComboBox_AddString` 等几乎所有控件函数都可以对整个 **Windows** 中的控件进行控制（有的游戏外挂的原理），在一个对话框内 `id` 不能重复，但是在不同的对话框内则可以重复，因此需要一个全局的不重复的句柄来表示控件，通过对话框+`id` 联合确定。两个不同的程序（哪怕是同一个程序启动的两个不同的实例，开两个记事本）是找到我爸，然后找我父亲的 `id=1` 的儿子，也就是我，然后再打我。这就是通过我父亲的把柄和我的 `id=1` 来定义我

`GetDlgItem` 由对话框句柄+`id` 得到控件的句柄。

`Main_OnInitDialog` 对话框初始化的时候被调用。

做一个例子

(2) 确定目前选项的索引：`int ComboBox_GetCurSel(`

`HWND hwndCtl);Cur:CurrentSel:Selection` 返回值是当前项的编号，从 0 开始。得到的不是“北京市”这样的字符串，而是索引（编号）如果没有项目被选中，那么从函数中传回的 `iIndex` 值为 `CB_ERR`。做一个例子

(3) 删除项：

```
int ComboBox_DeleteString(  
    HWND hwndCtl,    int index);
```

做一个例子

(4) 取得有多少项：

```
int ComboBox_GetCount(  
    HWND hwndCtl);
```

返回值是个数。

(5) 选定某一项 `int ComboBox_SetCurSel(  
 HWND hwndCtl, int index)`; 将 `index` 设定为 -1 则取消所有选择

(6) 得到某项的值

```
int ComboBox_GetLBText(  
    HWND hwndCtl, int index, LPCTSTR lpszBuffer);
```

怎么让 Combo 不能改? 类型: 下拉列表

更多 ComboBox 的用法

[http://msdn.microsoft.com/en-us/library/cc656426\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/cc656426(VS.85).aspx)

是宏, 不是函数。SendDlgItemMessage/SendMessage。

额外知识, 不讲, 自学 (不想变成填鸭式教育, 课上只讲思想性的东西, 会给大家学习资料, 希望大家培养自学能力)

实现四则运算器:

作业

(1)

给本节课的小练习提改进建议并且自己实现。

(2) 学生管理系统 (画原型界面布置) 增、删、查 (双击)、搜

多练手，光听视频什么用都没有，浪费时间，必须练手，再简单的代码也要自己写，不要眼高手低！！！你没有多牛 x

### 《C 语言也能干大事》第六节

板书：

上节课作业：学生管理系统（画原型界面布置）增、删、查（双击）、搜

接下来我们要实现小时钟。涉及到的主要知识点是定时器的使用，涉及到的 C 知识就是函数指针（回调函数）

使用定时器

回调就是不是你调别人，而是别人调你。时间到了会叫你，没到的时候不会烦你。

让 Windows 直接将定时器消息发送给您程序的另一个函数。

我们把以下的 callback 函数称为 TimerProc（您能够选择与其它一些用语不会发生冲突的任何名称），它只处理 WM\_TIMER 消息：

```
VOID CALLBACK TimerProc (  
HWND hwnd, UINT message, UINT iTimerID, DWORD dwTime);  
  
SetTimer (hwnd, iTimerID, iMsecInterval, TimerProc) ;
```

19、得到当前日期：

```
void GetLocalTime (LPSYSTEMTIME);
```

课上练习：

```
SYSTEMTIME stLocal;
```

```
char chBuf[256];
```

```
GetLocalTime(&stLocal);
```

```
//显示时间的间隔。
```

```
wsprintf(chBuf, TEXT("%u/%u/%u %u:%u:%u %u 周%d\r\n"),
```

```
stLocal.wYear, stLocal.wMonth, stLocal.wDay,
```

```
stLocal.wHour, stLocal.wMinute, stLocal.wSecond,
```

```
stLocal.wMilliseconds, stLocal.wDayOfWeek);
```

```
MessageBox(NULL, chBuf, "", MB_OK);
```

```
SetTimer(hwnd, 1, 1000, TimerProc);
```

```
void CALLBACK TimerProc (HWND hwnd, UINT message, UINT iTimerID,  
DWORD dwTime)
```

```
{  
  
SYSTEMTIME time;  
  
GetLocalTime(&time);  
  
TCHAR strTime[256];  
  
//%后面的 0 表示不够的位数补零，5 是总位数，就这样  
  
wsprintf(strTime, "%04d-%02d-%02d  
%02d:%02d:%02d", time.wYear, time.wMonth, time.wDay, time.wHour, time.wMin  
ute, time.wSecond);  
  
SetDlgItemText(hwnd, IDC_EDTTIME, strTime);  
  
}
```

作业：定时关机、定时关闭显示器，如何编程实现关机、关显示器。到整天播放音乐。自学！！！“关闭显示器 API”、“关闭显示器函数”

### 9、取得复选按钮的是否选中：

```
LRESULT Button_GetCheck(  
    HWND hwndCtl);  
BST_CHECKED BST_UNCHECKED
```



设置复选按钮是否选中:

```
LRESULT Button_SetCheck(  
    HWND hwndCtl,    int check);BST_CHECKED BST_UNCHECKED
```

## 7、ListBox(和 Combo 类似)

### (1) 添加项:

如果采用 **LBS\_SORT** 样式, 那么填入清单方块最简单的方法是借助 **ListBox\_AddString**:

```
int ListBox_AddString(  
    HWND hwndCtl,    LPCSTR lpsz);
```

如果没有采用 **LBS\_SORT**, 那么可以使用 **ListBox\_InsertString** 指定一个索引值, 将字符串插入到清单方块中:

```
int ListBox_InsertString(  
    HWND hwndCtl,    int index,    LPCSTR lpsz);
```

### (2) 删除项:

```
int ListBox_DeleteString(  
    HWND hwndCtl,    int index);
```

### (3) 取得有多少项:

```
int ListBox_GetCount(  
    HWND hwndCtl);
```

(4) 选定某一项 **int ListBox\_SetCurSel**(  
 HWND hwndCtl, int index); 将 **iParam** 设定为 -1 则取消所有选择 (5) 确定目前选项的索引: **int ListBox\_GetCurSel**(  
 HWND hwndCtl); 如果没有项目被选中, 那么从呼叫中传回的 **iIndex** 值为 **LB\_ERR**。(6) 将某项目复制到文字缓冲区中: **int ListBox\_GetText**(  
 HWND hwndCtl, int index, LPCTSTR lpszBuffer);

## 9、菜单资源

(1) 制作好菜单资源后，在对话框的 **Menu** 属性中选择刚才制作好的对话框。

(2) 响应菜单操作。菜单点击后也会发送 **WM\_COMMAND** 消息 **id** 就是被选择菜单的 **id**。例子：

```
case IDM_QUIT:
```

```
EndDialog(hwnd,id);
```

作业：文件操作。自己动手写记事本。路径写固定的路径。

下节课内容：作业点评，文件对话框、播放音乐

## 《C 语言也能干大事》第七节

板书：

作业点评：文件操作。自己动手写记事本。

RichEdit 控件。LoadLibrary( "riched32.dll" );

参考资料：[http://msdn.microsoft.com/en-us/library/bb787605\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/bb787605(VS.85).aspx)

## 5、给结构体所有字段置零

复习一下 C 语言中 **memset** 以及结构体的应用。C 语言中声明一个结构体变量之后，在使用这个变量之前要首先用 **memset** 来把各个位清零。使用 C 语言内置的函数 **memset** 来完成：

```
memset(&s1, sizeof(s1), 0);
```

在这里我们使用 ZeroMemory “函数”，ZeroMemory 其实并不是函数，其实是 memset 的宏定义

ZeroMemory 其实就是 memset，纸老虎而已，不是一个全新的函数。

## 6、打开文件对话框的使用

首先#include <commdlg.h>

```
OPENFILENAME ofn;
```

```
char szFile[MAX_PATH];
```

```
ZeroMemory(&ofn, sizeof(ofn));
```

```
ofn.lStructSize = sizeof(ofn);
```

```
ofn.lpstrFile = szFile;
```

```
ofn.lpstrFile[0] = TEXT('\0');
```

```
ofn.nMaxFile = sizeof(szFile);
```

```
ofn.lpstrFilter = TEXT("ALL\0*.*\0Text\0*.TXT\0");
```

```
ofn.nFilterIndex = 1;
```

```
ofn.lpstrFileName = NULL;

ofn.nMaxFileName = 0;

ofn.lpstrInitialDir = NULL;

ofn.hwndOwner = hwnd;//自学能力、探索能力、猜测能力

ofn.Flags = OFN_EXPLORER | OFN_PATHMUSTEXIST | OFN_FILEMUSTEXIST;

if (GetOpenFileName(&ofn))

{

    MessageBox(NULL, szFile, "", MB_OK);

}
```

参数太多只能搞 **Struct**。下面将对这段代码作解释：

(1)、ZeroMemory 的含义见上边

(2) char szFile[MAX\_PATH];//定义一个长度为 MAX\_PATH 的 char 数组。

MAX\_PATH 是系统定义的一个宏，260。Windows 操作系统文件路径的最大长度。

(3) `ofn.lpstrFile = szFile;`

//szFile 是一个字符数组，那么 szFile 这个名字就代表这个字符数组的首地址

(4) 文件类型过滤器

`*.*`就表示所有文件

`*.txt` 就表示所有文本文件

`abc*.*`就表示所有以 abc 开头的文件

`abc*.txt` 就表示所有以 abc 开头的文本文件 (txt)

`ofn.lpstrFilter = TEXT("ALL\0*.*\0Text\0*.TXT\0");`用 “\0” 分割出几个段，每两段是一组过滤器，每组的第一个段代表过滤器的显示的值，而第二段表示真正的过滤器

(5) `ofn.nFilterIndex = 2;`默认选择第几个过滤器，是 1 开始的，不是 0。这也是试验出来的。

(6) windows 的一个惯例：结构体的指针类型别名一般是 LP+结构体的名字

`GetOpenFileName(&ofn)`而不是 `GetOpenFileName(ofn)`，因为 `GetOpenFileName` 函数要的是 `OPENFILENAME` 的指针

(7) 用户关闭对话框，`GetOpenFileName` 函数返回，返回值是 `BOOL`，等于 `TRUE` 表示用户选择了文件，如果等于 `FALSE` 就表示用户选择了【取消】按钮

### 第三节

#### 1、GetOpenFileName

希望同学们不要只记我教的结果，而是要跟着我的思路进行思考，看解决问题的方式。否则就又变成“我教同学们学”的填鸭式教育了。

(2) 课上练习：打开对话框，用户选择一个音乐文件，然后用 `MCIsendString` 播放，注意 `ShortPath` 的问题。如何让音乐文件跟着文件走，`GetModuleFileName`、`GetCurrentDirectory`

(3) 打开多个文件：

增加 `OFN_ALLOWMULTISELECT` 选项后测试一下。发现如果选择一个文件后显示的还是这个文件的路径，可是如果选择多个文件显示的就只有目录的路径，怎么回事？

调试一下，看看 `szFile` 在内存中的样子。调试的是发现内存中的样子是目录的路径然后加上各个文件名，中间用 “\0” 分割，现在明白为啥值显示目录的路径了吗？

大家明白这一点就可以了，有兴趣的可以课后来做多文件选择的处理。做播放器的时候经常需要这个功能，也就是播放列表。有一篇文章可以参考一下：  
<http://xxkkff.blog.51cto.com/162016/26222>

## 2、GetSaveFileName

保存文件对话框

只要把上边函数换成 `GetSaveFileName`。`OFN` 本来是 `OpenFileName` 的简写，它 `SaveFileName` 也跑来凑热闹，吼吼。如果需要保存文件覆盖提示怎么办呢？看看 `OFN` 有选项吗？`OFN_OVERWRITEPROMPT`

`MciSendString` 播放音乐，配合 `GetOpenFileName` 实现播放器

`GetShortPathName`

```
mciSendString( "play ***" , "", 0, NULL);
```

Mcisendstring 更多用法（暂停、继续、录音、播放进度控制）参考

[http://baike.baidu.com/view/1622810.htm?fr=ala0\\_1](http://baike.baidu.com/view/1622810.htm?fr=ala0_1)

[http://msdn.microsoft.com/en-us/library/dd743572\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/dd743572(VS.85).aspx)

作业：提供打开、保存功能的记事本

作业：自己动手写播放器

<http://so.rupeng.com/?keyword=C%E8%AF%AD%E8%A8%80+%E6%92%AD%E6%94%BE%E5%99%A8>

更多控件用法

[http://msdn.microsoft.com/en-us/library/bb773169\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/bb773169(VS.85).aspx)

VC 就是全部、VC 不是唯一：C-Free、Pelles C、Dev C++、CodeBlocks。C-Free 使用演示（Vista、Win7 上资源编辑器可能有问题）。

学完第七节就可以开始从第一版 11 节开始继续学习了、第 12 节也是旧版的，不用看。第 1 版 1 至 10 节中还有一些在第二版中没有讲的东西，有兴趣的可以去粗略的过一遍，当然不学第一版也能继续向后学。

## 《C 语言也能干大事》第十一节：Socket 网络编程

板书：

这节课是自己动手写 QQ、自己动手写飞鸽传书的基础，以后想从事网络开发、信息系统开发、游戏开发、嵌入式系统开发等的同学必须学习。

演示一个简单的和网易 Email 服务器对话的过程。

### 1、网络基本概念

讲解 QQ 的基本原理

讲解浏览器上网的基本原理

服务器：Server。提供服务的计算机。

客户端：Client。和服务器端交互的一个设备。

IP 地址：计算机在网络中的地址。相当于每个家庭的电话。211.97.1.2

别名、域名：[www.RuPeng.com](http://www.RuPeng.com)。域名就相当于你家的地址。问 114，你家的电话（告诉地址）

DNS: 网络中的 114。Domain Name System。

端口: Port。不同的程序连接服务器时候用的插座。(程序和服务器通信的时候的一个开的一个口子)

Socket (套接字): 导线、网络连接。

2、

WinSock、WinSocket。

Socket:平台无关的。Socket 标准。所有操作系统都实现了这个标准。Send、recv。。标准的。一通百通!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

WinSock: 添加了一些 Windows 独有的函数。而且不用这些函数的话 Socket 程序还跑不起来。羞臊微软!!!

WinSock 编程:平台无关, 一通百通。WSA(Windows Socket API)

Sz1.tencent.com

Sz2.tencent.com

3、

LastError

调用 WindowsAPI 的时候, 每步执行完毕都会把执行结果放到 LastError 中去, 所以应该在函数执行完毕立即去取。

DeleteFile(f1)

DeleteFile(f2)

GetLastError()

LastError 只能容纳一个错误码

DeleteFile(f1)

GetLastError()

DeleteFile(f2)

GetLastError()

错误码。2233、344555

FormatMessage: 把错误码转换为错误消息。

Windows 错误处理

GetLastError

```
1. void ShowError()
2. {
```



```
3.  TCHAR* lpMsgBuf;
4.  FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER| //自动分配
   消息缓冲区
5.  FORMAT_MESSAGE_FROM_SYSTEM, //从系统获取信息
6.  NULL, GetLastError(), //获取错误信息标识
7.  MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT), //使用系统缺
   省语言
8.  (LPTSTR)&lpMsgBuf, //消息缓冲区
9.  0,
10. NULL);
11. MessageBox(NULL, lpMsgBuf, "", MB_ICONERROR);
12. }
```

#### 复制代码

在.h 中声明函数，在.c 中实现!!!

都会有哪些错误？

#### 4、模板代码：

(1)、添加 winsock2.h

(2)、连接库 “wsck32”。怎么知道添加它呢？或添加预处理文件#pragma  
comment(lib, "ws2\_32.lib")。

注意：在 mainDlg.h 里面添加 void SocketTest();

void ShowError();

#### (3)、模板代码

```
1.  WSADATA wsaData;
2.  //初始化 Socket 库
3.  WSAStartup(MAKEWORD(2,0), &wsaData);
4.  //创建一根电线
5.  SOCKET sock = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
6.  SOCKADDR_IN sa;
7.  sa.sin_family = AF_INET;
8.  //设置电线连接服务器端的端口
9.  sa.sin_port = htons(IPPORT_SMTP);
```

```
10. //123.125.50.135 是 ping smtp.163.com 出来的,后面会讲使用 gethostbyname
    来直接从
11. //主机名得到 ip 地址
12. sa.sin_addr.S_un.S_addr = inet_addr("123.125.50.135");
13. //为什么不用设置客户端的端口,难道不需要客户端的端口吗?
14. if(connect(sock,(SOCKADDR *)&sa,sizeof(sa))==SOCKET_ERROR)
15. {
16. ShowError();
17. return;
18. }
19. //做事情。把大象放冰箱总共分几步
20. closesocket(sock);
21. WSACleanup();
```

#### 复制代码

QQ 服务器的端口 8888。需要指定客户端的端口是 9999。

端口有一个特点：排他性！端口已经被别的程序占用，不能再用这个端口。

开两个 QQ。1111111111

2222222222 9999。如果端口可以被多个程序使用的话，QQ 聊天服务器发过来的消息被多个 QQ 收到。很荒谬！排他性！

如果在程序里写死了客户端用的端口。。。所以才需要 Socket 替客户端动态分配一个端口。所以不需要在程序中显示指定客户端的端口号！

客户端向服务器端发消息：send

服务器端想客户端发消息，客户端这边应该接收，recv。（receive）

SMTP 服务器，你连上来以后服务器端主动向你发一条消息

```
recv(sock,buffer,256,0);
```

第一个是使用的 socket，“导线”，套接字

接收数据的缓冲区的指针

接收数据的缓冲区的大小

0

```
send(sock,cQuit,strlen(cQuit),0);
```

第一个是使用的 socket, “导线”, 套接字

接收数据的缓冲区的指针

接收数据的缓冲区的大小

0

```
1. (4)一上来人家就主动和你唠嗑:
2. char buffer[256];
3. ZeroMemory(buffer,sizeof(buffer)/sizeof(char));
4. //接收问候语
5. recv(sock,buffer,256,0);
6. MessageBox(hwnd,buffer,"",0);
7. (5) 朋友再见
8. //注意不能忘了末尾的回车
9. TCHAR cQuit[] = "QUIT\n";
10. send(sock,cQuit,lstrlen(cQuit),0);
11.
12. ZeroMemory(buffer,sizeof(buffer)/sizeof(char));
13. //接收 GoodBye
14. recv(sock,buffer,256,0);
15. MessageBox(hwnd,buffer,"",0);
```

[复制代码](#)

## 《C 语言也能干大事》第十二节：从 CFree 到 VC

一、为什么现在要转换到 VC6 了？

(1)、亲身体会一通百通的威力。

(2)、C-Free 做入门级的学习工具比较合适，所以在《C 语言也能干大事》一开始我们选择它做入门工具。但是 C-Free 在工程化的大程序方面能力欠佳，而《C 语言也能干大事》后面将是大量的实战项目，比如自己动手写 QQ、自己动手写银行系统、自己动手写 VC 等，所以要转用 VC。

(3) 用 c-Free 的目的是让同学们知道不是 C/C++ 就是 VC，VC=编辑器+编译器；C-Free=编译器+可替换的编译器。C-Free 与 VC。

(4) 用 VC 开发出来的工程比较好和同学们交流，比较好做为课程设计、毕业设计的作业直接发给老师。

别害怕。只用 10 分钟就能让你从 C-Free 过渡到 VC，这就是一通百通的魅力。跟我来吧。

为什么不同 VC 更高的版本？

VS 更高版本和 VC6 比起来几乎没什么差别，特别是对于咱们《C 语言也能干大事》这门课约等于一模一样，而且高版本体积要大很多，要占 2G 多，装了很多用不着的东西，而 VC6 只有 200 多 MB。

## 二、VC 开发环境的搭建

### 1、VC 的下载

如果已经安装好 VC 的就不要再重新安装了。

如果没有安装的同学可以下载如鹏网专门为大家定制的“如鹏网版 VC6”，去掉了一些已经作废的组件，丝毫不影响使用，尺寸只有 103MB。

下载地址：[http://down1.rupeng.com/download/software/RuPengvc6\\_chs.rar](http://down1.rupeng.com/download/software/RuPengvc6_chs.rar)

注意如果使用 Vista 的同学请关闭 Vista 的 UAC 这个废物。

### 2、

MessageBox 归来

Win32Application→简单的 Win32 程序。

## 三、对话框图形化环境搭建

### 1、从 Win32 Application 向导生成对话框程序

自己配置太麻烦了，每次建一个程序就需要建一遍对话框、MainProc、添加 LoadLibrary 等等。

### 2、配置如鹏网为同学们定制的 Win32 DialogBased Application 向导

已经下载“如鹏网版 VC6”的同学从安装包的 Tools 目录下就可以找到

Win32DlgBasedAppWizard.zip

没有下载“如鹏网版 VC6”的同学从下面的地址单独下载：

<http://down1.rupeng.com/download/software/Win32DlgBasedAppWizard.zip>

只有 200 多 K 大小。

配置方法：

将 DlgBaseAppWizard.pdb 和 DlgBaseAppWizard.awx 拷贝到 VC6 的安装目录的 Common\MSDev98\Template 目录下即可，一般 VC6 都安装到了 C 盘，所以一般只要拷贝到 C:\Program Files\Microsoft Visual Studio\Common\MSDev98\Template 即可。然后重启 VC6，在 VC6 的向导页面中就可以看到“Win32 DialogBased Application”的向导选项了。

Win32 DialogBased Application 并没有做什么神奇的工作，只不过是帮大家快速搭建一个对话框工程而已，没有什么自己的特性，大家手动也能完成，这个工程拿到没有“Win32 DialogBased Application”的 VC 中也能运行和开发。

注：“Win32 DialogBased Application”就是在“自己动手写开发工具”，希望大家早日从用工具的人变成造工具的人。人类进化的一大标志就是学会制造工具！哈哈！

### 3、

Win32 DialogBased Application 帮大家做了什么

添加了一个对话框、添加了模板代码、添加了对 IP 地址控件、日历控件等高级高级控件的显示支持代码。

4、争取大家自己从头不用 Win32 DialogBased Application 建一个对话框工程。

## 五、配置 VC6，使之更好用

1、修改代码自动完成的快捷键，因为默认的 Ctrl+Space 和输入法切换冲突  
工具栏上点击右键，选择“自定义”（customize）；在对话框中切换到“键盘”（keyboard），“类别”选择“编辑”，在命令中找到“CompleteWord”（列表中快速查找的技巧，在几乎任何软件中都通用，在列表中快速敲入单词），然后焦点设置到“按下新快捷键”上，按快捷键“Alt+/"（也可以是其他的自己习惯的，只要是不和其他的冲突就可以），主要是按快捷键，不是敲入字符“Alt+/"，学了后面的课程你就会知道这是一个 HotKey 控件，而不是 TextBox。

然后注意不要忘了点击【分配】。

要学会使用快捷键，我看很多同学写程序太费劲了，经常用鼠标点来点去，慢的像一头牛，CtrlC、CtrlV 什么的起码不应该总是用鼠标，牛逼的程序员很少用鼠标。

### 2、

关闭工程的“设置”→“浏览信息”→“创建浏览信息文件”（Browsing Info）

Project→Setting→BrowseInfo→build BrowseInfo File

因为它有可能造成 VC 死锁

### 3、

如果 VC 还是有偶尔死锁，那么请杀掉 MSDev 进程，然后重启 VC。

移植几个程序：计算器；音乐图片贺卡；设置程序图标；

## 《C 语言也能干大事》第十三节

### Socket 网络开发 2

这节课是对上节课内容的复习和实战，并且融合了 C 语言中指针、函数、类型转换、结构体等知识。

#### 一、复习：

复述 QQ 的基本原理；在一个典型的网络程序中分为哪两端？

IP 地址、域名、DNS、端口是做什么用的？

演示一个简单的和网易 Email 服务器对话的过程。

Socket 是 Windows 特有的吗？什么是 WinSock？

WinSock 和 Socket 标准的区别和联系。

Windows 中函数的错误信息是通过什么来报告的？

创建客户端到服务器端的 Socket 连接的时候需要指定服务器端的端口吗？为什么？需要指定客户端的端口吗？为什么？客户端不用显式指定端口，但并不是说不需要端口，而是这个端口是由操作系统分配的。

有一个端口能同时被多个程序占用吗？

发送、接收数据各用什么函数？

## 二、开发网络定时器

### 1、看看 Windows 的网络授时功能（客户端）

### 2、网络授时原理

客户端连接到授时服务器上，授时服务器会马上将当前时间发送给客户端，原理很简单。

网络授时分为好几种协议，比如返回的是日期字符串，有的返回的是自 1900 年 1 月 1 日至今的秒数。“2009-03-08 22:13:58”、39393933993

扩展阅读：<http://zhidao.baidu.com/question/20121991.html>

### 2、

国家授时中心提供的网络授时返回的是自 1900 年 1 月 1 日至今的秒数。

### 3、

先移植第一节代码

可供选择的授时服务器（不止这些）：

www.time.ac.cn

国家授时中心

Clock.sgi.com

Sgi

Tick.mit.edu

麻省理工学院

Time.nist.gov

NIST

Compile: 编译

Link: 连接

编译是单一的文件编译成.o、.obj 文件，然后 Link 是.obj 连接成.exe

Link 库！！！！

一通百通的力量！！！！！！！！！！！！！！！！！！！！

C-Free、VC 没有任何差别！！！！！！！！！！！！！！！！！！！！

小技巧：在命令提示符下不能 CtrlC、CtrlV，但是选中后点右键就是复制、粘贴

4、

授时服务器是 37 端口 IPPROTO\_TIMESERVER。返回的是什么？

```
unsigned long ulTime;
```

```
recv(sock,(char *)&ulTime,sizeof(unsigned long),0);
```

深刻理解这段代码的含义

```
ulTime = ntohl(ulTime);
```

从**网络字节顺序**转换为主机字节顺序。这个是重点，很多公司面试的时候都会问这个，以后工作中也会遇到。

什么是网络字节顺序、什么是主机字节顺序。

不同的 CPU 中处理整数的方式不一样，有的是低位在前、有的是高位在前。计算机组成原理、数字电路。。。

网络字节顺序则是统一标准。

如果直接接收的话就会造成接受者接收的数据正好是相反的。

扩展阅读：

[http://hi.baidu.com/cdmember\\_daihw/blog/item/ad83090f1fb4ffebab6457e6.html](http://hi.baidu.com/cdmember_daihw/blog/item/ad83090f1fb4ffebab6457e6.html)

中国是东 8 区。猜的：时间服务器返回的是格林尼治时间。

Windows 中时间用 SYSTEMTIME 结构体表示时间，怎么将 ulTime 转换为 SYSTEMTIME？网上找到下面的代码：

```
#define HIGHTIME
21968699 // 21968708 // Jan 1, 1900 FILETIME.highTime

#define LOWTIME
4259332096 // 1604626432 // Jan 1, 1900 FILETIME.lowtime

SYSTEMTIME st;

UINT64 uiCurTime, uiBaseTime, uiResult;

uiBaseTime = ((UINT64) HIGHTIME << 32) + LOWTIME;

uiCurTime = (UINT64)dwTime * (UINT64)100000000;

uiResult = uiBaseTime + uiCurTime;

FileTimeToSystemTime((LPFILETIME)&uiResult, &st);
```

把它封装成函数，怎么设计接口？深刻理解指针以及 Windows 中 LP 的惯例。

把时间打印出来，怎么打印？？？？？

怎么时间不对

设置系统时间

```
//SetSystemTime(&st);
```

作业：定时的进行网络对视。SetTimer 函数，1 分钟对时一次。时钟。

Over !

## 《C 语言也能干大事》第十四节：数据库开发入门

板书：

网络编程由于比较难，留给同学们一段时间去学习，数据库开发讲若干节以后再回头来讲自己动手写邮件发送器（网络编程的应用）和网络编程的服务器端开发。



这个知识点要达到的目的：能够使用常用的 SQL 语句，能够使用 ODBC 操作数据库，熟悉结果集、SQL 注入漏洞、范式等一通百通的东西，为以后有必要的情况下用 ADO、ADO.Net、JDBC 学一通百通的东西。

课程参考教材：杨中科的《程序员的 SQL 金典》

这门课和《数据库原理》的关系，这门课是对《数据库原理》的应用，通过这门课将能够学会数据库原理的实际应用，不再云里雾里。即使没学过《数据库原理》也没关系，这门课并不假定大家学过《数据库原理》，相当于预习效果。

企业里数据库开发是很大的一部分，不会什么也要会数据库开发。

开发环境的搭建：

（1）如鹏版绿色 MYSQL 的下载：

<http://down1.rupeng.com/download/software/RuPengGreenMYSQL.rar>

（2）ODBC 连接 MYSQL 的驱动 MYODBC 下载：

<http://down1.rupeng.com/download/software/MYODBC.rar>

（3）管理工具 SQLyog 下载：

<http://down1.rupeng.com/download/software/SQLyog.rar>

依次安装三个软件。如鹏版绿色 MYSQL 的使用，使用 SQLyog 连接数据库。

注意：如果初始化 MYSQL 以后移动了 MYSQL 的文件夹，那么需要重新运行“运行前先初始化.exe”。

1、执行“启动 MySQL.bat”

2、运行 SQLyog。如鹏版 MYSQL，用户名密码都是 root。

什么是数据库，数据库能做什么？

可以记录到文件中，比如：

姓名	年龄	工号	职位
----	----	----	----

Kider
-------

20
----

001

超版

CALF

21

002

版主

劣势:

1、必须对文件操作非常熟悉

2、比如对算法非常熟悉；一旦算法不好的话很容易造成性能问题。

3、并发操作。

广义的来讲，能够存储数据的地方都可以叫数据库。DBMS(DataBase Management System，数据库管理系统)，由 DBMS 来负责管理数据，使用者只要“描述”要进行什么操作就可以了，DBMS 相当于仓库管理员，只要向仓库管理员发出命令就可以，他怎么做我们不用关心，只要告诉他 What To Do（要做什么），不关心 How To Do（怎么做）。

DBMS 和数据库的关系。大家都很懒，所以很多时候说数据库其实指的是 DBMS。行业内的潜规则。

不同的仓库管理员有自己的不同的优势。不同品牌的 DBMS 也有自己的不同的特点。

MYSQL、MSSQLServer、DB2、Oracle、Access。。。。

OS/2

Sybase SQLServer、Microsoft SQLServer

Sybase 和 MS 的恩恩怨怨。

主流数据库（DBMS）有哪些？

SQL<>SQLServer<>MSSQLServer。最常见的错误。行业内的潜规则。

和仓库管理员打交道的方式：1、跑过去和他说；2、打电话；3、通过电脑传递信息（自动化的方式）；

操作数据库的两种方式：管理工具和程序代码。

什么是 Catalog（分类）（数据库、表空间）

表（Table）

把仓库分成不同的区域。不同的区域放不同类型的物品。

将一个数据库（Catalog）分成不同的表，每个表放不同的数据。公司机密信息数据库：发票信息；合同信息；公司帐号；。。。。建不同的表。

列（Column）

已经将生肉放到单独的生肉区域里，每块肉都有不同的特性，取肉部位、重量。。。为每块肉都贴一个标签，

规定每一列的格式，重量应该是数字。生产日期应该是日期类型。数据类型（DataType），Int、Char、double 一个道理。

数据库，表、列、数据类型、记录。

SQLYog 中左边是数据库。

Int: 整数; varchar: 字符串 (String), Var (可变的), Char (字符);

Boolean、char、date

怎么使用 Varchar, 要指定长度, 就像申明 `char buff = new char[200]`

使用工具创建数据库表, 插入一条记录。表的命名规则、字段命名规则。常见数据类型说明。

下节课内容: SQL 语句、数据的增删改查。

课下参考《程序员的 SQL 金典》进行深一步的学习和预习。

## **《C 语言也能干大事》第十五节: 数据库开发 2**

板书:

本课程的参考教材: 杨中科的《程序员的 SQL 金典》第三章

什么是 SQL 语句。

SQL 语句是和 DBMS “交谈”专用的语句，和 C、Java 等的代码不一样，SQL 语句需要你告诉 DBMS What To Do 就可以，不需要告诉它 How To Do。

本课程的命名规范：所有表以 T\_ 开头，字段（列、Column、Field）以 F 开头，这样可以很容易的看清哪个是表、哪个是字段。而且不容易和关键字冲突。

增删改查：Insert、Delete、Update、Select。

SQL 语句是大小写不敏感的

1、简单的 Insert 语句。向数据库中插入数据。Into 后的列名和 values 一一对应，字符串用单引号

```
Insert Into T_Person(FName,FAge) values('Jim',25)
```

```
Insert into T_Person(FName) values('lily')
```

2、怎么样查看表中数据

```
Select * from T_Person
```

Select \* from 表名

### 3、简化的 Insert 语句格式，不推荐

Insert into T\_Person values('poly',33)

### 4、数据的更新 Update

Update T\_Person Set FAge=30

### 5、多字段更新

### 6、带 Where 语句的 Update

update T\_Person Set FAge=50 where FName='Tom' OR FName='Jim'

### 7、数据的删除

Delete From T\_Person



是删除 T\_Person 表中的数据，并不是删除 T\_Person 整个表结构。表还在，数据没了，人去楼空。

## 8、带 Where 语句的删除

```
delete from T_Person where FName='Jim'
```

```
delete from T_Person where FAge>30
```

下节课内容：数据的检索。

课下参考《程序员的 SQL 金典》进行深一步的学习和预习。

## **《C 语言也能干大事》第十六节：数据库开发 3**

本课程的参考教材：杨中科的《程序员的 SQL 金典》第四章

### 1、SELECT 基本用法

a)

检索所有列、检索指定列

b)

按条件过滤。通配符、Between and

```
select * from T_Employee where FSalary>4000 and FSalary<8000
```

```
select * from T_Employee where FSalary BETWEEN 4000 And 8000
```

c)

数据汇总

```
select SUM(FSalary),AVG(FSalary),MAX(FSalary),MIN(FSalary) from T_Employee
```

```
select SUM(FSalary),AVG(FSalary),MAX(FSalary),MIN(FSalary) from T_Employee  
where FSalary<8000
```

```
select COUNT(*) from T_Employee where FSalary>5000。满足条件的数据个数
```

d)

排序

Ascending: ASC, 升序

Descending: Des, 降序

```
select * from T_Employee order by FSalary desc
```

## 2、C 语言中访问数据库

a)

ODBC 简介。ODBC 是微软提供的访问数据库的一种标准接口, 通过 ODBC 可以连接 MSSQLServer、MYSQL、DB2、Oracle、Access 等各种数据库, 通过统一的函数进行访问, 也就是访问各种数据库都可以使用统一的函数。屏蔽了连接不同数据库的差异性。

b)

除了 ODBC 之外还有 ADO、ADO.net 等, Java 中有 JDBC 等。都有连接、结果集、游标、事务、参数化 SQL 等概念, 一通百通。

c)

没安装 MYODBC(MYSQL 的 ODBC 驱动)的首先安装 MYODBC。

<http://down1.rupeng.com/download/software/MYODBC.rar>

d)

sql.h、sql.h、sqltypes.h

e)

添加连接库“odbc32.lib odbccp32.lib”。如果使用的是 rupeng 的 DialogBased 向导则已经自动添加。如果使用其他向导或者 C-Free、PellesC 等开发工具，则需要手动添加

f)

看模板代码。不用关心每个函数，用的时候 copy 以后改一改就可以，理解只要流程、主要概念即可，不要深究。有兴趣的可以研究 CHECKDBSTMTError 宏的实现。

g)

访问数据库可能遇到的错误：连接错误、执行错误。

h)

连接数据库、执行 SQL、断开连接。连接字符串。数据库错误处理。

SQLHDBC hdbc：代表一个数据库连接句柄。和 Socket 里边的 Socket 连接类似，要访问数据库，先要连接到数据库。SQLHDBC：SQL、H 句柄、DB（DataBase）、C（Connection）

SQLHSTMT hstmt：代表一个 SQL 语句。STMT（Statement，语句）

SQLRETURN：执行结果。

ODBC 中字符串用 SQLCHAR，SQLCHAR 其实就是 char 的别名。

SQLCHAR ConnStrIn[MAXBUFLEN]和 char ConnStrIn[MAXBUFLEN]一样。

ConnStrIn: 连接字符串, 你要连接到的数据库的驱动、ip 地址、用户名密码、数据库名 (Catalog) 等等都在连接字符串里描述。DRIVER: 使用的驱动名; SERVER: ip 地址; UID: 用户名 (UserID); PWD (Password): 密码。DataBase: 数据库名, Catalog。

SQLDriverConnect: 创建到数据库的连接, 使用 ConnStrIn 连接。result 表示执行结果。如果失败了 `SQL_ERROR==result`。ShowDBConnError(hwnd,hdbc);来显示“连接错误”。

`result = SQLPrepare(hstmt,(SQLCHAR*)"insert into T_Person(FAge,FName) values(20,'kider')",SQL_NTS)`。创建 SQL 语句的句柄。

`result = SQLExecute(hstmt);` 执行 SQL 语句。

`SQLFreeStmt(hstmt,SQL_CLOSE);` 释放 SQL 语句

`SQLDisconnect(hdbc);` 断开数据库连接。

课下参考《程序员的 SQL 金典》进行深一步的学习和预习。

模板代码：

购买此贴后可以得到整个工程的源代码：

```
1. #include "stdafx.h"
2. #include <windows.h>
3. #include <windowsx.h>
4. #include <sql.h>
5. #include <sqlext.h>
6. #include <sqltypes.h>
7. #include "resource.h"
8. #include "MainDlg.h"
9. #define LOGIN_TIMEOUT 30
10. #define MAXBUFLen 255
11. #define CHECKDBSTMTError(hwnd,result,hstmt)
    if(SQL_ERROR==result){ShowDBStmterror(hwnd,hstmt);return;}
12.
13. BOOL WINAPI Main_Proc(HWND hWnd, UINT uMsg, WPARAM wParam,
    LPARAM lParam)
14. {
15.     switch(uMsg)
16.     {
17.         HANDLE_MSG(hWnd, WM_INITDIALOG, Main_OnInitDialog);
18.         HANDLE_MSG(hWnd, WM_COMMAND, Main_OnCommand);
19.         HANDLE_MSG(hWnd, WM_CLOSE, Main_OnClose);
20.     }
21.     return FALSE;
22. }
23. BOOL Main_OnInitDialog(HWND hwnd, HWND hwndFocus, LPARAM
    lParam)
24. {
```

```
25.     return TRUE;
26. }
27. void Main_OnCommand(HWND hwnd, int id, HWND hwndCtl, UINT
    codeNotify)
28. {
29.     switch(id)
30.     {
31.         case IDC_OK:
32.             DBTest(hwnd);
33.             break;
34.             default:
35. break;
36.     }
37. }
38. void Main_OnClose(HWND hwnd)
39. {
40.     EndDialog(hwnd, 0);
41. }
42. void ShowDBError(HWND hwnd,SQLSMALLINT type,SQLHANDLE
    sqlHandle)
43. {
44.     char pStatus[10], pMsg[101];
45.     SQLSMALLINT SQLmsglen;
46.     char error[200] = {0};
47.     SQLINTEGER SQLerr;
48.     long erg2 = SQLGetDiagRec(type, sqlHandle,1,
49.                               (SQLCHAR *)pStatus,&SQLerr,(SQLCHAR
    *)pMsg,100,&SQLmsglen);
50.     wsprintf(error,"%s (%d)\n",pMsg,(int)SQLerr);
51.     MessageBox(hwnd,error,TEXT("数据库执行错误
    "),MB_ICONERROR|MB_OK);
52. }
```

```
53. void ShowDBConnError(HWND hwnd,SQLHDBC hdbc)
54. {
55. ShowDBError(hwnd,SQL_HANDLE_DBC,hdbc);
56. }
57. void ShowDBStmtError(HWND hwnd,SQLHSTMT hstmt)
58. {
59. ShowDBError(hwnd,SQL_HANDLE_STMT,hstmt);
60. }
61.
62. void DBTest(HWND hwnd)
63. {
64.     SQLHENV henv = NULL;
65.     SQLHDBC hdbc = NULL;
66.     SQLHSTMT hstmt = NULL;
67.     SQLRETURN result;
68.     SQLCHAR ConnStrIn[MAXBUFLEN] = "DRIVER={MySQL ODBC 5.1
        Driver};SERVER=127.0.0.1;UID=root;PWD=root;DATABASE=test;CharSet
        =gbk;";
69.     SQLCHAR ConnStrOut[MAXBUFLEN];
70. //分配环境句柄
71.     result = SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE,
        &henv);
72. //设置管理环境属性
73.     result = SQLSetEnvAttr(henv, SQL_ATTR_ODBC_VERSION,
        (void*)SQL_OV_ODBC3, 0);
74. //分配连接句柄
75.     result = SQLAllocHandle(SQL_HANDLE_DBC, henv, &hdbc);
76. //设置连接属性
77.     result = SQLSetConnectAttr(hdbc, SQL_LOGIN_TIMEOUT,
        (void*)LOGIN_TIMEOUT, 0);
78. //连接数据库
79.     result = SQLDriverConnect(hdbc,NULL,
```



```
80.          ConnStrIn,SQL_NTS,
81.          ConnStrOut,MAXBUFLEN,
82.          (SQLSMALLINT
      *)0,SQL_DRIVER_NOPROMPT);
83.    if(SQL_ERROR==result)
84.    {
85.        ShowDBConnError(hwnd,hdbc);
86.        return;
87.    }
88. //初始化语句句柄
89.    result = SQLAllocHandle(SQL_HANDLE_STMT, hdbc, &hstmt);
90.    //SQL_NTS telling the function the previous parameter is Null-Terminated
      String,
91. //please alculate the string length for me
92. result = SQLPrepare(hstmt,(SQLCHAR*)"insert into T_Person(FAge,FName)
      values(20,'kider')",SQL_NTS);
93. CHECKDBSTMTERrror(hwnd,result,hstmt);
94.    result =SQLExecute(hstmt);
95.    CHECKDBSTMTERrror(hwnd,result,hstmt);
96. SQLFreeStmt(hstmt,SQL_CLOSE);
97.    SQLDisconnect(hdbc);
98.    SQLFreeHandle(SQL_HANDLE_DBC,hdbc);
99.    SQLFreeHandle(SQL_HANDLE_ENV,henv);
100. MessageBox(hwnd,TEXT("执行成功"),TEXT("标题"),MB_OK);
101. }
```

[复制代码](#)

## 《C 语言也能干大事》第十七节：数据库开发 4

### 1、处理数据库查询

序号从 1 开始，不是 0

```
1.  
2.  
3.  SQLINTEGER cbsatid=SQL_NTS;  
4.      while (SQLFetch(hstmt)!=SQL_NO_DATA_FOUND)  
5.      {  
6.          SQLCHAR name[20];  
7.          SQLGetData(hstmt,1,SQL_C_CHAR,name,20,&cbsatid);  
8.      }
```

[复制代码](#)

```
1.  SQLINTEGER cbsatid=SQL_NTS;  
2.  while (SQLFetch(hstmt)!=SQL_NO_DATA_FOUND)  
3.  {  
4.      SQLINTEGER i;  
5.      SQLGetData(hstmt,2,SQL_C_LONG,&i,sizeof(SQLINTEGER),&cbsatid);  
6.  }
```

[复制代码](#)

乱码问题解决方式，连接字符串使用 GBK，修改数据库默认编码为 gbk

4、保存 HDBC 到全局变量中，长连接、短连接。比如 Java、C#中访问数据库复杂多少。一通百通。

要掌握调试的技巧，分析问题可能出错的原因和出错的地方，然后调试分析。一定不能和书上不一致，就懵了

5、自己动手写用户管理系统：新增密码字段、新建用户，检验登录；

```
1.  SQLINTEGER cbsatid=SQL_NTS;
2.  //需要一行一行的读取，这种方式就叫做通过游标读取，无论是在 JDBC、
    ADO/ADO.net。。。结果集
3.  //调用 SQLFetch 一次就向下读取一行，直到返回值为
    SQL_NO_DATA_FOUND 的时候表示读到了最后
4.  //是不是和 C 语言文件访问中 EOF 有点像。
5.      while (SQLFetch(hstmt)!=SQL_NO_DATA_FOUND)
6.      {
7.          TCHAR name[20]; //字符数组，SQLCHAR 其实就是 char 的一个别名，
            所以和 char name[20]一样。
8.          //调用 SQLGetData 来取列（Column）的内容
9.          //第一个参数就是代表 SQL 语句的 hstmt、第二是要读取的列的序号（从
            1 开始）
10.         //第三个是列的类型(SQL_C_CHAR 字符串，在 SQLExt.h 中所有以
            SQL_C_开头的都是可选值)
11.         //第四个参数就是要接收的值的指针！！
12.         //第五个是指针指向的缓冲区的大小
13.         //第六个不用管，那么调就行
14.         SQLGetData(hstmt,1,SQL_C_CHAR,name,sizeof(name)/sizeof(SQLC
            HAR),&cbsatid);
15.         MessageBox(hwnd,name,TEXT(""),MB_OK);
16.     }
17.
18. while (SQLFetch(hstmt)!=SQL_NO_DATA_FOUND)
19.     {
20.         SQLINTEGER i; //字符数组，SQLCHAR 其实就是 char 的一个别名，所
            以和 char name[20]一样。
21.         //调用 SQLGetData 来取列（Column）的内容
```

```
22. //第一个参数就是代表 SQL 语句的 hstmt、第二是要读取的列的序号（从
    1 开始）
23. //第三个是列的类型(SQL_C_CHAR 字符串，在 SQLExt.h 中所有以
    SQL_C_开头的都是可选值)
24. //第四个参数就是要接收的值的指针！！
25. //第五个是指针指向的缓冲区的大小
26. //第六个不用管，那么调就行
27.     SQLGetData(hstmt,2,SQL_C_LONG,&i,sizeof(i),&cbsatid);
28.     TCHAR name[20];
29.     SQLGetData(hstmt,1,SQL_C_CHAR,name,sizeof(name)/sizeof(TCHAR),&
        cbsatid);
30.     TCHAR s[20];
31.     wsprintf(s,TEXT("%s 年龄是:%d"),name,i);
32.
33.     MessageBox(hwnd,s,TEXT(""),MB_OK);
34.     }
35. result = SQLPrepare(hstmt,(SQLCHAR*)"select FUserName,FPassword from
    T_User",SQL_NTS);
36. CHECKDBSTMERROR(hwnd,result,hstmt);
37.     result =SQLExecute(hstmt);
38.     CHECKDBSTMERROR(hwnd,result,hstmt);
39. SQLINTEGER cbsatid=SQL_NTS;
40. TCHAR inputUserName[20];
41. GetDlgItemText(hwnd,IDC_EDITUSERNAME,inputUserName,sizeof(input
    UserName)/sizeof(TCHAR));
42. TCHAR inputPassword[20];
43. GetDlgItemText(hwnd,IDC_EDITPASSWORD,inputPassword,sizeof(inputPa
    ssword)/sizeof(TCHAR));
44.
45. BOOL found=FALSE;
46. while (SQLFetch(hstmt)!=SQL_NO_DATA_FOUND)
47.     {
```

```
48.  TCHAR userName[20];
49.  SQLGetData(hstmt,1,SQL_C_CHAR,userName,sizeof(userName)/sizeof(TCHAR),&cbsatid);
50.  TCHAR password[20];
51.  SQLGetData(hstmt,2,SQL_C_CHAR,password,sizeof(password)/sizeof(TCHAR),&cbsatid);
52.  if(0==lstrcmp(inputUserName,userName))
53.  {
54.      if(0==lstrcmp(inputPassword,password))
55.      {
56.          MessageBox(hwnd,TEXT("输入正确，登陆成功！"),TEXT("提示"),MB_OK);
57.          found = TRUE;
58.          break;
59.      }
60.  }
61.  }
62.  if(FALSE==found)
63.  {
64.      MessageBox(hwnd,TEXT("输入错误"),TEXT("报错"),MB_OK|MB_ICONERROR);
65.  }
```

[复制代码](#)

缺点是？？？

数据量大了以后运行速度很慢。比如说咱们如鹏 100 万会员。

Wsprintf

课下作业：

(1)

细化报错信息：当用户名不存在的时候提示“用户名不存在”，当用户名存在而密码错误的时候报告“密码错误”。

(2)

保存用户的时候，如果用户已经存在则更新已有的用户信息；否则新增用户信息。

(3)删除指定用户名的用户；

(4)有能力的同学研究 ListView 控件的使用，列出当前数据库中的所有用户信息，选择某个用户可以删除、可以更新。

下节课内容：

作业点评：ListView 控件的使用与用户管理系统。

SQL 注入漏洞及参数化 SQL

### 《C 语言也能干大事》第十八节：对话框高级操作

板书：

1、

程序中打开新对话框，其实有代码可参考，WinMain。

```
DialogBox(hInstance, MAKEINTRESOURCE(IDD_MAIN), NULL, Main_Proc);
```

只要复制一份 MainDlg.cpp 和 MainDlg.h 改一下就行了

```
DialogBox(hInstance, MAKEINTRESOURCE(IDD_MYDLG), NULL, Dlg1_Proc);
```

怎么传递参数？使用 DialogBoxParam 函数，对比差别。

传入 int；传入字符串；传入结构体。

传出。

怎样判断按下的按钮？通过返回值配合

怎么样在对话框关闭的时候把参数回传回去？SetWindowLong 设置的东西是和窗口的实例绑在一起的。什么是窗口实例？一个窗口打开两个、一个窗口打开两次(隐藏/打开、销毁/打开)。

```
SetWindowLong(hwnd, GWL_USERDATA, (LPARAM)param);
```

1、在资源中插入新的对话框，并且调整控件

2、

```
HINSTANCE hInstance =  
(HINSTANCE)GetWindowLong(hwnd, GWL_HINSTANCE);  
DialogBox(hInstance, MAKEINTRESOURCE(IDD_LOGINDIALOG), NULL,  
Main_Proc);
```

3、每个对话框都有自己的一套 MainDlg.cpp 类似的代码，Main\_Proc、Main\_OnInitDialog 等等

修改 Main\_Proc、Main\_OnInitDialog 等的前缀为 LoginDlg\_\*\*\*\*，函数名不能重复

复制一份 LoginDlg.h

#ifndef \_MAIN\_H 等也要修改

DialogBoxParam

parameter:参数

```
#include "LoginDlg.h"
```

在 OnInitDialog 中处理传入参数，IParam 就是传入的参数

通过 DialogBoxParam 来传入参数，在对话框中的\*\*\*\_OnInitDialog 的 LPARAM IParam 来取参数。

long 类型，指针就是 long 了。

传递字符串

```
TCHAR* buff = "abcd";
```

```
DialogBoxParam(hInstance, MAKEINTRESOURCE(IDD_LOGINDIALOG), NULL,  
LoginDlg_Proc,(LPARAM)buff);//LoginDlg_Proc
```

在对话框之间传字符串（指针）

```
HINSTANCE hInstance =  
(HINSTANCE)GetWindowLong(hwnd,GWL_HINSTANCE);
```

```
LoginData ld;
```

```
ld.userName = "yzk";
```

```
ld.password = "123456";
```

```
//TCHAR* buff = "abcd";
```

```
DialogBoxParam(hInstance, MAKEINTRESOURCE(IDD_LOGINDIALOG), NULL,  
LoginDlg_Proc,(LPARAM)&ld);//LoginDlg_Proc
```

```
BOOL LoginDlg_OnInitDialog(HWND hwnd, HWND hwndFocus, LPARAM  
lParam)  
{
```

```
LoginData* ld = (LoginData*)lParam;
```

```
SetDlgItemText(hwnd,IDC_EDITUSERNAME,ld->userName);
```

```
SetDlgItemText(hwnd,IDC_EDITPASSWORD,ld->password);
```

```
return TRUE;  
}
```

DialogBoxParam 默认是阻塞运行的

可以设置与窗口句柄关联的数据，SetWindowLong 可以看做是把一些数据与窗口管理起来，“让它代为保管”

SetWindowLong、GetWindowLong 来在窗口中保存指针就可以了



只有一个文本框的对话框，有两个按钮【确定】、取消，允许对话框的使用者指定一个函数用来进行校验，能够校验值的正确性，如果不正确还会显示错误信息。提示：函数指针。

EndDialog(hwnd, 1);

在调用代码中通过 DialogBoxParam 的返回值就可以得到 EndDialog 设定的参数

不要重复同样的错误。不要忘了 break。通过调试功能发现的问题。在怀疑的地方加断点调试

LBN\_DBLCLK:DoubleClick

```
if(LBN_DBLCLK==codeNotify)
```

```
{
```

```
//MessageBox(hwnd,TEXT("双击"),TEXT(""),MB_OK);
```

```
//LB_GETCURSEL 消息
```

```
int index = SendDlgItemMessage(hwnd,IDC_LIST1,LB_GETCURSEL,0,0);
```

```
TCHAR buff[255];
```

```
SendDlgItemMessage(hwnd,IDC_LIST1,LB_GETTEXT,index,(LPARAM)buff);
```

```
MessageBox(hwnd,buff,TEXT(""),MB_OK);
```

```
}
```

课后作业：允许用户定制校验策略的 **InputDialog**

2、

Codenotify=LBN\_DBLCLK 用户双击列表框中的字符串。

```
if(LBN_DBLCLK==codeNotify)
```

```
{  
  
int index = SendDlgItemMessage(hwnd,IDC_LIST1,LB_GETCURSEL,0,0);  
  
TCHAR buff[255];  
  
SendDlgItemMessage(hwnd,IDC_LIST1,LB_GETTEXT,index,(LPARAM)buff);  
  
MessageBox(hwnd,buff,TEXT(""),MB_OK);  
  
}  
  
3、  
LBN_SELCHANGE
```

[http://msdn.microsoft.com/en-us/library/bb773169\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/bb773169(VS.85).aspx)

### 《C 语言也能干大事》第十九节：C 语言开发网站 1

- 1、  
C 语言也能开发网站!!! 一通百通!!! 一切语言都是纸老虎!
- 2、
- 3、  
动态 Web 的原理：浏览器向服务器发送请求，服务器响应请求然后将响应的结果以 HTML 的方式发送给浏览器，浏览器将 HTML 显示出来。
- 4、  
本课程讲解的 C 语言开发 Web 的技术叫“CGI”，网上搜索相关资料的关键字“C 语言 CGI”。
- 5、  
课程需要 HTML 的基础，如果 HTML 不熟悉的话请参考《自己动手写网站》的课程，视频下载页面：<http://www.rupeng.com/forum/forum-28-1.html>
- 6、  
环境准备：
  - a)  
下载安装 apache，此软件是 Http 服务器，和 Tomcat、IIS 等类似。推荐下载地址：<http://www.skycn.com/soft/1218.html>。可能遇到的问题，已经安装了 Tomcat、IIS 等软件，将他们卸载（也可以修改 Apache 的端口，不过新手还是直接卸载其他软件比较好）。
  - b)

VC、C-Free、DevC++、TC、PellesC 等任意一款可以开发控制台 C 程序的开发工具（几乎所有 C 开发工具都支持）。课上将使用 VC。

7、

开发第一个 CGI 程序

a)

新建控制台程序

b)

代码：

```
#include "stdafx.h"
```

```
int main(int argc, char* argv[])  
{
```

```
printf("Contenttype:text/html\n\n"); //根据 HTTP 协议，这里一定要有个空行。
```

```
printf("Hello RuPeng!");
```

```
return 0;  
}
```

c)

编译生成 exe 程序（比如 aaa.exe），将 exe 拷贝到 apache 的 cgi-bin 目录，一般是 C:\Program Files\Apache Software Foundation\Apache2.2\cgi-bin

d)

打开浏览器，敲入：<http://127.0.0.1/cgi-bin/aaa.exe>

e)

分析：127.0.0.1 是什么？main 程序就是普通的控制台程序，printf 只是前边多了一句“printf("Contenttype:text/html\n\n");”，暂时不关心它的意义。

f)

每次修改后都要重新拷贝 exe 到 cgi-bin，有没有简单的方法？VC 中设置输出路径：设置（settings）→连接（Link）→输出文件名（Output FileName）。注意每次修改要重新编译才能起作用。

8、

一定注意程序是运行在服务器上的，比如将 CGI 程序放到如鹏的服务器上，那么 aaa.exe 是运行在如鹏服务器上的，只不过是把 printf 的结果打印到用户的浏览器上而已。CGI 程序中可以使用所有的 C 语言特性、函数、调用 ODBC 访问数据库等等。如果不调用 Windows 的函数，则可以一行不动的移植到 Linux/Unix 等操作系统下。

9、

输出完整的 HTML 页面：

```
#include "stdafx.h"
```

```
#include "stdlib.h"//add getenv
```

```
int main(int argc, char* argv[])
```

```
{  
  
printf("Contenttype:text/html\n\n"); //根据 HTTP 协议，这里一定要有个空行。  
  
printf("<html>");  
  
printf("<body>");  
  
printf("hello <a href='http://www.rupeng.com'>RuPeng.com</a>");  
  
printf("</body>");  
  
printf("</html>");  
  
return 0;  
}
```

10、

输出环境变量：

a)

getenv，在 `stdlib.h` 中，

`char * getenv(const char *)`，参数是环境变量的名字，返回值是环境变量的值，都是字符串（确切的说是字符指针）类型。

b)

```
printf("%s ",getenv("SERVER_SOFTWARE"));
```

c)

可选值：

`HTTP_USER_AGENT`

递交表单的浏览器的名称、版本 和其他平台性的附加信息。

`SERVER_NAME`

CGI 脚本运行时的主机名和 IP 地址。

`SERVER_SOFTWARE`

你的服务器的类型如：

`CERN/3.0`

或

`NCSA/1.3.`

`SCRIPT_NAME`

指向这个 CGI 脚本的路径，  
是在 URL 中显示的(如，

/cgi-bin/thescrypt).

#### REMOTE\_HOST

递交脚本的主机名，这个值不能被设置.

#### REMOTE\_ADDR

递交脚本的主机 IP 地址.

#### QUERY\_STRING

脚本参数或者表单输入项(如果是用 GET 递交).

#### QUERY\_STRING

包含 URL 中问号后面的参数.

#### CONTENT\_TYPE

如果表单是用 POST 递交,  
这个值将是  
application/x-www-form-urlencoded.  
在上载文件的表单中,  
content-type  
是个  
multipart/form-data.

#### CONTENT\_LENGTH

对于用 POST 递交的表单,  
标准输入口的字节数.

#### REQUEST\_METHOD

POST  
或  
GET,  
取决于你的表单是怎样递交的.

#### HTTP\_REFERER

递交表单的文本的  
URL，不是所有的浏览器都发出这个信息，不要依赖它

#### HTTP\_ACCEPT

浏览器能直接接收的 Content-types

#### GATEWAY\_INTERFACE

运行的 CGI 版本.

#### SERVER\_PROTOCOL

服务器运行的 HTTP 协议.

SERVER\_PORT

服务器运行的 TCP 口。

11、

怎么无法换行？

“\n”是文本级别的换行，浏览器把“<br/>”才当成换行。但是“a\nb”和“ab”没有不同吗？看一看，分析一下。

12、

请求参数

a)

可以通过地址给 CGI 程序传递参数“aa.exe?name=tom”，参数和 CGI 名称之间用“?”间隔，参数的“名”、“值”之间用“=”间隔

b)

取参数 `printf("%s\n",getenv("QUERY_STRING"));`

c)

打招呼，单个参数：

```
char* pQueryStr=getenv("QUERY_STRING");
```

```
char pName[256];
```

```
sscanf(pQueryStr,"name=%s",pName);
```

```
printf("Hello %s!\n",pName);
```

d)

计算器，多个参数。多个参数键值对之间用“&”分割：

```
char* pQueryStr=getenv("QUERY_STRING");
```

```
int i1,i2;
```

```
sscanf(pQueryStr,"i1=%i&i2=%i",&i1,&i2);
```

```
printf("%i+%i=%i\n",i1,i2,i1+i2);
```

13、

课下练习：

a)

两个参数，“girl=lily&boy=tom”，打印“lily love tom”

b)

登录功能，“username=admin&password=123456”，打印“密码错误”、“登陆成功”。

c)

思考：CGI 程序太危险了，在 CGI 程序中写 `system("format d:")` 就可以格式化浏览者的硬盘了？？？

14、

从 HTML 向 CGI 提交参数

HTML 有头 Head、有身体 Body

表单：form。Please fill the form。注册、登录都是一种申请，都需要 Fill the form。

`<input type="text" name="boy"/>`文本框

`<input type="submit"/>`提交按钮

`action=http://127.0.0.1/love` 表示表单要提交给谁处理。否则大街上随便有人给你表单，你填了 都不知道给谁

乱码问题，后面的课程

14、课后作业：

连接数据库，结合 18 节 ODBC 的知识，开发注册、登录功能的 Web 程序。

判断浏览者的 IP 地址、浏览器类型、操作系统类型等

## 《C 语言也能干大事》第二十节：多线程开发

《C 语言也能干大事》第二十节 多线程开发

1、多线程也是所有语言中都有的一个重要特性，一通百通的一个东西。

2、Sleep 函数简介。为了简化问题，用 Sleep 函数模拟耗时操作，实际项目中用长时间数据库操作、读取大文件、网络操作等替换 Sleep 就可以。Sleep 函数例子：msgbox 1,sleep 50000,msgbox 2。放一个额外 button 上去，更有效果。Sleep 过程中界面是没有响应的，同理长时间的数据库操作、文件操作、网络操作等也会造成没有响应，原理简单分析：消息循环。

3、CreateThread 函数：CreateThread( NULL, 0, ThreadFunc, 0, 0, 0 );一般注意第 3 个参数即可，回调，回调函数格式。Void 指针（可以指向任何类型的数据）。

DWORD WINAPI ThreadFunc( LPVOID lpParam )

```
{  
return 0;  
}
```

改造 msgbox 的例子，暂时第一个参数传 NULL。

在 CreateThread 之后加一句 msg，更明白主线程不会等着子线程。

小例子：取消登录功能的使用。

总结，什么叫线程。普通的执行顺序是顺序执行一句执行完成执行另外一句，但是一旦某一句执行时间很长，后面的就要等；线程（Thread）就是主干执行流程中分出去的一条线，独立于主线程运行，在主线程中只是通过 CreateThread 把线程调度起来，然后就继续向后执行了，不等着线程执行完毕，让分出去的线程自己折腾去。

4、给线程传递参数。回忆：给子对话框传递参数。给子线程传递参数使用第四个参数，子线程函数中 lpParam 参数的值就是传递过来的参数。注意传递的是指针

(1)

CreateThread( NULL, 0, ThreadFunc, TEXT("如鹏网"), 0, 0 );

MessageBox(NULL,(TCHAR \*)lpParam,TEXT(""),MB\_OK);

(2):

```
CreateThread( NULL, 0, ThreadFunc, hwnd, 0, 0 );  
SetDlgItemText(hwnd, IDC_OK, "hello");
```

讲故事：数据导入。课下作业：读取 data.txt，将数据导入数据库，这个整合了线程、ODBC、IO 操作的知识。听明白我讲的东西你只掌握了 5%，只有把这个作业做出来才是真的懂了。实时显示数据处理进度，并且用户可以随时取消数据处理。

## 《C 语言也能干大事》第二十一节：动态链接库

上节课作业点评。待优化点。最好不能每次插入数据都重新连接数据库，因为非常慢。每次插入一条数据要 10 秒。数据库连接是非常耗时的，所以最好是连接一次，然后多次插入，这样速度会快很多。

1、 函数库：把函数组织起来，供程序调用，函数的复用。静态库与动态库。动态库、静态库是其他语言中也都有对应概念的东西，也是一通百通的。C# 中的 Assembly、Java 中的 jar 等。

静态库：编译时代码编译进 exe 中，会使得程序体积非常庞大。不利于模块的共享，比如 ODBC 的库如果是静态库的方式的话计算机里访问数据库的程序都会连接进去 ODBC 的代码，也许你的 Windows 就不止 2、3GB 那么大了，可能 20GB 都不止。优点：不会有 dll hell 的问题。好像“企业间的吞并”

动态库：dll。代码在 dll 中，其他程序调用 dll 中的代码，多个程序可以共享。缺点：dll hell(dll 地狱)，版本问题。好像“企业间的合作”。哪怕不需要被其他程序共享，仍然可以通过 dll 的方式减小单个 exe 的尺寸，并且有利于开发人员的分工。

2、 编写 dll。新建 Win32 dynamic link library

只有在函数前标注：extern "C" \_\_declspec(dllexport) 的函数才能被其他的程序通过 dll 的方式调用

```
extern "C" __declspec(dllexport) int add(int i1,int i2)  
{
```

```
    return i1+i2;
```

```
}
```

(1) 静态链接：新建 win32 DialogBased application。把 dll 和 lib 拷贝到工程目录下：

声明：

```
#pragma comment(lib, "dll1.lib")
```

```
extern "C" __declspec(dllexport) int add(int i1,int i2);
```

然后就可以像调用普通函数一样调用了。

如果 dll 中函数很多的话，一般都是由 dll 开发者开发好.h 文件，把函数的声明都写好，然后供其他人调用，只要 include 就好了。

(2) 动态链接：

//有 \_\_cdecl \_\_stdcall 等不同的调用约定，也就是参数的压栈顺序等，暂时不用关心，只要保证调用的时候和 dll 中的调用约定一样就可以。



//否则会报错: The value of ESP was not properly saved across a function call. This is usually a result of calling a function declared with one calling convention with a function pointer declared with a different calling convention.

```
typedef int(__cdecl* FunctionAdd)(int,int);  
函数指针
```

```
HMODULE hModule;
```

```
FunctionAdd add;
```

```
hModule = LoadLibrary("dll1.dll");//调试时 hModule 为 0x10000000  
If(NULL==hModule )  
{  
    //error.  
}
```

```
add =(FunctionAdd)GetProcAddress(hModule,"add");
```

```
If(NULL==add)  
{  
    //error  
}
```

```
int r = add(1,1);  
Wsprintf msgbox
```

```
FreeLibrary(hModule);
```

2、其他语言也有对应的概念，C#、Java 中的反射。

静态链接和动态链接的区别：静态链接在程序启动的时候就会去检查 dll 是否存在，如果不存在在启动的时候就会报错，程序无法启动；动态链接是在运行到的时候才会去检查是否存在，而且可以由程序员决定在 dll 不存在的时候判断逻辑。

要区分静态库、动态库。静态调用 dll 和动态调用 dll。

3、 插件机制。课下作业：

a. 将你编程中经常用到的重复性代码封装成 dll，并且写验证程序调用它们。

b. 参考机制 FindFirstFile 等函数。导出 GetPluginName、Execute 两个函数。

```
void GetPluginName(TCHAR* name);
```

```
void Execute(...);
```

参考扩展阅读：调用约定：cdecl、stdcall。

## 《C 语言也能干大事》第 22 节：C 语言面向对象开发

1、Dll 的调试。填入主 EXE 的全路径。

GetCurrentDirectory、getcwd。不能依靠他们用他们取得 exe 的文件夹。因为他们是“当前目录”，而不是“exe 所在的目录”（Linux、DOS：目录，Windows：文件夹）。

GetModuleFileName()：当前 EXE 文件的全路径。D:\俺的文档\培训\用 C 语言写有用的东西\第二十二节\zuoye\Debug\zuoye.exe

strcpy

指向同一个数组的指针可以做减法，结果就是两个指针的距离。

通配符（搜一下）

\*.dll 所有 dll 文件

\*.txt 所有 txt 文件

a.\*就是文件名是 a，后缀任意的文件

a\*.txt 就是所有文件名以 a 开头的文本文件。

C 语言：堆数据、栈数据。很多公司面试、笔试必考内容。.

调试的时候注意设置活动工程。

类：属性（字段）、方法（Method、也就是函数）

函数名字一样，但是参数不一样，函数的重载（overload）。把对象的指针放在函数的第一位，约定而已。

2、作业：开发如鹏 tools，最基本的记事本的功能，增加一个“插件”菜单项，菜单项的内容是动态根据插件目录中的插件构造出来的，可能的插件有：文字转大写、文字转小写、文字去空格、从一个网址中粘贴文字、插入计算器计算结果、自动定时保存记事本中的文件等。

自己写出来，扩展成为如鹏的一个开源项目。如鹏 NotePad

3、C 语言面向对象开发

a) 面向对象不是代替面向过程的，面向对象是面向过程的一个补充和发展，面向对象的本质仍然是面向过程。

b) 面向对象只是一种组织数据和算法的方法思维方式而已，与语言无关，Java、C#等面向对象的语言只不过是从语法层面上简化了面向对象代码的编写而已，C 之类的面向过程的语言也能实现面向对象。

c) 面向对象的三个特征：封装、继承、多态。许多书上来就把封装、继承、多态稀里哗啦的讲完了，这是不符合认知规律的。只有编写过大量面向过程的代码以后，再来学封装，才能真正的掌握封装；只有熟练应用封装一段时间以后才能真正的掌握继承，而只有编写过大量的使用封装、继承的代码以后才能真正的掌

握多态。所以不要急于求成，咱们的课只讲 C 语言实现面向对象中的封装。对于初学者来说把封装用好了、用熟了就可以。高级的东西确实看起来 NB，也很好炫耀，但是功力不到位的话玩儿高级的东西很容易走火入魔。

d) 通过 C 语言学面向对象更不容易像学 Java、C#那样把类当成多高级的技术来学的人，学 Java、C#的人很容易写出“伪类”来。

类：一堆有共同特征对象的抽象，对象就是真实的东西。人只是抽象，看不到摸不着的，杨中科才是一个人的具体的实例，才是对象。

### 5、第一个类

a) C 语言进行面向对象开发并不会引入新的语法。结构体、指针、函数。

b) 用结构体实现类的定义

```
struct Person
{
```

```
    TCHAR* name;
```

```
    int age;
```

```
};
```

对比 C#中 Person 类的定义：

```
Class Person
```

```
{
    Public String Name{set;get;}
    Public int Age{set;get;}
}
```

创建一个 Person 类的实例：Person p1;

c) 为成员变量赋值：p1.name="tom";p1.age=20;

d) 创建一个 Person 类的方法：void SayHello(Person p)

e) 创建一个增长年龄的方法：void IncreaseAge(Person p)，为什么会出现这种现象。结构体是值传递的，克隆人，因此在需要改变结构体状态的环境中（比如面向对象编程）必须使用结构体指针。这一点在其他语言中也有类似的问题，也是一个易错点，因此需要注意。理解一下学 C 语言时 Swap:交换两个变量。

f) PPerson、Person\*; typedef Person \*PPerson; typedef struct Person{} \*PPerson。

g) 约定，把类的实例指针做为方法的第一个参数。

h) 类的多个实例。

设计一个矩形类，包含相交判断、相交运算等

Rectangle: 长、宽，左上角坐标。X,y,height,width.两个方法：计算矩形的面积，一个是计算两个矩形是否有相交的部分。（是某年的微软笔试题。）