## 第四篇 网络与通信

## 4.1 静态 IP 自动配置的实现

当前随着网络技术的飞速发展,网络管理也从软件 和硬件甚至是网络构架上都有了重大的变革,但是在实际的运用中,还有很多经典的技术继续使用。例如静态 分配主机 IP 地址这一经典局域网地址分配方案仍然在 很多的网络构架中延续使用,很多的网管人员仍然在维 护着这类的网络,如何轻松地做好静态 IP 地址分配仍 然是一个比较烦琐的工作。在实际的工作中结合单位特 点,通过编写客户端软件的方式实现了自动配置主机 IP 地址。

## 1 需求分析

在实际工作中,虽然在 Windows 环境中配置静态 IP 地址对于一个熟悉操作系统的用户非常简单,但是由于 以下两点原因使得一般用户很难自行配置一个合法可 用的 IP 地址。

(1)由于大多数的用户在家中使用的是 xDSL 拨号接入网络,因此大多数的普通用户均没有配置过 IP 地址,甚至对于 IP 地址的概念都不清楚,那么仅仅通过网管部门的电话指导或者相关宣传手册是不容易让用户完成静态 IP 地址的配置工作的。

(2) 在实际的工作中,单位中不同的部门使用的 静态 IP 地址参数是不相同的,除了 IP 地址不能重复外, 不同网段接入的主机可能还要求网管、DNS 等参数都不 相同,因此,很多情况下需要通过电话联系网管部门, 这样造成网管部门大量的人力花在电话指导配置 IP 地 址这样的琐碎小事上,严重影响了网管部门员工的工作 积极性。

因此对于网管部门工作而言就希望能够让配置静态 IP 地址的工作进一步的简单,但对于一台尚未接入 网络的主机,任何的网络服务都是不可用的,所以使用 独立的客户端软件是解决这类问题的唯一可行的方案。

## 2 软件设计

在 Windows 环境中开发应用程序的工具很多,但是

考虑到实际应用环境,即编译后的程序要能够在 Windows 2000/XP/Vista 等平台上无安装绿色运行,那么 使用 Visual C++ 6.0 比较好的选择。

注意: 自 1998 年发布了 Visual Studio 6.0 至今,微 软已经针对 Visual Studio 6.0 发布了 6 个补丁包对开发 平台进行了更新,并且随着 Windows XP 等操作系统的 更新,随 Visual Studio 6.0 发布的 API 函数库已经不能 完全满足当前的开发,因此需要独立安装最新的平台 SDK,本设计就是用了 Microsoft Platform SDK February 2003 这个版本的 API 作为开发基础库,这也是微软最后 一套能够支持 Visual C++ 6.0 的平台 SDK。下载地址: http://www.microsoft.com/msdownload/platformsdk/sdkup date/psdk-full.htm。

#### 2.1 关键技术分析

在 Windows 平台中修改指定网卡上的 IP 地址的方法比较多,但要求不重新启动操作系统的方法主要有以下三种:

#### 2.1.1 命令行使用 netsh 命令

Netsh 是命令行脚本实用工具,它允许从本地或远程显示或修改当前正在运行的计算机的网络配置,因此可以直接使用命令或者实用教本编写网络 IP 地址修改,例如下面的命令将配置名为"本地连接"的接口,其静态 IP 地址为 10.0.5.99,子网掩码为 255.255.255.0,默认网关为 10.0.5.1。

Netsh set address name="本地连接" source=static addr=10.0.5.99 mask=255.255.0 gateway=10.0.5.1

2.1.2 使用微软未公开的 API 进行设置

自从 Windows 发布以来, 微软除了开放了大量的系统 API 共程序设计者使用, 但由于种种问题还有很多 API 并未正式发布过, 其中在 Windows 2000 以后的操作系统中就包含了修改网络适配器的 API 函数, 函数原形如下:

DhcpNotifyConfigChange [DllImport("dhcpcsvc.dll")]

public static extern bool DhcpNotifyConfigChange( string lpwszServerName, // 本地机器为 NULL string lpwszAdapterName, // 适配器名称 bool bNewIpAddress, // TRUE 表示更改 IP long dwIpIndex, // 指明第几个 IP 地址,如果只有 //该接口只有一个 IP 地址则为 0 long dwIpAddress, // IP 地址

long dwSubNetMask, //子网掩码

long nDhcpAction); //对 DHCP 的操作 0:不修 //改, 1:启用 DHCP, 2:禁用 DHCP

在 C+++中通过动态调用 DLL 中 API 函数的操作就可以使用了。

2.1.3 使用 WMI (Windows Management Instrumentation)中Win32\_NetworkAdapterConfiguration 类实现 IP 地址管理

Windows Management Instrumentation (WMI) 是 Microsoft WBEM 的实现,通过企业网络为访问和共享 管理信息主动建立的标准。WMI 符合 WBEM 标准, 并为"公用信息模型(CIM)"(它是描述存在于管理环 境中的对象的数据模型)提供完整的支持。WMI 可以 被 Microsoft 技术和工具使用,如可以与编程或脚本系 统(如 Windows 脚本宿主)一起使用 WMI 来检索大 部分计算机系统方面的详细配置信息(包括服务器应用 程序),或者对系统的配置进行更改。

在 WMI 中 Win32\_NetworkAdapterConfiguration 类 实现了大多数的网络适配器的配置工作,可以通过 EnableStatic 方法设置 IP 地址和子网掩码;通过 SetGateways方法设置网关;通过SetDNSServerSearchOrder 方法设置 DNS 服务器地址。

由于 WMI 使用了 CIM 作为实现基础,因此在以上的类和方法均使用一种接口定义语 MOF (Managed object Format,受管对象格式)来准确描述模型中的类和关联,以上的方法 MOF 描述如下:

uint32 EnableStatic(
 [in] string IPAddress[],
 [in] string SubnetMask[]
);
uint32 SetGateways(
 [in] string DefaultIPGateway[],
 [in] uint16 GatewayCostMetric[]
);
uint32 SetDNSServerSearchOrder(
 [in] string DNSServerSearchOrder[]
);

以上三种方案均可以实现在 Windows 环境中不重 新启动修改网络适配器参数,但是使用 netsh 命令行在 C++程序中调用效率不高,而使用微软未公开得 API 实现可能会受到操作系统变更的影响,为了支持个不 同版本的 Windows 本方案使用 WMI 作为本程序实现 的基础。

#### 2.2 程序开发

通过以上的分析,本程序主要需要实现的功能是在 Windows 环境中根据给定的配置参数对指定的网络适 配器进行配置,程序流图如下:



图1所示的程序流图中表明了程序开始首先等待用 户的输入,即选择接入网络的物理地址,然后程序根据

配置文件中的数据选择恰当的配置参数,这些参数被预 先存放在 Config.ini 文件中,需要注意的是 IP 地址参数 分为两种情况,一种是确定的 IP 地址,另一种是仅提 供 IP 地址的前 3 部分,最后一部分为 0,如 192.168.1.0。 因为在实际网络中不可能配置这样的 IP 地址,程序会 自动判断参数,如果 IP 不确定那么会随机生成 IP 的最 后一部分,然后程序尝试配置,如果发生 IP 地址冲突 那么程序将对原 IP 进行自增一的操作后尝试 IP 设置, 直到成功配置一个合法的 IP 地址。

为了便于使用 WMI,这里将所有的有关 WMI 的操 作封装到 CWMI 类中,如图 2 所示。

CWMI
#chAdapterName
#ClassPath
#hr
#index
#InstancePath
#MethodName
#NetWorkResource
#pClass
#pEnumerator
#pInClass
#pInInst
#pLocator
#pNamespace
#pOutInst
+CWMI()
+~CWMI ()
#Clean()
#CreateOneElementBstrArray(inout v : tagVARIANT*, in s : unsigned shortconst *)
#EnableStatic(inout IPAddress : unsigned short*, inout SubnetMask : unsigned short*)
+setAdapterNameQStringA(inout lpAdapterName : char*)
+SetDns(inout DNS1 : unsigned short*, inout DNS2 : unsigned short*)
#SetDNSServerSearchOrder(inout DNS1 : unsigned short*, inout DNS2 : unsigned short*)
+SetGate(inout GW : unsigned short*)
#SetGateways(inout GW : unsigned short*)
#setIndexByAdapterName()
+SetIP(inout IPAddress : unsigned short*, inout SubnetMask : unsigned short*)

图 2 CWMI 类

在图 2 中, chAdapter、ClassPath 等成员变量均为 CWMI 类中成员函数的功能实现使用的公共变量,具体 功能可以参考源代码。需要指出的是为了方便使用, CWMI 类仅提供了 SetIP、SetDns 和 SetGate 三个公共成 员函数分别实现 IP 地址设置、DNS 服务器设置和网关 设置。

SetIP、SetDns 和 SetGate 三个成员函除均是通过 C++调用 WMI 组件中对应的方法实现的,其实现基本 类似,这里仅以 SetIP 作简要说明,源码如下:

```
void CWMI::SetIP(LPWSTR IPAddress, LPWSTR
SubnetMask){
```

```
setIndexByAdapterName();
EnableStatic(IPAddress,SubnetMask);
}
```

setIndexByAdapterName 是 CWMI 类实现的成员函数,实现了对当前主机的网络适配器的选择操作。

```
EnableStatic 成员函数是对 WMI 中 Win32_
NetworkAdapterConfiguration类中EnanleStatic方法的封装,源代码如下:
```

HRESULT CWMI::EnableStatic(LPWSTR IPAddress,LPWSTR SubnetMask){

```
ULONG Status=0;
NetWorkResource = SysAllocString(L"root\\cimv2");
WCHAR AdapterIndex[128];
wsprintf(AdapterIndex,
  L"Win32_NetworkAdapterConfiguration=%ld",index);
InstancePath = SysAllocString(AdapterIndex);
ClassPath = SysAllocString(L"Win32_Network
  AdapterConfiguration");
MethodName = SysAllocString(L"EnableStatic");
LPCWSTR Arg1Name = L"IPAddress";
VARIANT var1;
LPCWSTR Arg2Name = L"SubnetMask";
VARIANT var2;
hr = CoInitializeEx(NULL,COINIT_
  MULTITHREADED);
if(FAILED(hr)){
  MessageBox(NULL,L"CoInitializeEx",
      L"Error",MB OK);
  Clean();
  return hr;
```

}

CreateOneElementBstrArray(&var1,IPAddress); CreateOneElementBstrArray(&var2,SubnetMask); CoCreateInstance(CLSID WbemLocator, 0, CLSCTX INPROC\_SERVER, IID\_IWbemLocator, (void\*\*)&pLocator); hr = pLocator->ConnectServer(NetWorkResource,NULL, NULL, NULL, 0, NULL, NULL, &pNamespace); hr = CoSetProxyBlanket(pNamespace, RPC\_C\_AUTHN\_WINNT, RPC\_C\_AUTHZ\_NONE, NULL. RPC\_C\_AUTHN\_LEVEL\_CALL, RPC C IMP LEVEL IMPERSONATE, NULL EOAC\_NONE); if(FAILED(hr)) { Clean();MessageBox(NULL, L"CoSetProxyBlanket",L"Error",MB OK); return hr; } else hr = pNamespace->GetObject(ClassPath, 0, NULL, &pClass, NULL); if(FAILED(hr)){Clean();MessageBox(NULL, L"GetObject",L"Error",MB OK);return hr; } else hr = pClass->GetMethod(MethodName, 0, &pInClass, NULL): if(FAILED(hr)){ Clean();MessageBox(NULL, L"GetMethod",L"Error",MB\_OK);return hr; } else hr = pInClass->SpawnInstance(0, &pInInst); if(FAILED(hr)) { Clean();MessageBox(NULL,L"SpawnInstance", L"Error",MB\_OK);return hr; } else hr = pInInst->Put(Arg1Name, 0, &var1, 0); if(FAILED(hr)){ Clean();MessageBox(NULL,L"Put", L"Error",MB\_OK);return hr; } else hr = pInInst->Put(Arg2Name, 0, &var2, 0); if(FAILED(hr)){ Clean();return hr; } else{ VARIANT resultParameter; ULONG Status: hr = pNamespace->ExecMethod(InstancePath, MethodName, 0, NULL, pInInst, &pOutInst, NULL);

```
if(SUCCEEDED(pOutInst->Get(L"ReturnValue",0,
          &resultParameter,NULL,NULL))){
          Status = resultParameter.ulVal;
          VariantClear(&resultParameter);
          TRACE1("返回值:%lu\n",Status);
        }
        else{
          TRACE0("参数未能获得");
          MessageBox(NULL,L"ExecMethod",
            L"Error",MB_OK);
      }
      VariantClear(&var1);
      VariantClear(&var2);
      Clean():
      return Status:
    其中 CreateOneElementBstrArray 是 CWMI 类的成
员函数,实现了封装一个 BSTR 数组,代码如下:
    void CWMI::CreateOneElementBstrArray(VARIANT* v,
LPCWSTR s){
      SAFEARRAYBOUND bound[1];
      SAFEARRAY* array;
      bound[0].lLbound = 0;
      bound[0].cElements = 1;
      array = SafeArrayCreate(VT_BSTR, 1, bound);
      long Index = 0;
      BSTR bstr = SysAllocString(s);
      SafeArrayPutElement(array, &Index, bstr);
      SysFreeString(bstr);
      VariantInit(v);
      v->vt = VT_BSTR | VT_ARRAY;
      v->parray = array;
```

#### 2.3 设计技巧

使用 WMI 测试程序进行 WMI 测试, Windows Management Instrumentation (WMI) 测试程序,也叫做 WBEMTest,是一个在 Windows Management Instrumentation (WMI) 服务程序和 WMI 应用程序开发期间,用于查 看和更改公共信息模型 (Common Information Model, CIM)类、实例和方法的常规工具。

在编写本软件程序的过程中,往往需要首先测试获 得获取当前系统的某些参数,如果直接在程序中编码调 试将增加设计难度,因此可以首先在 WMI 测试程序中 进行。例如获取当前网络适配器信息就可以通过以下步 骤获得:从"开始"→"运行",或者是命令行提示符 下输入 wbemtest。

单击"连接"按钮,在名称空间后输入\root\cimv2, 并单击"连接"按钮,如图3所示。



图 3 连接名称空间

单击"查询"按钮,并输入需要测试的查询语句, 例如"SELECT \* FROM Win32\_NetworkAdapter Configuration",单击查询后即可获得查询结果,如图 4 所示,单击每个查询结果可以查看每个对象的详细属性等 信息。

muL: SELECI * FROM #IN32_MetworkAdapterConfiguration	关闭
14 个对象 批处理最大值: 1C 完毕	
Win32_NetworkAdapterConfiguration_Index=1	
Win32_NetworkAdapterConfiguration.Index=3 Win32_NetworkAdapterConfiguration.Index=4	
min32_MetworkAdapterConfiguration_index=5 Win32_NetworkAdapterConfiguration_Index=6 Win32_NetworkAdapterConfiguration_Index=7	
Win32 NetworkAdapterConfiguration. Index=9 Win32 NetworkAdapterConfiguration. Index=10	
Win32_NetworkAdapterConfiguration. Index=11 Win32_NetworkAdapterConfiguration. Index=12	
Win32_NetworkAdapterConfiguration. Index=13	<u> </u>
添加(a) I MBA m) I	

图 4 查询结果

使用 WBEMTest 主要来解决 WMI 和依赖于 WMI 的程序中的问题,可以执行以下任务:

- \* 创建、列举及删除 CIM 类和实例。
- \* 执行方法。
- \* 运行查询。

\* 显示关于类和实例的受管理对象格式(Managed Object Format, MOF)代码。

### 3 应用

按照以上的思路和方法可以编写出一个自动静态

IP地址分配设置工具,经过编译后运行界面如图5所示。



图 5 静态 IP 自动设置程序

启动程序后依次选择网卡和主机所在位置后单击 "自动设置"即可将相应的网络参数配置到指定的网络 适配器中。其中不同位置的网络信息参数预先已经存放 在数据文件 config.ini 文件中,每一小节配置对应图 5 中主机所在位置下拉选框中数据,配置格式如下:

[我的办公室配置] IP\_Address=192.168.1.0 MASK=255.255.255.0 GW\_Address=192.168.1.1 DNS1=192.168.1.1 DNS2=192.168.1.2 [教学楼] IP\_Address=192.168.2.0 MASK=255.255.255.0 GW\_Address=192.168.2.1 DNS1=192.168.1.1 DNS2=192.168.1.2

经过实际应用和测试本程序获得了用户的一致好 评,特别是在禁止了 UAC (User Account Control)的 Vista 环境中也能正常工作,大大简化了 IP 地址的配置 过程。本方案虽然从本质上不能解决静态 IP 网络中 IP 地址的分配问题,但是通过适当的管理措施加上本程序 的应用,也最大限度的降低了网管员在相关工作的劳动 强度,并且通过客户端的编写进一步深刻理解了 WMI 等网管技术。

## 4.2 IE 快捷菜单扩展编程实现

#### 陈建丽 宋克强

现在工作中大家经常到互联网上去查询需要的各种资料。为了方便以后的使用,在找到需要的资料后一般都要将相应的网页保存在自己的计算机里面。但是在

使用 IE 本身保存网页的过程中也觉得有一个小小的不便: IE 只能将整个网页完全保存,而不能只保存网页中的部分内容。而自己需要的往往只是网页中的部分内容

(特别是文字信息),如果每次都保存整个网页的话, 不仅浪费磁盘空间也浪费了自己的时间。因此自然而然 地就想,要是能够只保存选中的网页内容就好了。经过 一段时间的摸索,终于用扩展 IE 快捷菜单的方法实现 了预期的部分功能。

## 1 扩展 IE 的功能

要能够保存网页中的选定部分内容,必须对 IE 的 现有功能进行扩展。常见的扩展 IE 现有功能的方法主 要有:扩展 IE 的工具菜单、扩展 IE 的工具栏,以及扩 展 IE 的快捷菜单等。在此选择使用扩展 IE 快捷菜单的 方法。

## 1.1 修改注册表

在注册表的分支HKEY\_CURRENT\_USER\Software\ Microsoft\Internet Explorer\MenuExt 中建立一个项,用于 在 IE 的快捷菜单中显示可选择的命令。

例如,此处建立的项为:

HKEY\_CURRENT\_USER\Software\Microsoft\Intern et Explorer\MenuExt\保存选定的网页内容

表示在 IE 的快捷菜单中会增加一个命令"保存选 定的网页内容"。

## 1.2 设置默认值

用于指定当用户选择 IE 快捷菜单中相应命令时要执行的脚本文件的 URL。该脚本文件可以通过操作代表快捷菜单所在窗口的对象 window.external.menuArguments, 进而处理 IE 窗口的各种对象(如 document、location、 history 等)。 此处将其默认值建立为

#### $C:\\\\\\SaveSelectedHTML.htm$

则表示当用户选定 IE 快捷菜单中的命令"保存选 定 的 网 页 内 容"时 系 统 将 自 动 执 行 文 件 C:\SaveSelectedHTML.htm。

### 1.3 指定扩展菜单

如果愿意,还可以在第一步建立的注册表项中建立 一个名为"Contexts"、类型为"DWORD"的值,用于 指定扩展的菜单命令在何种情况下才会出现在 IE 的快 捷菜单中。该值可以是如表1所示几个值的逻辑"或" (OR)运算的组合。

= 1

衣工				
值	快捷菜单的类型			
0x1	缺省快捷菜单			
0x2	图像的快捷菜单			
0x4	控件的快捷菜单			
0x8	表单域的快捷菜单			
0x10	选中文字的快捷菜单			
0x20	锚点的快捷菜单			

例如,如果想在图像和选定文本的快捷菜单中显示 扩展的命令,则可以将 Contexts 的值设置为 0x12。由 于此处希望可以保存 IE 中选中的任何内容,因此不设 置 Contexts,这样扩展的命令随时都会出现在 IE 的快 捷菜单中。

现在,注册表中对应位置如图1所示。

《 注册表编辑器					_ 0 ×
注册表(R) 编辑(E) 查看(V) 收藏(E) 帮助(H)					
😑 🧰 MenuExt		名称	类型	数据	
—————————————————————————————————————	_	四(默认)	REG_SZ	C:\SaveSelectedHTML.htm	
😥 😥 🧰 Security	-	4			
我的电脑\HKEY_CURRENT_USER\Software\Microsoft\Intern	net Expl	orer(MenuExt)(	果存选定的网	页内容	8,

图1 添加注册表项

## 2 设计

通过上面的步骤,在 IE 的快捷菜单中就会出现命 令"保存选定的网页内容",当用户选择该命令的时 候 就 可 以 执 行 包 含 可 执 行 脚 本 的 文 件 (C:SaveSelectedHTML.htm)。下面就进行该脚本程序 的设计了虽然需要的功能实际上都可以在该文件中用 脚本语言(如 JavaScript、VBScript)直接实现,但是这 样会有一些限制。因为程序会涉及到读写本地文件,如 果直接用脚本语言读写本地文件,有可能被杀毒软件误 认为病毒程序而被删除;即使不被杀毒软件删除,也有可能因为 IE 的安全设置的原因而被禁止运行。另外,也不能自由地设置文件的名字和保存位置等信息。

为解决这些问题,可以使用 VB 等程序先建立并注 册一个 COM 服务器,该服务器用于实现系统需要的接 口,本例主要是实现文件的读写等操作; IE 在运行时通 过在脚本文件中调用相应的 COM 对象的方法从而实现 需要的功能。

下面以 Visual Basic 和 JavaScript 为例说明程序的具体实现过程。

(1) 启动 VB 6.0, 选择 "File-New Project-ActiveX

DLL", 创建一个 ActiveX DLL 文件, 设置工程名为 "SaveHTML"。

(2) 将默认的类 class1 的 Name 设置为 Saver, 然 后为该类加入如下代码,实现接口 Savelt。

Public Sub SaveIt(Recieved As String) 'Recieved 用于接收 传递过来的参数,即选定的网页内容的 HTML 代码

SelectedHTML = Recieved '将传递过来的参数赋值给公共 变量 MainForm.Show vbModal End Sub

iu Sub

(3)选择 "Project-Add Module", 添加一个模块 Module1, 在其中加入以下公共变量的声明语句:

Public SelectedHTML As String '公共变量,用于保存选 定的网页内容的 HTML 源代码

(4)选择 "Project-Add Form", 添加一个窗体和需要的控件,设置其属性如表 2 所示。

Ŧ	$\mathbf{a}$
রহ	2

对象	属性	设置值	作用
	Name	MainForm	用户操作界面,用于设置文件的保
Form	Caption	设置文件保存位置和文件名	存位置和文件名
	Border Style	3-Fixed Dialog	
Label	Caption	保存位置	記想示作田 使田野认的 Name
Label	Caption	文件名	起还小作用, 使用颜 Khi Name
TortDor	Name	TextLocation	显示用户选择的文件保存位置
TextBox	Locked	True	
TextBox	Name	TextName	用于输入文件名
CommandButton	Name	CommandBrowse	单击该按钮可选择文件保存位置
	Caption	选择文件夹	
CommandButton	Name	CommandStart	单击该按钮则保存选定内容
	Caption	开始保存	

设置好后该窗体的界面如图2所示。

。设置文件保存位置和文件名	×
保存位置	选择文件夹
文件名	开始保存

#### 图 2 程序运行界面

下面为窗体加入如下一些声明和代码: '用于显示选择文件夹浏览窗口的常数、类型和 API Private Const BIF RETURNONLYFSDIRS = 1 Private Const BIF DONTGOBELOWDOMAIN = 2 Private Const MAX\_PATH = 260 Private Declare Function SHBrowseForFolder Lib "shell32" (lpbi As BrowseInfo) As Long Private Declare Function SHGetPathFromIDList Lib "shell32" (ByVal pidList As Long, ByVal lpBuffer As String) As Long Private Declare Function lstrcat Lib "kernel32" Alias "lstrcatA" (ByVal lpString1 As String, ByVal lpString2 As String) As Long Private Type BrowseInfo hWndOwner As Long pIDLRoot As Long pszDisplayName As Long lpszTitle As Long

ulFlags	As Long
lpfnCallback	As Long
lParam	As Long
iImage	As Long
End Type	
Private LastFileNam	ne As String

'以下为具体的实现代码

Private Sub CommandBrowse\_Click() '选择保存文件的路径 TS= SelectedDir() If TS ↔ '''' then TextLocation.Text =TS endif End Sub Private Function SelectedDir() As String '自定义函数,显示 文件夹选择对话框并返回用户的选择 Dim lpIDList As Long Dim sBuffer As String Dim sBuffer As String Dim szTitle As String Dim tBrowseInfo As BrowseInfo szTitle = "选择文件的保存位置" With tBrowseInfo .hWndOwner = Me.hWnd

.lpszTitle = lstrcat(szTitle, "") .ulFlags = BIF\_RETURNONLYFSDIRS + BIF\_DONTGOBELOWDOMAIN

End With

lpIDList = SHBrowseForFolder(tBrowseInfo)

If (lpIDList) Then sBuffer = Space(MAX\_PATH) SHGetPathFromIDList lpIDList, sBuffer sBuffer = Left(sBuffer, InStr(sBuffer, vbNullChar) - 1) SelectedDir = sBuffer End If End Function Private Sub CommandStart\_Click() '开始保存选定内容

If TextName.Text = "" Then TI = MsgBox("错误! 必须指定保存文件的名字。", vbOKOnly, "文件名为空")

Else

- TStart = "<HTML>" + vbCrLf + "<head>" + vbCrLf + "<title></title>" + vbCrLf + "</head>" + vbCrLf + "<body>" 'HTML 网页文件的前半部分 TSEnd = "</body>" + vbCrLf + "</HTML>" 'HTML 网页文
- 件的后半部分
- TSResult = TStart + SelectedHTML + TSEnd '将选定内容 的 HTML 代码加入中间,构成最终的网页文件的 HTML 代码 「下面将 HTML 代码写入最终的 HTML 文件中
- Open TextLocation.Text + "\" + TextName.Text + ".htm" For Output As #1

Print #1, TSResult

Close #1

TI=MsgBox("选定网页内容保存成功。", vbOKOnly, "提示") Unload Me End If

End Sub

Private Sub Form Load()

TextLocation.Text = App.Path '启动时设置默认的保存位 置为 DLL 所在文件夹

End Sub

(5)选择"File-Save Project",以系统默认的名字 保存整个工程(包括窗体、模块、类);然后选择 "File-Make SaveHTML.DLL",生成最终的 ActiveX DLL文件。

(6) 注册刚刚生成的 ActiveX DLL。在系统文件 夹 Windows\System32 中找到文件 regsvr32.exe,将

SaveHTML.DLL 拖到其上方,即可出现注册成功的提示。 (7)用记事本等编辑软件建立内容如下的 HTML

文件 SaveSelectedHTML.HTM,并将其保存到 C:\。

<SCRIPT LANGUAGE="JavaScript">

var HTMLObj;

HTMLObj =new ActiveXObject("SaveHTML.Saver");//建 立 COM 对象

HTMLObj.SaveIt(window.external.menuArguments.docume nt.selection.createRange().htmlText);//调用对象的接口,保存选定内容到本地文件

</SCRIPT>

现在打开一个网页,选定其中的一部分内容,然 后右击,就会出现如图 2 所示的对话框;选择好文件 的保存位置,设定好文件的名字,然后单击"开始保 存",选定部分的内容就会保存在你的计算机里了。

以上程序用 VB 6.0 和 JavaScript 配合实现,在 Windows 2000/XP 环境下均验证通过。

#### 3 结语

由于程序只是保存了选定部分的 HTML 代码,因此 网页中的图像、声音等对象的 URL 都是 Internet 上的地 址。这样在 IE 脱机的情况下虽然文字可以正常显示, 但是图像等对象却只能显示为占位符;只有在 IE 连接 互联网的情况下才能正常显示图像等对象。为解决此问 题,可以在脚本中通过 window.external.menuArguments. document 对象的 images 数组元素的 src 属性来获得图像 的 URL,通过 window.external.menuArguments.document 对象的 embeds 数组元素的 src 属性来获得其他多媒体 对象的 URL,然后将这些 URL 作为参数传递给 COM 对象,由 COM 对象完成这些文件的下载,并修改 HTML 代码中这些文件的 Internet 地址为本地地址。限 于篇幅,在此只是提出实现思路,有兴趣的读者可以 自己做一下。

## 4.3 网页摘文快车程序设计

#### 国 林

## 1 概述

浏览网页时,当看到一篇有用或有趣的文字,想把 它"当"下来时,有的朋友选择将网页直接"另存为"保 存,而有些朋友则选择:选中文本→复制→新建记事本 →粘贴→保存→重命名,殊不知这样做无形中浪费了上 网时间, 久而久之, 浪费的就不只是一点点琐碎的时间了。 设计本程序的目的就是为了方便用户, 提高用户上 网摘文得效率, 节约宝贵的时间。VC 是 Microsoft 的一 个非常好用的 C++编程工具, 现在已经出到 Microsoft Visual Studio 2008, 大多数的 C++程序员都喜欢用 VC, 不但开发环境友好, 而且有功能强大的 MFC(微软基 础类库, Microsoft Foundation Class) 支持。然而, MFC 在.NET 平台下没有太多的改进。因此,大部分的 VC 程序员都喜欢用 VC++ 6.0 开发 MFC 应用程序。

### 2 程序设计

首先,创建一个基于 MFC 的对话框界面的应用程 序,命名为 SuperGet。

#### 2.1 皮肤设计

俗话说:"人靠衣装,马靠鞍"。软件同样应该具备 一个友好、美观的用户界面。然而,VC 的界面设计和 时间处理方法非常复杂,无法在有限的时间内设计出美 观、用户熟悉的操作界面。

于是在这里使用 USkin 来设计皮肤。

USkin 的特性:

1) 支持 20 多种 Windows 标准控件。

2) 支持文件、颜色、打印、字体等标准对话框。

3) 支持第三方/自定义控件。

4) 支持所见即所得的皮肤文件编辑。

5) 支持 MDI/SDI/Dialog 风格的应用程序。

6) 支持 WindowsBlind 皮肤文件导入,多达万个现 成皮肤供您使用。

7) 支持颜色主题, 一套皮肤可以演变成多种皮肤。

8) 支持多种编程语言。

9) 支持多线程。

10) 支持动态换肤。

11)支持所有弹出菜单,包括 Edit 控件右键弹出菜单。

12) 只需三行代码即可实现换肤。

13) 支持 ListView/TreeView 等控件滚动条换肤。

14)皮肤资源文件所占空间小。

(1)包含 USkin.lib 和 USkin.h 头文件。首先,将
 USkin.lib 和 USkin.h 文件拷贝到工程目录下,然后工程
 →设置→Link→L 对象/库模块中,添加 USkin.lib,接着
 在 SuperGetDlg.h 文件中#include "USkin.h"。

(2)加载皮肤文件。首先,将皮肤文件(这里以 ToMorning.u3 为例)拷贝到工程目录下(编译后将 ToMorning.u3 和 USkin.dll 放到 Debug 文件夹中),然后 在 SuperGet.cpp 添加如下代码:

BOOL CSuperGetApp::InitInstance()

£

```
AfxEnableControlContainer();
SetRegistryKey(_T("应用程序向导生成的本地应用程序"));
USkinInit(NULL,NULL,_T("ToMorning.u3"));
//加载 u3 皮肤文件
CSuperGetDlg dlg;
m_pMainWnd = &dlg;
INT_PTR nResponse = dlg.DoModal();
if (nResponse == IDOK)
{
// TODO: 在此放置处理何时用"确定"来关闭
```

```
//对话框的代码
```

```
}
else if (nResponse == IDCANCEL)
```

// TODO: 在此放置处理何时用"取消"来关闭 //对话框的代码

USkinExit(); //USkin 退出

由此,编译→构建→生成,一个漂亮的对话框界面 就生成了。

绘制界面如图1所示。

保存位重:	2000 100 100 100 100 100 100 100 100 100	
文件名命名:		
③自动捕	提文本内容的文件名	R
○用数字	编号文件名 从 🗵 🚔 开始	
○用户自	命名文件名	1
🗌 自动取	消文本中的空行、空格	说
说明		1
Ctrl+Alt+Z	复制所选文本内容,粘贴在记事本中,按指定格式命名文件并 保存到指定位置。	
A1++7	调用此执键,可对网网候存的文件进行重命名。	

图1 绘制界面

#### 2.3 程序代码

2.3.1 主要的函数

```
PerFormCtrlC(); 模拟 Ctrl+c 按键
   void
   BOOL
           GetCopy(); 获得复制到剪贴板中的文本内容
          File_Save(); 将得到的文本内容保存到指定位置
   void
   void
          CutBlank(); 消除文本中的空格
   LRESULT OnHotKey(WPARAM wp,LPARAM lp);
   注册热键
   void
          Release Focus();释放本程序的焦点
   2.3.2 函数说明
   1) 模拟 Ctrl+c 按键
   void CSuperGetDlg::PerFormCtrlC()
   ł
      EmptyClipboard();
                       //清空剪贴板
                       //延时 1s
   Sleep(1000);
   //模仿键盘击键 Ctrl+C
   keybd event(VK CONTROL,MapVirtualKey(VK CONTR
OL,0),0,0); //按住 Ctrl, 不放开
   keybd event(0x43,MapVirtualKey(0x43,0),0,0);
   //按住C键,不放开
     keybd_event(0x43,MapVirtualKey(0x43,0),
       KEYEVENTF_KEYUP,0);
                                //放开C键
    keybd_event(VK_CONTROL,MapVirtualKey
       (VK CONTROL,0), KEYEVENTF
        KEYUP,0); //放开 Ctrl
   2) 从剪贴板中获得复制文本
   BOOL CSuperGetDlg::GetCopy()
     BOOL m_bcopy=TRUE;
                          //设置 copy 为成功
```

205

```
//打开系统剪贴版
    if(!OpenClipboard())
      AfxMessageBox("无法打开剪贴版");
      m bcopy=FALSE;
                         //copy 失败
    }
     //判断剪贴版上的数据是否是制定的数据格式
    if(IsClipboardFormatAvailable(CF TEXT)||IsClipboardFor
matAvailable(CF OEMTEXT))
   {
     //从剪贴版上获取数据
     HANDLE hClipboardData=GetClipboardData(CF TEXT);
      //通过给内存句柄加锁,获得指向指定格式数据的指针
      char * pchData=(char *)GlobalLock(hClipboardData);
      //本地变量获得数据
      strClipBoard=pchData;
      //给内存句柄解锁
      GlobalUnlock(hClipboardData);
      if(!m Blank)
      strData=LPCTSTR(strClipBoard);
      //关闭剪贴版
      CloseClipboard();
    }
      else
      {
        AfxMessageBox("剪贴版中无文本数据");
        m bcopy=FALSE;
                             //copy 失败
      }
      return m_bcopy;
    }
    3)添加消除空行的函数
    void CSuperGetDlg::CutBlank()
    ł
      //处理文本中的空行、空格
        CStringArray arr;
                             //定义 CStringArray 数组
      CString strtemp;
      TCHAR * pszBuf=NULL;
                             //Tchar 类型指针
      pszBuf=(TCHAR *)LPCTSTR(strClipBoard);
      //pszBuf指向文本
      char * p;
      for (p=strtok(pszBuf,"\r\n");p!=NULL;
    p=strtok(NULL,"\r\n")) /*分解 pszBuf 字符串为一组标记
串。pszBuf 为要分解的字符串, \r\n 为分隔符字符串。返回指
向下一个标记串,当没有标记串时则返回空字符 NULL。*/
      -{
       arr.Add(p);
                   //将得到的字符串添加到 arr 数组中
      3
       //删除空行
      for (int i=0;i<arr.GetSize();i++)
                     //将每一个字符串赋给 strtemp
       strtemp=arr[i];
       if(strtemp.IsEmpty()) //如果 strtemp 为空
         arr.RemoveAt(i); //移除此项
         i--:
                        //i 自减
         continue;
       }
         //删除空格
```

```
int i=1:
       while(j>0)
   {
     strtemp.TrimLeft(); //没有参数,从左删除字符空格
     strtemp.TrimRight(); //没有参数,从右删除字符空格
     j = strtemp.Replace(" ","");
     //用无替换字符串中间的空格
   3
   strData+=strtemp+"\r\n";
   //将无空行、空格的文本传给 strData
4) 将文本保存在指定位置
void CSuperGetDlg::File_Save()
Ł
 CStdioFile File; //CStdioFile 类型变量 File
 NameFile(m Namenumber); //给文件命名
 CString FileName=m SavePlace+"//"+strName+".txt";
 //FileName 得到文件全路径
 if (!File.Open(FileName, CFile::modeWrite)
   CFile::typeText)) //创建文本文件
   if (!SetCurrentDirectory(m_SavePlace))
   //检测是否有此目录
     CreateDirectory(m_SavePlace, NULL);
     //如果没有则创建
   if(!File.Open(FileName,CFile::modeCreate)
     CFile::modeWrite|CFile::typeText))
   //如果不能打开打开
   ł
     MessageBox(m_SavePlace+strName+".txt"+"不能被
       创建","信息提示",MB_OK); //显示错误提示
   }
   else
   -{
     File.SeekToEnd(); //先定位到文件尾部
     File.Write(strData,strData.GetLength()+1);
     //将内容写入文件中
     File.Close(); //关闭文件
   }
 }
}
5) 注册快捷键
LRESULT CSuperGetDlg::OnHotKey
    (WPARAM wp,LPARAM lp)
ł
 if(HIWORD(lp)==90&&LOWORD(lp)==
   MOD ALT|MOD CONTROL) //响应 Ctrl+Alt+Z
   ReleaseFocus();
                        //释放本程序焦点
   PerFormCtrlC();
                        //模拟 Ctrl+c 按键
   UpdateData(TRUE);
                        //更新数据
   strData.Empty();
                        //清空 strData
   BOOL bCopy=GetCopy(); //拷贝剪贴板中的文本内容
```

206

```
if(m_Blank)
      CutBlank();
                        //清除空行
                                                          gFocus = wnd;
                                                          //使本进程和激活窗口的进程联系起来,接收本进
   if(bCopy)
                                                          //程的按键消息
   -{
     if(!m SavePlace.IsEmpty())
                                                          AttachThreadInput(
                                                            GetWindowThreadProcessId(m hWnd,NULL),
    File Save();
                                                            GetWindowThreadProcessId(gFocus,NULL),
  else
   AfxMessageBox("请选择保存路径");
                                                            TRUE);
   }
                                                       3
 else if(HIWORD(lp)==88&&LOWORD(lp)==
   MOD_ALT)
                    //响应 Alt+X
                                                      return CDialog::PreTranslateMessage(pMsg);
  ł
   ReNameFile();
                    //重命名文件名字
                                                        //释放本程序的窗口激活状态
 3
                                                        void CSuperGetDlg::ReleaseFocus()
 return 0;
                                                         ł
                                                          if(IsWindow(gFocus))
3
                                                          {
6) 激活当前程序
                                                            HWND wnd = ::GetForegroundWindow();
BOOL CSuperGetDlg::PreTranslateMessage(MSG* pMsg)
                                                            if(IsWindow(wnd))
                                                             £
 HWND wnd = ::GetForegroundWindow();
                                                               if(wnd == gFocus)
 if(IsWindow(wnd))
                                                              ł
 {
                                                               return;
   //如果当前激活窗口不是本程序的窗口
                                                                }
   if(wnd != this->m_hWnd)
                                                              3
   {
                                                              //设置保存的焦点窗口处于激活状态
     //激活窗口发生了改变
                                                        ::SetForegroundWindow(gFocus);
     if(gFocus != wnd)
                                                        ::SetFocus(gFocus);
     {
                                                      3
       if(IsWindow(gFocus))
                                                     }
       ł
                                                     由于篇幅所限,本程序的具体功能就不一一介绍
        //释放本进程和 gFocus 线程的联系
         AttachThreadInput(
                                                 了,有兴趣的朋友可以阅览本程序的源代码。本摘文快
          GetWindowThreadProcessId
                                                 车 (SuperGet) 在 VC++ 6.0, Windows XP 环境下调试
          (m_hWnd,NULL),
                                                 通过,编译并运行本程序,摘文快车的界面效果如图 2
          GetWindowThreadProcessId(gFocus,NULL),
                                                 所示。
          FALSE);
       }
```





3 结语

复制→粘贴→保存→重命名过程。读者在本程序的基 础上可添加更多的功能,实现功能更加完善的摘文 快车。

本文详细的介绍了在 VC++环境下, 网页摘文中

## 4.4 用 Java 实现 Yahoo 天气预报客户端

吴亚峰 刘 佳 李志昕

1 Yahoo 天气预报

Yahoo 作为大型的中文门户网站之一,提供了种类 繁多的各种信息服务,其中也包括了非常实用的天气预 报信息服务。非常值得注意的是,Yahoo 提供的天气预 报信息不但可以通过登录 Yahoo 的网站浏览,也可以获 取特定格式的 XML 文档自行解析。有了第二种天气预 报服务,就可以自己开发天气预报的客户端软件,或把 天气预报功能嵌入到自己的桌面软件或 Web 应用中。

获取 Yahoo 天气预报的 XML 文档非常简单,只要 使用如下格式的 URL 即可:

http://xml.weather.yahoo.com/forecastrss?u=c&p=CH XX0008

其中查询参数 u 表示温度单位,若摄氏度则为"c", 华氏度则为"f"。查询参数 p 表示地点,后跟城市代码。 城市代码是一个 8 位长度的字符串,前四位固定为 "CHXX",后四位为数字,表示不同的城市。例如, "0008"表示北京,常见的城市代码如表 1 所列。

代码	城市	代码	城市
CHXX0008	北京	CHXX0044	杭州
CHXX0116	上海	CHXX0502	海口
CHXX0037	广州	CHXX0131	唐山
CHXX0138	武汉	CHXX0321	扬州

表1 城市代码

输入了上述的 URL 后,就可以获得 Yahoo 提供的 当天北京地区天气预报信息的 XML 文档,其基本内容 如下:

<?xml version="1.0" encoding="UTF-8"

```
standalone="yes" ?>
```

<rss version="2.0"

xmlns:yweather ="http://xml.weather.yahoo.com/ns/rss/1.0" xmlns:geo ="http://www.w3.org/2003/01/geo/wgs84\_pos#"> <channel>

<!-- 此处省略了一些不重要代码-->

<yweather:wind chill="32"

direction="130" speed="11.27" />

```
<yweather:atmosphere humidity="49"
```

visibility="999.00" pressure="999" rising="0" />

```
<yweather:astronomy
sunrise="4:53 am" sunset="7:46 pm"/>
```

<!-- 此处省略了一些不重要代码 --> <item> <!-- 此处省略了一些不重要代码 --> <pubDate>Sun, 06 Jul 2008 3:00 pm CST</pubDate> <yweather:condition text="Fair" code="34" temp ="32" date="Sun, 06 Jul 2008 3:00 pm CST" /> <!-- 此处省略了一些不重要代码 --> <yweather:forecast day="Sun" date="6 Jul 2008" low="22" high="31" text="Sunny" code="32" /> <yweather:forecast day="Mon" date="7 Jul 2008" low="22" high="31" text="Partly Cloudy" code = "30" /> <!-- 此处省略了一些不重要代码 --> </item> </channel>

</rss>

上述文档中包含了当时的天气情况信息、当天以及 明天的天气预报信息,详细说明如下所列:

(1) yweather:wind 标记中的 chill 属性表示当前的 风冷热系数, direction 属性表示当前的风向, speed 表示 当前的风速。

(2) yweather:atmosphere 标记中的 humidity 属性 表示当前的湿度, visibility 属性表示当前的能见度, pressure 属性表示当前的气压。

(3) yweather:astronomy 标记中的 sunrise 属性表示 当天日出时间, sunset 属性表示当天日落时间。

(4) pubDate 标记中的内容是天气预报的发布时间。

(5) yweather:condition 标记中的 code 属性表示当前的天气情况代码, temp 属性表示当前温度。

(6)两个 yweather:forecast 标记中包含的分别是今 天与明天的天气预报信息,其中 date 属性表示日期, low 属性表示最低温度, high 属性表示最高温度, code 属性 表示天气情况代码。

## 2 案例功能

将结合 Yahoo 天气预报客户端的开发来介绍 JAXP (Java API for XML Processing)的简单使用,下面首先 对本软件的功能、界面进行简单的介绍。

#### 2.1 程序界面

程序中主要包含两个窗体,第一个用于选择天气预 报的城市(如图 1 所示),第二个用来显示获取的天气 预报信息(如图 2 所示)。





图 2 天气预报信息显示窗口

## 2.2 软件实现的功能

(1)程序启动后首先显示的是如图 1 所示的城市 选择窗口,用户可以在其中选择需要的城市,并单击"确 定"按钮。

(2) 按下确定按钮后,程序通过网络连接 Yahoo 的服务器,获取所需的天气预报 XML 文档。

(3) 获取 XML 文档后,软件进行解析,并将当前 天气信息,今明两天的天气预报信息显示在界面上。

## 3 城市选择窗体的开发

城市选择窗体的主要功能就是让用户选择需要天 气预报的城市,其核心就是供城市选择的下拉列表框。 为了方便维护以及程序的可扩展性考虑,下拉列表中的 城市信息将从一个自定义的XML文档CityList.xml中加 载,其基本内容如下:

<?xml version="1.0" encoding="GBK" standalone="yes" ?> <CityList> <cityItem> <code>CHXX0008</code><name>北京</name> </citvItem> <cityItem> <code>CHXX0116</code><name>上海</name> </cityItem> <citvItem> <code>CHXX0037</code><name>广州</name> </cityItem> <cityItem> <code>CHXX0138</code><name>武汉</name> </cityItem> <cityItem> <code>CHXX0044</code><name>杭州</name> </cityItem>

<citvItem> <code>CHXX0502</code><name>海口</name> </citvItem> <citvItem> <code>CHXX0131</code><name>唐山</name> </cityItem> <cityItem> <code>CHXX0321</code><name>扬州</name> </cityItem> </CityList> 使用 XML 文档的好处是,未来需要添加新的城市 时不需要修改源代码,只需要在 XML 中添加新的 cityItem 元素即可。 有了记录城市信息的 XML 文档后就可以对城市 选择窗体类 YahooWeatherFramePre 进行开发了,其代 码框架如下: package wyf; import java.util.\*; import javax.swing.\*; import java.awt.\*; import java.awt.event.\*; import javax.xml.parsers.\*; import org.w3c.dom.\*; import java.io.\*; import java.net.\*; public class YahooWeatherFramePre extends JFrame{ Image image;//窗体图标 JLabel jl=new JLabel("请选择城市"); //城市选择下拉列表 JComboBox jcb=new JComboBox(); JButton jb=new JButton("确定"); //用来封装每个城市名称及代码的内部类 public static class CityItem{ String cname;//城市名称 String ccode;//城市代码 // CityItem 内部类构造器 public CityItem(String cname,String ccode) {this.cname=cname; this.ccode=ccode;} //重写 toString 方法 @Override public String(){return cname;} // YahooWeatherFramePre 类构造器 public YahooWeatherFramePre() { //对标题和 logo 图片进行初始化 this.setTitle("Yahoo 天气预报客户端"); image=new ImageIcon("img/ico.png").getImage(); this.setIconImage(image); //对界面中的控件进行摆放 this.setLayout(new FlowLayout(FlowLayout.LEFT,10,6)); jl.setPreferredSize(new Dimension(200,20)); this.add(jl); jcb.setPreferredSize(new Dimension(150,20)); this.add(jcb); jb.setPreferredSize(new Dimension(60,20)); this.add(jb); //对城市列表初始化 this.initCityList(); //为"确定"按钮添加监听器 jb.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e)

#### {

//获取当前选中的城市

CityItem currCI = (CityItem)jcb.getSelectedItem(); String selectedCitvCode = currCL.ccode: //未来在此处添加创建天气预报信息显示 //窗口的代码..... YahooWeatherFramePre.this.dispose(); }): //设置窗体首次出现的大小和位置 Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize(); int centerX=screenSize.width/2;//屏幕中央 x 坐标 int centerY=screenSize.height/2; //屏幕中央 y 坐标 int w=250://本窗体宽度 int h=100;//本窗体高度 //设置窗体出现在屏幕中央 this.setBounds(centerX-w/2,centerY-h/2-100,w,h); this.setVisible(true); this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); this.setResizable(false);//设置窗体不可修改大小 //从 CityList.xml 文件中加载城市列表 public void initCityList(){} //主方法 public static void main(String args[]) {new YahooWeatherFramePre();} } 上述代码中"ImageIcon("img/ico.png")"加载了窗 体的 logo 图片,因此要将 ico.png 图片文件存放到 img 文件夹下, img 文件夹放置在 class 包所在目录下, 否则 程序不能正确执行。

在完成了代码框架的开发后就可以开发加载城市 列表 XML 文档 CityList.xml 的 initCityList 方法了,其 代码如下:

public void initCityList()

ł try // 为解析 XML 文件创建 DOM 对象 DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance(); DocumentBuilder builder = factory.newDocumentBuilder(); //解析 XML 文件 Document doc = builder.parse(new File("CityList.xml")); // 规格化 doc.normalize(); //解析城市列表 NodeList cityItems = doc.getElementsByTagName("cityItem"); int cityCount=cityItems.getLength(); CityItem[] cia=new CityItem[cityCount]; for(int i=0;i<cityCount;i++){ Element cityElement = (Element)cityItems.item(i);

String ccode= cityElement.getElementsByTagName("code") .item(0).getFirstChild().getNodeValue(); String cname= cityElement.getElementsByTagName("name") .item(0).getFirstChild().getNodeValue(); cia[i]=new CityItem(cname,ccode);} jcb.setModel( new DefaultComboBoxModel(cia)); }catch(Exception e){e.printStackTrace();} }

上述 initCityList 方法将 CityList.xml 文件中的每个 cityItem 加载为一个 CityItem 类对象,并添加到 CityItem 数组中,最终将数组添加到 DefaultComboBoxModel 对 象以显示到下拉列表中。完成 YahooWeatherFramePre 类的开发后,将代码编译运行一下,可以见到如图 1 所 示的城市列表选择窗体。

要注意的是,由于上述代码访问了 CityList.xml 文件,因此要将此文件放到 class 包所在目录下,否则程序不能正确执行。

## 4 显示窗体

完成了城市列表窗体的开发后,就可以着手开发用 于显示天气信息的窗体类 YahooWeatherFrame,其代码 框架如下:

package wyf; import java.util.\*; import javax.swing.\*; import java.awt.\*; public class YahooWeatherFrame extends JFrame{ Font font= new Font("楷体 GB2312",Font.PLAIN,18); //左侧显示详细天气情况的面板 JPanel jpz=new JPanel(); //右上侧显示今天天气预报的面板 JPanel jps=new JPanel(); //右下侧显示明天天气预报的面板 JPanel jpx=new JPanel(); JsplitPane jspy = new JSplitPane( JSplitPane.VERTICAL\_SPLIT,jps,jpx); JSplitPane jspz = new JSplitPane( JSplitPane.HORIZONTAL\_SPLIT,jpz,jspy); //当前天气情况的各个显示标签 JLabel jlTodayImg=new JLabel();//天气情况图片 JLabel jlTodayTitle=new JLabel();//天气总情况描述 JLabel jlTodayDegree=new JLabel();//当前气温 JLabel jlTodayChill=new JLabel();//当前风冷热系数 JLabel jlTodaySpeed=new JLabel();//当前风速 JLabel jlTodayDirection=new JLabel();//当前风向 JLabel jlTodayHumidity=new JLabel();//当前湿度 JLabel jlTodayPressure=new JLabel();//当前气压 JLabel jlTodayVisibility=new JLabel();//当前能见度 JLabel jlTodaySunrise=new JLabel();//日出时间 JLabel jlTodaySunset=new JLabel();//日落时间

210

JLabel jlTodayPublishTime=new JLabel()://发布时间 //今天天气预报情况的各个显示标签 JLabel ilTomorrowTime=new JLabel()://今天时间 //今天情况图片 JLabel jlTomorrowImg=new JLabel(); //今天最高温度 JLabel jlTomorrowHigh=new JLabel(); //今天最低温度 JLabel jlTomorrowLow=new JLabel(); //明天天气预报情况的各个显示标签 JLabel jlDATTime=new JLabel();//明天时间 JLabel jlDATImg=new JLabel();//明天情况图片 JLabel jlDATHigh=new JLabel();//明天最高温度 JLabel jlDATLow=new JLabel();//明天最低温度 public YahooWeatherFrame(String cityCode, Image image){ //对界面进行初始化 this.initWeatherFrame(image); //未来添加解析天气并填充到界面代码...... //设置窗体首次出现的大小和位置 Dimension screenSize= Toolkit.getDefaultToolkit().getScreenSize(); int centerX=screenSize.width/2;//屏幕中央 x 坐标 int centerY=screenSize.height/2;//屏幕中央 y 坐标 int w=620;//本窗体宽度 int h=358;//本窗体高度 //设置窗体出现在屏幕中央 this.setBounds(centerX-w/2,centerY-h/2-100,w,h); this.setVisible(true); //设置窗体的默认关闭动作 this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); this.setResizable(false); //对界面进行初始化的方法 public void initWeatherFrame(Image image){} 在完成了 YahooWeatherFrame 类框架的开发后,就 可以开发对界面控件进行初始化的 initWeatherFrame 方 法了,其代码如下: public void initWeatherFrame(Image image) //对标题和 logo 图片进行初始化 this.setTitle("Yahoo 天气预报客户端"); this.setIconImage(image); //设置三个面板的背景色 jpz.setBackground(new Color(231,231,231)); jps.setBackground(new Color(252,237,180)); jpx.setBackground(new Color(176,237,255)); //对总界面进行初始化 this.add(jspz,BorderLayout.CENTER); jspz.setDividerLocation(300); jspy.setDividerLocation(165); jspz.setDividerSize(0); jspy.setDividerSize(0); //对本日天气界面进行初始化 jpz.setLayout(null); jlTodayImg.setBounds(50,5,175,120); jpz.add(jlTodayImg);jlTodayTitle.setFont(font);

jlTodayTitle.setBounds(5,130,260,20); jpz.add(jlTodayTitle);jlTodayDegree.setFont(font); jlTodayDegree.setBounds(5,150,260,20); jpz.add(jlTodayDegree);jlTodayChill.setFont(font); jlTodayChill.setBounds(150,150,150,20); jpz.add(jlTodayChill);jlTodaySpeed.setFont(font); jlTodaySpeed.setBounds(5,170,180,20); jpz.add(jlTodaySpeed);jlTodayDirection.setFont(font); jlTodayDirection.setBounds(190,170,170,20); jpz.add(jlTodayDirection); jlTodayHumidity.setFont(font); jlTodayHumidity.setBounds(5,190,170,20); jpz.add(jlTodayHumidity); jlTodayPressure.setFont(font); jlTodayPressure.setBounds(100,190,170,20); jpz.add(jlTodayPressure); ilTodayVisibility.setFont(font); jlTodayVisibility.setBounds(5,210,170,20); jpz.add(jlTodayVisibility); jlTodaySunrise.setFont(font); jlTodaySunrise.setBounds(5,230,150,20); jpz.add(jlTodaySunrise); jlTodaySunset.setFont(font); jlTodaySunset.setBounds(5,250,150,20); jpz.add(jlTodaySunset); jlTodayPublishTime.setFont(font); jlTodayPublishTime.setBounds(5,270,290,20); jpz.add(jlTodayPublishTime); //对明日天气界面进行初始化 jps.setLayout(null); jlTomorrowTime.setFont(font); jlTomorrowTime.setBounds(5,5,300,20); jps.add(jlTomorrowTime); jlTomorrowImg.setBounds(55,20,175,120); jps.add(jlTomorrowImg); jlTomorrowHigh.setFont(font); jlTomorrowHigh.setBounds(5,140,175,20); jps.add(jlTomorrowHigh); jlTomorrowLow.setFont(font); jlTomorrowLow.setBounds(155,140,175,20); jps.add(jlTomorrowLow); //对后天天气界面进行初始化 jpx.setLayout(null); jlDATTime.setFont(font); jlDATTime.setBounds(5,5,300,20); jpx.add(jlDATTime); jlDATImg.setBounds(55,20,175,120); jpx.add(jlDATImg);jlDATHigh.setFont(font); jlDATHigh.setBounds(5,140,175,20); jpx.add(jlDATHigh);jlDATLow.setFont(font); jlDATLow.setBounds(155,140,175,20); jpx.add(jlDATLow); } 完成了上述代码的开发后将 YahooWeatherFrame 类

进行编译,并在 YahooWeatherFramePre 类代码框架中 "未来在此处添加创建天气预报信息显示窗口的代 码....."处添加如下代码:

new YahooWeatherFrame(selectedCityCode, image);

```
添加完上述代码后,再次编译 YahooWeatherFramePre
类并运行,此时再单击"确定"按钮就可以看到天气预
报窗体了,不过此时程序还不能从网络中获取天气预报
信息解析显示。
```

## 5 信息解析类

完成了两个窗体类的开发后,就可以进行天气预报 信息解析类 YahooWeather 的开发了,其代码框架如下: package wyf; import javax.xml.parsers.\*; import org.w3c.dom.\*; import java.io.\*; import java.net.\*; import javax.swing.\*;import java.text.\*; import java.util.\*; public class YahooWeather //获取天气信息并解析填充的方法 public static void parseWeather (YahooWeatherFrame ywf, String cityCode) {} //将天气情况代码转换为文字信息的方法 public static String fromCodeToText(int code){} //将风向角度转化为文字信息的方法 public static String fromDegreeToDirectionStr(int degree){} 从上述代码中可以看出, YahooWeather 是一个工具 类,其中提供了用于获取解析天气的3个静态方法。首 先应该开发将风向角度转化为文字信息的 fromDegreeToDirectionStr 方法,其代码如下: public static String fromDegreeToDirectionStr(int degree) if(degree>337 && degree<=360){return "北风";} else if(degree>=0 && degree<=22){return "北风";} else if(degree>22 && degree<=67){return "东北风";} else if(degree>67 && degree<=112){return "东风";} else if(degree>112 && degree<=157){return "东南风";} else if(degree>157 && degree<=202){return "南风";} else if(degree>202 && degree<=247){return "西南风";} else if(degree>247 && degree<=292){return "西风";} else if(degree>292 && degree<=337){return "西北风";} else {return "错误的方向值";} } 开发完将风向角度转化为文字信息的 fromDegree ToDirectionStr 方法后,就可以开发将天气情况代码转换 为文字信息的 fromCodeToText 方法了,其代码如下: public static String fromCodeToText(int code){ switch(code){ case 0: return "龙卷风";case 1: return "热带风暴"; case 2: return "飓风";case 3: return "雷暴雨"; case 4: return "雷雨";case 5: return "雨夹雪"; case 6: return "雨夹冰雹";case 7: return "雪夹冰雹"; case 8: return "冻毛毛雨";case 9: return "毛毛雨"; case 10: return "冻雨";case 11: return "阵雨";

case 12: return "阵雪";case 13: return "雪小雪"; case 14: return "小雪骤雨";case 15: return "风雪"; case 16: return "雪";case 17: return "冰雹"; case 18: return "雨夹雪";case 19: return "尘"; case 20: return "雾";case 21: return "霾"; case 22: return "黑烟";case 23: return "狂风"; case 24: return "风";case 25: return "冷"; case 26: return "多云"; case 27: return "夜间大部多云"; case 28: return "日间大部多云"; case 29: return "夜间部分多云"; case 30: return "日间部分多云"; case 31: return "夜间晴朗";case 32: return "日间晴朗"; case 33: return "夜间清晰";case 34: return "日间清晰"; case 35: return "雨夹雹";case 36: return "热"; case 37: return "局部雷阵雨";case 38: return "零星雷阵雨"; case 39: return "零星雷阵雨";case 40: return "零星暴雨"; case 41: return "大雪";case 42: return "零星暴雪"; case 43: return "大雪";case 44: return "部分多云"; case 45: return "雷阵雨";case 46: return "阵雪"; case 47: return "局部地区雷阵雨"; case 3200: return "不可用"; default: return "没有此代码"; } } 在两个用于信息转换的辅助方法都开发完成后,就 可以开发用于获取并解析显示天气 XML 文档的 parseWeather 方法了,其代码如下: public static void parseWeather (YahooWeatherFrame ywf,String cityCode){ try{ //为解析 XML 文件创建 DOM 对象 DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance(); DocumentBuilder builder= factory.newDocumentBuilder(); //远程 URL 解析 URL url=new URL("http://xml.weather.yahoo."+ "com/forecastrss?u=c&p="+cityCode); Document doc=builder.parse(url.openStream()); doc.normalize(); //规格化 //解析天气预报发布时间 NodeList links = doc.getElementsByTagName("pubDate"); String dateStr = links.item(0).getFirstChild().getNodeValue(); SimpleDateFormat sdf=new SimpleDateFormat( "EEE, dd MMM yyyy K:mm aa zzz",Locale.US); Date date=sdf.parse(dateStr); Calendar cd=Calendar.getInstance(); cd.setTime(date); dateStr=(cd.get(Calendar.YEAR))+"年" +(cd.get(Calendar.MONTH)+1)+"月" +(cd.get(Calendar.DATE))+"日" +cd.get(Calendar.HOUR\_OF\_DAY)+"时" +((cd.get(Calendar.MINUTE)>9)?cd.get(Calendar.MINUTE )+"":"0"+cd.get(Calendar.MINUTE))+"分";

ywf.jlTodayPublishTime.setText( "发布时间:"+dateStr); //解析风相关信息 links =doc.getElementsByTagName("yweather:wind"); NamedNodeMap nnm=links.item(0).getAttributes(); ywf.jlTodayChill.setText("风冷热系数:" +nnm.getNamedItem("chill").getNodeValue()); ywf.jlTodaySpeed.setText("风速:" +nnm.getNamedItem("speed").getNodeValue() +"公里/小时"); String directionStr= nnm.getNamedItem("direction").getNodeValue(); int directionInt=Integer.parseInt(directionStr); ywf.jlTodayDirection.setText("风向:" +fromDegreeToDirectionStr(directionInt)); //解析大气相关信息 links =doc.getElementsByTagName( "yweather:atmosphere"); nnm=links.item(0).getAttributes(); ywf.jlTodayHumidity.setText("湿度:"+ nnm.getNamedItem("humidity").getNodeValue()+"%"); ywf.jlTodayPressure.setText("气压:"+ nnm.getNamedItem("pressure").getNodeValue()+"毫巴"); ywf.jlTodayVisibility.setText("能见度:"+ ((int)Double.parseDouble(nnm.getNamedItem("visibility").g etNodeValue())\*100)/10000.0+"公里"); //解析天文学有关信息 links =doc.getElementsByTagName("yweather:astronomy"); nnm=links.item(0).getAttributes(); ywf.jlTodaySunrise.setText("日出时间:" +nnm.getNamedItem("sunrise").getNodeValue().replaceAll( "\\s","")); ywf.jlTodaySunset.setText("日落时间:+ nnm.getNamedItem("sunset").getNodeValue().replaceAll("\\ s","")); //解析当前天气相关信息 links =doc.getElementsByTagName("yweather:condition"); nnm=links.item(0).getAttributes(); int code= Integer.parseInt( nnm.getNamedItem("code").getNodeValue()); ywf.jlTodayTitle.setText("当前天气情况:" +fromCodeToText(code)); ImageIcon ii=new ImageIcon("img\\"+code+".png"); ywf.jlTodayImg.setIcon(ii); ywf.jlTodayDegree.setText("当前气温:"+ nnm.getNamedItem("temp").getNodeValue()+"°C") //解析天气预报信息 links =doc.getElementsByTagName("yweather:forecast"); //解析今天天气预报信息 nnm=links.item(0).getAttributes(); code=Integer.parseInt(nnm.getNamedItem("code").getNode Value()); ii=new ImageIcon("img\\"+code+".png"); ywf.jlTomorrowImg.setIcon(ii); sdf=new SimpleDateFormat("dd MMM yyyy",Locale.US); date=sdf.parse(

nnm.getNamedItem("date").getNodeValue()); cd=Calendar.getInstance(); cd.setTime(date); dateStr=(cd.get(Calendar.YEAR))+"年" +(cd.get(Calendar.MONTH)+1)+"月" +(cd.get(Calendar.DATE))+"日"; ywf.jlTomorrowTime.setText("今天:"+dateStr+" " +fromCodeToText(code)); ywf.jlTomorrowHigh.setText("最高温度:"+ nnm.getNamedItem("high").getNodeValue()+"°C"); ywf.jlTomorrowLow.setText("最低温度:"+ nnm.getNamedItem("low").getNodeValue()+"°C"); //解析明天天气预报信息 nnm=links.item(1).getAttributes(); code=Integer.parseInt( nnm.getNamedItem("code").getNodeValue()); ii=new ImageIcon("img\\\"+code+".png"); ywf.jlDATImg.setIcon(ii); sdf=new SimpleDateFormat("dd MMM yyyy",Locale.US); date=sdf.parse(nnm.getNamedItem("date").getNodeValue()); cd=Calendar.getInstance(); cd.setTime(date); dateStr=(cd.get(Calendar.YEAR))+"年" +(cd.get(Calendar.MONTH)+1)+"月" +(cd.get(Calendar.DATE))+"日"; ywf.jlDATTime.setText("明天:"+dateStr +" "+fromCodeToText(code)); ywf.jlDATHigh.setText("最高温度:"+ nnm.getNamedItem("high").getNodeValue()+"°C"); ywf.jlDATLow.setText("最低温度:"+ nnm.getNamedItem("low").getNodeValue()+"°C"); }catch(Exception e){e.printStackTrace();} }

此方法中使用 JAXP 中的 DOM 方式对从网络上获 取的天气信息 XML 文档进行解析,并将解析的结果填 充到界面中的各个控件进行显示。

要特别注意的是,在显示天气情况时为了生动形象,采用了图片。对每种不同的天气情况代码各提供一幅图片,统一存放在 img 文件夹中,图片的名称即为情况代码,如图 3 所示。



图 3 天气情况图片

另外要注意将 img 文件夹与 class 文件包放在同一个目录中,否则程序无法正确运行。

完成了 YahooWeather 类的开发后,将其进行编译, 并在 YahooWeatherFrame 类代码框架中"未来添加解析 天气并填充到界面代码....."处添加如下代码:

YahooWeather.parseWeather(this,cityCode);

添加完上述代码后,再次编译 YahooWeatherFrame 类,并运行 YahooWeatherFramePre 类。此时如图 1、图 2 所示,程序就能够从网络中获取对应城市的天气预报 信息并显示在界面上了。

## 6 程序的打包与发布

完成了所有代码的编写后,就可以将程序打包发布 了,在本案例中使用 JavaSE 6.0 自带的 jar 工具将应用 程序打包成可执行 jar 包发布,按如下步骤操作即可。

首先,在进行打包之前首先要编写一个名称为 MANIFEST.MF的清单文件,用来对应用程序进行描述, 其内容如下:

Mainfest-Version: 1.0

Main-Class: wyf.YahooWeatherFramePre

Created-By: 1.6.0(Sun Microsystems Inc.)

请读者特别注意的是,这个文件的格式非常重要,

每一行要顶格写,前面不能留空格。同时,":"前面不能有空格,后面一定要留一个空格,大小写也不能出错。

其次,将 MANIFEST.MF 文件保存到编译后 class 文件包所在的目录,在命令行窗口执行如下命令,执行 结果如图 4 所示。

path=C:\Program Files\Java\jdk1.6.0\bin jar cvfm YWC.jar mainfest.mf wyf/\*

| CA C:\VIHDOVS\system32\cmd.exe   | <u>- 0 ×</u>          |
|--|-----------------------|
| C:\software)jar cufm YWC.jar mainfest.mf wyf/*<br>标明清单(manifest)   | <u> </u>              |
| 増加: wyf/YahooWeather.class(決入= 7491)(写出= 3904)(压缩了 47%)<br>増加: wyf/YahooWeatherFranc.class(決入= 4087)(写出= 2008)(压缩了 48%<br>増加: wyf/YahooWeatherFrancFroit.class(決入= 492)(写出= 548)(圧缩了<br>増加: wyf/YahooWeatherFrancFreitclitylten.class(決入= 513)(写出= 320)( | )<br>11×)<br>压缩了 37×) |
| 增加: wyf/YahooWeatherFramePre.class(读入= 3398) (写出= 1938)(压缩了:   | 42:>                  |

图 4 制作可执行 jar 包的过程

请读者注意,笔者机器上 JavaSE 6.0 的安装目录为 "C:Java\jdk1.6.0",读者需要根据自己机器的情况对路 径进行相应修改。

第三, jar 包制作完毕后,可以通过运行如下命令启 Yahoo 天气预报客户端应用程序。

path=C:\Program Files\Java\jdk1.6.0\bin

java -jar YWC.jar

由于本应用程序使用了资源图片,因此在运行可 执行 jar 包时需要将 jar 包与 img 文件夹放在同一个目 录下。另外,如果读者机器上的 JavaSE 安装完全正确 的话,也可以通过鼠标双击 YWC.jar 文件启动此应用 程序。

## 7 结语

通过开发 Yahoo 天气预报客户端程序,读者应该对 JavaSE 中提供的 JAXP DOM 有了一定的了解。同时, 读者也应该了解到使用 Yahoo 提供的天气预报信息服务 来丰富自己软件的功能也是十分方便的。另外,本案例 虽然是桌面版的,但获取天气信息并解析的 YahooWeather 类是独立开发的。如果有需要,读者只要 稍加修改就可以将其与 Web 应用程序结合使用,让自己 的 Web 应用轻松拥有天气预报的功能。

## 4.5 多媒体网络集中管理

#### 王海峰

随着高校信息化建设的不断推进,多媒体网络教学 已经在全国各地的各类高校及单位得到广泛的应用,面 对着学校众多分布在不同校区和楼宇内的多媒体教室, 其管理与维护问题已经突显出来。

本管理系统分服务器管理系统与客户端管理系统, 都是以 VB 6.0 编程实现,由于整个系统模块比较多, 不能全部叙述,现就系统中的关键技术:远程唤醒(即 远程启动计算机)、远程通信、远程关机、三个模块进 行详细阐述。

## 1 远程唤醒模块的设计与实现

该模块主要是实现对远程计算机的集中唤醒,对计

算机进行网络远程控制。唤醒模块在服务器当中,其主 要功能就是要能够实现计算机远程启动,当需要通过网 络唤醒一台指定的远程计算机时,可以从本地计算机向 网络中发送一个基于 MAGIC PACKET 标准的唤醒数据 包,该数据包中含有欲被唤醒计算机所带网卡的物理地 址,此时,欲被唤醒的计算机虽然处于关机状态,但其 网卡所带的控制芯片是由计算机电源的专用线路供电 的,因此仍然可以接受和处理网络上的数据包。当控制 芯片通过检查数据包内的 MAC 地址,发现并且确认数 据是发给自己的时候,就通过专用线路发出开启电源信 号,通知主板开机启动。

唤醒数据包的用户数据分为前后两部分,前面部分 是 6 个 FF,用来表示这是一个特殊的唤醒数据包。后

214

面部分是被重复了 16 次的网卡的 MAC 地址,用于指定 欲唤醒的远程计算机唤醒数据包数据格式如表1所示, 如假设要唤醒的远程主机所带网卡的 MAC 地址为 00-10-B5-68-8D-75, 其对应的远程唤醒数据帧为:

FFFFFFFFFF6010B5688D750010B5688D750010

D750010B5688D750010B5688D750010B5688D750010B 5688D750010B5688D750010B5688D75 远程唤醒部分用 VB.NET 编写。 圭 1

B5688D750010B5688D750010B5688D750010B5688D75

0010B5688D750010B5688D750010B5688D750010B5688

| 目的地卡号 | 来源地卡号 | FFFFFFFFFF | 重复十六次的远程计算机 MAC | 其他数据 |  |  |
|-------|-------|------------|-----------------|------|--|--|

控制台应用程序其实就是在 DOS 下运行的可执行文 件,编写 wol.exe 唤醒程序以便在其他程序中调用。在 Visual Studio VB.NET 环境中使用命令行参数, Command() 函数读取要唤醒的计算机 MAC, Command()返回一个字 符串,其中包含在命令行上位于程序名称之后输入的所 有文本。为了添加参数,须按照下面几个步骤:

首先,在"项目"中选中 WOL 工程以打开工程的 属性页面,在属性页面的左边,选择 Configuration Properties下的 Debugging,如图1所示。



图1 WOL 工程的属性页面设置

其次,在标题为命令行参数的文本框中输入(data) 一个或者使用空格分隔的多个参数,注意如果参数中包 含有空格的话则添加双引号。

最后,单击"确定"按钮,在运行该工程时,指定 的参数将被传送到程序中,就像在命令行提示符处输入 一样。

```
下面则是编好的 wol.exe 控制台可执行程序:
Imports System.Net.Sockets '引入 Socket 类
Imports System.Net '引入网络.NET 类
Module Module1
Dim n As Integer '定义 n 为重复次数变量(整型)
Dim i As Integer '定义 i 为 MAC 个数变量(整型)
Dim data As String '定义 data 为字符串
Dim str(5) As String
Dim myudp As UdpClient '定义 myudp 为 UdpClient 类
Dim intport As Integer '定义端口变量 intport 为整型
Dim wol(101) As Byte '定义 wol 为重复 16 次的字节型数组
Dim groupip As IPEndPoint
'定义 groupip 为发送唤醒数据终端 IPEndPoint 型
```

Sub Main()

data = Command() '接收 DOS 控制台输入的 MAC 地址 intport = 3600 '唤醒数据端口为 3600 'try...catch 错误处理机制 trv For i = 0 To 5 str(i) = Mid(data, 2\*i+1, 2) '取 MAC 地址中两个为一组 str(i) = "&H" + str(i) '在选取字符串前添加"&H" wol(i) = CByte(&HFF) '转换6个FF为字节型 Next For n = 1 To 16 '设置重复次数 n 为 16 次 i = n \* 6wol(i) = CByte(str(0)) '转换 str 数组六个字符串为字节型 wol(i + 1) = CByte(str(1))wol(i + 2) = CByte(str(2))wol(i + 3) = CByte(str(3))wol(i + 4) = CByte(str(4))wol(i + 5) = CByte(str(5))Next groupip = New IPEndPoint(IPAddress.Parse ("255.255.255.255"), intport) '设置 groupip 为网络广播地址 255.255.255.255 端口 3600 myudp = New UdpClient '重载 myudp myudp.Send(wol, 102, groupip) '以 UDP 协议向网络中发送要唤醒的 MAC 物理地址 myudp.Close() '发送完毕关闭 myudp MsgBox("MAC 地址当中有错误") End Try End Sub End Module 在工程中调用 wol.exe 的时候后面要空一格,譬如

在 VB 中调用这个控制台程序的时候,要远程启动的电 脑 MAC 地址为 00-10-B5-68-8D-75, 那么在调用的时候 为这样 shell(c:\wol.exe 0010B5688D75)大小写都不要 紧。执行这个语句后就能把远程电脑给唤醒(要求计算 机的 CMOS 设置中的"Power Up On LAN"或者"Power Up On PCI Card" 必须被设置"Enable")。

2 客户端与服务器通信

服务器要控制多个客户端,如实施远程关机、发送 命令等,必须通过网络通信。设计中用 VB 6.0 中的 Winsock 控件来实现网络通信。服务器要可靠的与客户 端通信,同时要在服务器管理软件中实时地显示连接状 态,只有确保连接后才能够实现网络控制,所以采用 TCP/IP 协议建立网络连接。

计算机通信端口从 1~65535,设计中通信端口从

1000 开始,服务器一个端口对一个远程计算机进行 TCP/IP 通信,例如计算机 1000 申请向服务器通信端口 1000 申请 TCP/IP 连接,服务器设计的每个通信端口都 处于侦听状态,只要有远程计算机申请就连接,其连接 示意图如图 2 所示。





```
服务器端侦听代码:
                          '设置服务器监听端口
   sckserver(0).LocalPort = 1000
   sckserver(0).Listen
                          '启动监听
   服务器端接收连接代码:
   Private Sub sckserver_ConnectionRequest(Index As Integer,
ByVal requestID As Long)
   If sckserver(Index).state > 0 Then
      sckserver(Index).Close
      '如申请连接的服务器端口没关,则先关闭此端口
   End If
   sckserver(Index).Accept requested
   '对应服务器端口接收客户端申请连接
   sckserver(Index).SendData "你已经与服务器建立连接!"
   '向客户端回发已连接成功的信号
   End Sub
   客户端申请连接代码:
   sckclient.RemoteHost = remoteIP
   '设置客户端的 IP 地址或计算机名
   sckclient RemotePort = 1000
   '设置需要连接的远程服务器端口
                       '连接远程服务器
   sckclient.Connect
   发送数据采用 Winsock 的 SendData 方法,通过上
述代码可以进行网络通信。
```

3 远程关机

## 3.1 远程关机的设计思想

在 TCP/IP 网络协议的 C/S 结构中,在一个计算机 上装上服务器端的软件(控制方通过端口给受控方的端 口发送数据包),在受控的另外的计算机上装上客户端 的软件(用于侦听端口并接收数据包),通过 TCP/IP 网 络协议来进行网络通信。建立网络连接后,如果要关闭 客户端计算机,只要向客户端发送"关机"命令,当客 户端收到该命令后调用系统中的关机函数,就可完成关 机操作。假设需要控制的计算机的 IP 地址为 192.168.45.68.服务器端的主要程序代码如下: remoteIP = "192.168.45.68" '设置需要连接的 IP 地址 sckclient.RemoteHost = remoteIP sckclient.RemotePort = 1001 '设置需要连接的远程端口 sckclient.Connect '连接远程计算机 连接上之后,如果要求关闭远程计算机,发送"关 机"字符串命令: Private Sub sckserver Connect() sckserver.SendData "关机" End Sub 客户端侦听服务器连接请求,只要连接上,客户端 根据服务器的指令来作出操作,其主要程序代码与步骤 如下: Private Sub sckserver ConnectionRequest (ByVal requestID As Long) sckserver. Accept requested '服务器首先接收对应客户端申请连接的请求 End Sub 彼此建立 TCP/IP 连接成功后服务器发送"关机" 命令。 Private Sub sckserver DataArrival(ByVal bytesTotal As Long) Dim remotedata As String sckserver.GetData remotedata '接收关机命令 If Trim(remotedata) = "关机" Then

```
Call closeclient() '调用关机程序
End If
接下来讲解客户端关机程序 closeclient()。
```

#### 3.2 客户端关闭操作系统

目前计算机系统多为 Windows 2000 和 Windows XP。Windows 98 系统下的关机比较容易,只需要直接 调用关闭计算机函数 ExitWindowEx 就可以了,而在 Windows 2000 系统下的调用这个函数只能"注销"而不

能实现系统关机。在 Windows 2000 系统实现关机操作 需得到 Shutdown 许可,得到 Shutdown 许可需要调用相 应的 API 函数,具体过程如下:

①首先得到当前程序的进程句柄;

②使用这个进程句柄得到当前程序的访问标记;

③生成 TOKEN\_PRIVILEGES 结构;

④有了 TOKEN PRIVILEGES 结构后再获得 Shutdown 权限:

⑤最后使用 ExitWindowEx 函数进行相关操作。

3.2.1 相关的 API 函数

1)获得当前进程伪句柄函数 GetCurrentProcess,函 数声明如下:

Private Declare Function GetCurrentProcess Lib "kernel32" () As Long

2) 取得程序访问标记函数 OpenProcessToken, 函 数声明如下:

Private Declare Function OpenProcessToken Lib "advapi32" (ByVal ProcessHandle As Long, ByVal DesiredAccess As Long, TokenHandle As Long) As Long

3) 查找进程权限函数 LookupPrivilegeValue, 函数 声明如下:

Private Declare Function LookupPrivilegeValue Lib "advapi32" Alias "LookupPrivilegeValueA" (ByVal lpSystemName As String, ByVal lpName As String, lpLuid As LUID) As Long

4) 获得关机权限函数 AdjustTokenPrivileges, 函数 声明如下:

Private Declare Function AdjustTokenPrivileges Lib "advapi32" (ByVal TokenHandle As Long, ByVal DisableAllPrivileges As Long, NewState As TOKEN\_PRIVILEGES, ByVal BufferLength As Long, PreviousState As TOKEN PRIVILEGES, ReturnLength As Long) As Long

5) 关机函数 ExitWindowEx。用于关机操作的一些 常数如表2所示。

| 圭  | 2 |
|----|---|
| ৰহ | 2 |

| 常数值             | 字符标识                    | 作用          |
|-----------------|-------------------------|-------------|
| &H20            | TOKEN_ADJUST_PRIVILEGES | 关机权限标识      |
| <b>&amp;</b> H8 | TOKEN_QUERY             | 标识查询        |
| &H2             | SE_PRIVILEGE_ENABLED    | 私有进程有效      |
| 0               | EWX_LOGOFF              | 终止进程系统注销    |
| 1               | EWS_SHUTDOWN            | 关闭系统电源      |
| 2               | EWS_REBOOT              | 重新启动        |
| 4               | EWS_FORCE               | 强行终止没有响应的进程 |

#### 3.2.2 关闭计算机的程序

代码如下: Private Sub closeclient() Dim hProcessHandle As Long Dim hTokenHandle As Long Dim tmpLuid As LUID Dim tkpNew As TOKEN\_PRIVILEGES Dim tkpPrevious As TOKEN\_PRIVILEGES Dim lBufferNeeded As Long '获取当前进程的一个伪句柄 hProcessHandle = GetCurrentProcess() OpenProcessToken hProcessHandle, TOKEN ADJUST PRIVILEGES Or TOKEN QUERY, hTokenHandle LookupPrivilegeValue "", "SeShutdownPrivilege", tmpLuid' 获取 LUID 关闭权限 tkpNew.PrivilegeCount = 1 '设置权限 tkpNew.TheLuid = tmpLuid tkpNew.Attributes = SE PRIVILEGE ENABLED lBufferNeeded = 0 '使关闭程序标识有效 '允许当前应用程序有关闭操作系统的权限 AdjustTokenPrivileges hTokenHandle, False, tkpNew,

Len(tkpPrevious), tkpPrevious, lBufferNeeded ExitWindowsEx EWX FORCE Or EWX POWEROFF, 0 End Sub

将上述关机代码写入每个需要控制的客户端计算 机当中, 就可以关闭计算机了。图 3、图 4 分别为多媒 体网络集中管理系统的服务器端与客户端主界面。

| : 若栏<br>教 | 室 | MAC物理地址       | IP地址             | 使用频率 | 连接状态       | 服务器 | 上课时间  | <u></u> ۲ |
|-----------|---|---------------|------------------|------|------------|-----|-------|-----------|
|           | ± | 00115BC5ADC1  | 192.168.1.13     | 40   | 没有连接或连接失败! | 0   | 09:30 | 11        |
| 2         | ± | 001111908462  | 192. 168. 1. 15  | 7    | 没有连接或连接失败! | 1   | 08:30 | 10        |
| 2         | 3 | 000A4814F02F  | 192. 168. 1. 20  | 0    | 没有连接或连接失败! | 2   | 17:00 | 09        |
| 5         | 3 | 000A4814E1B2  | 192. 168. 1. 57  | 0    | 没有连接或连接失败! | 3   | 08:45 | 07        |
| 2         | 3 | OOOFEAB843BB  | 210.29.192.50    | 0    | 没有连接或连接失败! | 4   | 08:00 | 09        |
| 2         | 3 | 0003400003424 | 210. 29. 192. 35 | 29   |            | 0   | 08:00 | 09        |
|           |   |               |                  |      |            |     |       |           |

图 3 服务器端控制软件



图 4 客户端控制软件

4 结语

本文所阐述的只是多媒体网络控制系统中的几个 关键技术,唤醒程序是在 VB.NET 2003 版中实现的控制 台应用程序;关机程序是在 VB 6.0 中实现的。整个多 媒体网络集中管理系统在实际运行中效果很好、各项功 能都能实现。

## 4.6 ASP 环境下网络硬盘的设计与实现

张锋

## 1 功能

网络硬盘 NetDiskV1.0 是一款办公系统应用软件。 系统基于 B/S 架构,实现了多用户的并发访问。每个网 盘用户分配 1G 的空间,用户可在自己的硬盘空间内创 建 4 级文件夹,单个文件夹可再创建最多 10 个文件夹。 并主要实现了文件夹的切换、用户占用空间的显示、文 件夹内容的显示、文件夹的创建和删除、文件的上传、 下载和删除等功能。

## 2 设计原理

### 2.1 用户登录

网络硬盘用户登录系统时,从数据库中提取用户 名、用户 ID 号和是否为网络硬盘用户的信息,并存储 到相应的 session 变量中。

## 2.2 用户文件映射

利用 server.MapPath()映射网络硬盘空间 userDisk 文件夹在网络服务器上的位置,网络硬盘用户登录系统时

在该文件夹下创建以"用户名+用户 ID"命名的文件夹, 作为网络硬盘用户的硬盘空间(一个为该用户使用的文 件夹),相应的目录为络硬盘用户的根目录。

## 2.3 文件夹切换

网络硬盘用户进入系统和进行文件夹切换的时,更 改当前文件夹路径变量的值,并存储到 session 变量中。 每次页面调用时,当前文件夹路径在该 session 变量中 取值。

## 2.4 用户操作

利用自定义的 FSO 实现文件夹的创建、删除、使用 空间的计算等操作。

## 2.5 文件上传

利用化境无组件上传控件实现文件的上传。

## 3 界面设计

界面设计力求简洁、直观和方便操作,界面元素和 设计效果分别如表1和图1所示。

| 名称           | 类型     | 取值       | 说明                  |
|--------------|--------|----------|---------------------|
| currentfcn   | Select | 当前文件夹的路径 | 显示当前文件夹的路径,进入下一级文件夹 |
| bttnReturnUp | button | 返回上一级    | 返回上一级文件夹            |
| bttnUpload   | button | 上传文件     | 调用文件上传窗口, 实现文件上传    |

表1 界面元素列表

| 名称        | 类型     | 取值    | 说明   |
|-----------|--------|-------|--|
| bttnUpNew | button | 新建文件夹 | 调用新建文件窗口,实现文件创建                            |
|           | table  |       | 显示空间占用比例                                   |
|           | table  |       | 显示当前文件夹下的文件夹和文件列表,给出文件(夹)<br>删除链接,给出文件下载链接 |

| 网络硬盘      |             |           |          | 4  | 使用提示        |
|-----------|-------------|-----------|----------|----|-------------|
| 您现在       | 的位置:        | <b>~</b>  |          | 己用 | ]空间: ឫ      |
|           | 🤳 👪<br>:传文件 | ٤         | 新建文件夹    |    | <b>W</b>    |
| 编号 名称     | 类型          | 大小        | 文件上传时    | 时间 | 是否劃除        |
| y 🛅 🕸 (🕸) | 文件夹         | <b>W</b>  |          |    | <b>10</b>   |
| u 🔀 🙀     | <b>W</b>    | <b>19</b> | <b>W</b> |    | <b>1</b> 33 |
| -         |             |           |          |    |             |
|           |             |           |          |    |             |
|           |             |           |          |    |             |
|           |             |           |          |    |             |
|           |             |           |          |    |             |

## 图 1 系统设计界面

## 4 系统页面文件和函数

下面列出系统实现时用到的主要页面文件。主要页

面文件包括 index.asp, default.asp, login.asp, logout.asp, fileOperater.asp, persionDisk.asp, createNewFileClip.asp, delfile.asp, delfileClip.asp, upfile.asp。文件说明如表 2 所示。

| 页面名称                  | 说明                                     |
|-----------------------|--|
| index.asp             | 系统主页,利用 iframe 内嵌 persionDisk.asp 页面   |
| default.asp           | 系统登录窗口                                 |
| login.asp             | 用户名是否为空检验,强制用户登录后才能访问 persionDisk 页面   |
| logout.asp            | 系统退出窗口,清空 Session 变量                   |
| fileOperater.asp      | 文件操作文件,包括西楼冷月的 FSO 操作类 FsoCls 和自己定义的函数 |
| persionDisk.asp       | NetDiskV1.0 主体逻辑页面,包括主要的功能模块           |
| createNewFileClip.asp | 创建文件夹                                  |
| delfile.asp           | 删除文件                                   |
| delfileClip.asp       | 删除文件夹                                  |
| upfile.asp            | 化境无组件文件上传控件                            |

在对文件进行操作时用到了FSO操作类FsoCls,文中也写了一些文件操作的函数把它们一并写在fileOperater.asp页面中。限于篇幅,下面给出本系统调

用的 FsoCls 的属性和函数的说明,分别如表 3 和表 4 所示。

219

| 属性名称  | 参数   | 功能说明                                |
|---|--|-------------------------------------|
| IsFileExists(ByVal FileDir)                                 | 参数 FileDir 为文件的绝对路径                                  | 判断文件是否存在,存在则 返回 True,否则返回 False     |
| DelFile(ByVal FileDir)                                      | 参数 FileDir 为文件的绝对路径                                  | 删除一个文件,成功返回<br>True,否则返回False       |
| IsFolderExists(ByVal FolderDir)                             | 参数 FolderDir 为文件的绝对路径                                | 判断文件夹是否存在,存在<br>则返回 True,否则返回 False |
| FileItem(ByVal FolderDir)                                   | FolderDir 为文件夹绝对路径                                   | 文件夹里的文件集合                           |
| CreateFolderA(ByVal ParentFolderDir,<br>ByVal NewFoldeName) | ParentFolderDir 为父文件夹的绝对路径, NewFolderName 为要新建的文件夹名称 | 在某个特定的文件夹里创建<br>一个文件夹               |

#### 表 3 FSO 对象调用属性说明表

#### 表4 自定义函数说明表

| 属性名称   | 参数  | 功能说明                        |
|--|---|-----------------------------|
| getCurrentFileClipPath(ByVal<br>parentFilePath,ByVal currentfcn) | 参数 parentFilePath 为父文件的绝对路径, currentfcn 为文件夹的名称 | 返回父文件夹下某个文件夹的绝<br>对路径       |
| returnUpClass(ByVal<br>currentFilePath)                          | 参数 currentFilePath 为文件的绝对路径                     | 返回当前目录的上一级目录                |
| returnFileExt(ByVal FileName)                                    | 参数 FileName 为文件名                                | 返回某个文件的类型                   |
| returnFileClipNum(Byval<br>thiscurrentfcp)                       | thiscurrentfcp 为当前文件夹绝对路径                       | 返回当前文件夹目录下子文件夹<br>中文件和文件夹个数 |
| eturnFileClipSize(Byval<br>thiscurrentfcp)                       | thiscurrentfcp 为当前文件夹绝对路径                       | 递归函数返回文件夹的大小                |
| returnGuiFan(ByVal FileName)                                     | FileName 为文件名                                   | 处理文件名过长,显示 8 个字符,<br>其余显示"" |
| showSize (ByVal Space)   | Space 数值,字节量                                    | 显示磁盘空间                      |

## 5 实现方案

系 统 主 要 的 逻 辑 结 构 和 主 要 功 能 模 块 在 persionDisk.asp 实现。主要功能模块包括: 创建网络硬 盘用户个人文件夹和存取当前文件夹路径模块、文件夹 操作及切换模块、空间占用信息显示模块、文件夹详细 信息显示模块、文件夹创建模块、文件夹删除模块等。

#### 5.1 主体逻辑结构

```
<!--#include file="login.asp"-->
<!--#include file="Connections/conn.asp"-->
<!--#include file="fileOperater.asp"-->
<%
if session("user_disk")=1 then
%>
<!--如果是网络硬盘用户,进行相应的网络硬盘的创建和
操作-->
<%
Else if session("user_disk")=0 then
%>
<!--如果不是网络硬盘用户给出信息-->
```

<%

response.write("对不起! 你未授权使用网络硬盘") end if

%>

用户登录系统时,获取该用户是否是网络硬盘用户 的信息,并存储在 session("user\_disk")变量中。用户成 功登录,导航到 persionDisk.asp 页面。跟据 session ("user\_disk")判断是否是网络硬盘用户,如果是,进行相 应的网络硬盘的创建和操作;如果不是给出相应的提示 信息。

注意,下面的模块都是在当前用户是网络硬盘用户 的前提下调用的。

## 5.2 创建个人文件夹和当前路径

该模块主要用来设置当前文件夹的路径。在应用程 序文件夹下已经创建 userdisk 文件夹,本模块在 userdisk 文件夹创建以网络硬盘用户的"用户名+注册 ID"命名 的文件夹作为网络硬盘用户自己的文件夹,相应的该文 件的路径就是网络硬盘用户的根目录。

判断个人文件夹是否存在,如果不存在就创建,如

果存在就在该文件夹下进行相应的文件(夹)的操作。 网络硬盘用户进入系统在个人文件夹下操作,当前 文件夹路径为网络硬盘用户的根目录,进行文件夹之间 的切换时,改变当前文件夹的路径。把当前文件夹路径 的值写到 session 变量中,每次页面调用时,当前文件 夹路径在该 session 变量中取值,代码如下: <% dim sfp,fileFoiderName,bfileFlipExsit,userfp 'sfp 用户文件夹路径, string 类型 'fileFoiderName 文件夹的名称, string 类型 'bfileFlipExsit 标识变量,指示某个路径的文件(夹)是 否存在, bool 类型 'userfp 网络硬盘用户的根目录 sfp=server.MapPath("userDisk") fileFoiderName=session("user\_truename")&session("user\_id") userfp=sfp&"\"&fileFoiderName dim objFileOperater set objFileOperater=new FsoCls bfileFlipExsit=objFileOperater. IsFolderExists(userfp)if bfileFlipExsit then dim currentfcp'当前文件夹路径 dim curentfcn'当前文件夹的名称 dim tagDownOrUp'标志变量,指示进入下一级还是返回 '上一级 curentfcn=trim(request("fileclipName")) response.Write("curentfcn="&curentfcn) tagDownOrUp=trim(request("tagupordawnClass")) txtFileClipName=trim(request("txtFileClipName")) if session("currentfcpSession") <> "" then else session("currentfcpSession")=userfp end if currentfcp=session("currentfcpSession") '如果是在当前文件夹下操作,不改变当前文件夹路径, '不用更新 session("currentfcpSession")变量 '如果是进行文件夹切换,需要改变当前文件夹路径的 '值,并更新到 session("currentfcpSession")变量 if tagDownOrUp="down" then currentfcp=getCurrentFileClipPath (currentfcp,curentfcn) end if if tagDownOrUp="up" then currentfcp=returnUpClass(currentfcp) end if session("currentfcpSession")=currentfcp else call objFileOperater.CreateFolderA(sfp,fileFoiderName) response.Redirect("PersonalDisk.asp")'创建结束后,回到 '个人网盘界面 end if end if %> 如果是网络硬盘用户,首先映射应用程序的文件夹 "userdisk"在网络服务器上的位置,然后在该文件夹下 创建以"用户名+用户 ID"命名的文件夹,作为该网络 硬盘用户的个人文件夹,相应的该文件夹的目录是网络

硬盘用户的根目录。

网络硬盘用户进入系统在个人文件夹下操作时, session("currentfcpSession")为空值,当前文件夹的路径 为个人文件夹的根目录 userfp,通过判断持久存放当前 文件夹路径的变量 session("currentfcpSession")是否为空 值来判断当前文件夹路径是否是在网盘用户的根目录 下。如果网盘用户在自己的根目录下操作,就初始化 session("currentfcpSession")为 userfp 变量; 然后设置 currentfcp 变量为 session("currentfcpSession")的值。

在当前文件夹下的操作分为两类:一类是文件夹切 换,如进入下一级文件夹和返回上一级文件夹;另一类 是文件操作,如文件的上传、文件夹的创建、文件(夹) 的删除等操作。文件夹切换操作,改变当前文件夹路径 的值 currentfcp,并更新 session("currentfcpSession")变量 的值;文件操作,不改变当前文件夹路径变量的值,不 用更新更新 session("currentfcpSession")变量的值。

### 5.3 文件夹操作及切换模块

利用下拉列表 (select) 控件显示当前文件夹路径的 相对个人文件夹的相对路径和实现到下一级文件夹下 文件夹的切换;利用按钮 (botton) 控件实现向上一级 文件夹切换,代码如下:

```
<select name="curentfcn" id="curentfcn"
                                               onChange=
"return valueUrl()" >
    <%
      dim currentPosition,currentfcptem
      currentfcptem=split(currentfcp,"\")
      currentPosition=""
      '从网络硬盘的根目录算起有盘符、应用程序文件夹、
      'userdisk 文件夹,所以用户操作的文件夹应该从第四层
      '开始,变量i取值为4
      '如果是根目录显示当前位置为"\\"
      if ubound(currentfcptem)=3 then
      currentPosition="\\"
      end if
      if ubound(currentfcptem)>=4 then
      for i= 4 to ubound(currentfcptem)
        currentPosition=currentPosition&"\"&currentfcptem(i)
      next
      end if
    %>
        <option value="<%=currentPosition%>">
        <%=currentPosition%></option>
    <%
    dim fileFolderList,ifoulderCount
    fileFolderList=split(objFileOperater.FolderItem(currentfcp),"|")
    ifoulderCount=fileFolderList(0)
    '显示子文件夹的名称
    for i=1 to ubound(fileFolderList)
    %>
       <option value="<%=fileFolderList(i)%>">
       <%=fileFolderList(i)%></option>
    <%
    next
```

| %≥ |  |
|----|--|
|    |  |
|    |  |

<input name="bttnReturnUp" type="button"
style="font-size: 10pt" onClick="return returnUpclass()"
value="返回到上级目录"
<%if currentfcp=userfp then %> disabled="true"<%end if%>>
客户端脚本,用 JavaScript 语言实现:
function valueUrl(){
var fileName="PersonalDisk.asp";
var fileclipName=curentfcn.value;
location.href=fileName+"?fileclipName="+fileclipName+"&
tagupordawnClass=down";
}

function Upclass() {
var fileName="PersonalDisk.asp";
var fileclipName=curentfcn.value;
location.href=fileName+"?tagupordawnClass=up";
}

在文件夹切换部分,首先设置下拉列表默认选项的 值和显示。通过分割 currentfcp 变量构造相对于网络硬 盘用户文件夹的路径 currentPosition 变量的值。如果在 网盘用户的根目录下,设置该值为"\\",设置下拉列表 的默认选项的值为 currentPosition,显示内容为 currentPosition。其次处理其他选项的值和显示。获取当 前路径下的子文件夹的数目和名称,依次显示出来。这 样在文件夹切换时,调用客户端脚本 valueUrl(),传递 传递文件夹名称和 tagupordawnClass 变量为 "down"。

在处理返回上一级部分时,判断当前文件夹路径是 否在网络硬盘用户的根目录。如果是,就把该按钮灰掉; 如果不是,该按钮可用,点击时,简单调用客户端脚本, 传递文件夹名称和 tagupordawnClass 变量为"up"。

客户端脚本,主要用来页面的传值和回调。

#### 5.4 空间占用信息显示模块

指定每个网盘用户的空间为 1GB, 查找每个网盘用 户的根目录, 获取该用户的使用空间值, 把该值与 1073741824B(1GB)作比较得到以用户占用空间的比 例;利用一个单元格来显示占用空间的比例。在显示占 用空间比例时,进行单位转换。文件小于 1MB 时显示 多少 KB, 大于 1MB 时显示多少 MB, 实现代码如下: <td width="164" height="20" align="center" bgcolor=

```
"#CCFFCC">已用空间: <%
```

'计算网盘用户占用的空间,每个用户指定1GB的大小 dim contain,containbfb,containtiao,MySpace,Surpluscapacity MySpace=returnFileClipSize(userfp) '个人空间目前总使用量

Surpluscapacity=1073741824- MySpace'存储剩余空间

session("Surcapacity")= Surpluscapacity contain=MySpace/1073741824'以 1GB 容量为参考 containbfb=contain\*100'把容量百分数转换成百分比 containtiao=contain\*164'164 为比例表格 Table 的宽度 if containtiao<1 and containtiao>0 then

```
'特殊处理容量显示
  containtiao=1
  end if
  '正确显示文件大小,以 M 为单位,具体在自定义函数
  'sizeshow()实现
Call sizeshow(MySpace)
 %>
<div id="shiyongliang" align="center">
  <%
 if containbfb=0 or Round(containbfb,5)=0 then
 '考虑上传文件过小
   response.Write("0%")
 elseif containbfb<1 then
  response.Write("0"&Round(containbfb,5)&"%")
  '保留五位小数
 else
  response.Write(Round(containbfb,2)&"%")
  '保留两位小数
 end if
%>
 </div>
<!--显利用单元格相对宽度显示空间比例信息-->
    <%
  if containtiao <>0 then%>
    " bgcolor=
    "#00FFCC">
    <%end if%>
    <td>
    首先设定每个网盘用户的空间为 1GB, 通过调用
```

returnFileClipSize()函数得到网盘用户占用的空间 MySpace。把MySpace与1073741824B容量大小做比得 到占用的比例值 contain。接着设定 containbfb= contain\*100,把容量百分数转换成百分比, containtiao= contain\*164得到单元格的宽度比例。

在显示占用空间时进行单位换算。如果空间小于 1MB显示多少 KB,如果大于 1MB显示多少 MB。精确 到两位有效数字。在显示占用空间百分比 containbfb 时, 如果 containbfb 为零或足够小就显示 "0%",小于 1%显 示五位有效数字,大于 1%显示两位有效数字。

最后通过指定一个单元格在表格中的相对宽度比例来显示占用空间的比情况,即: " bgcolor="#00FFCC">

#### 5.5 文件夹详细信息显示模块

在表格中显示当前目录下的文件夹和文件。显示项 为编号、名称、类型、大小、文件上传时间、是否删除。 在显示文件夹和文件时根据编号改变每一行的颜色。显 示文件夹时,名称项显示当前目录下的子文件夹的名称 以及该子文件夹下文件(夹)的个数,类型项显示"文 件夹",是否删除项显示是否删除该文件夹的链接。大 小和文件上传时间项显示为空格。 在显示文件时,名称项显示文件的名称,类型显示 文件的类型,大小显示文件的大小,文件上传时间项显 示长时间格式的时间值,是否删除项显示是否删除该文 件的链接, 实现代码如下: <table width="614" border="0" align="center" cellpadding= "0" cellspacing="0"> <!--DWLayoutTable--> <strong>编号</strong> <strong>名称</strong> <strong>类型</strong> <strong>大小</strong> <strong>文件上传时间 </strong> <strong>是否删除 </strong> <!--文件夹列表--> <% '获取当前目录下文件夹的数组 dim fileClipArray,thiscurrentfcp,NumberClipOrFile fileClipArray=split(objFileOperater.FolderItem (currentfcp),"|") NumberClipOrFile=ubound(fileClipArray) '记下文件夹的个数 for k=1 to NumberClipOrFile thiscurrentfcp=currentfcp&"\"&fileClipArray(k) '子文件夹路径 fileClipArray(k)=returnGuiFan(fileClipArray(k)) %> <!--根据文件编号奇偶性设定表格行的颜色--> <%if (k mod 2)=1 then%> <tr bgcolor="#f7f7f7" align="center" ><%end if%> <%if (k mod 2)=0 then%><tr bgcolor="#FFFFCC" align="center"><%end if%> <%=k%> <img src="images/ file.gif" width="16" height="16"><%=fileClipArray (k)%>(<%=returnFileClipNum(thiscurrentfcp) %>) 文件夹 <% dim FileClipSize FileClipSize=returnFileClipSize(thiscurrentfcp) call sizeshow(FileClipSize) %> <!--DWLayoutEmptyCell-->&nbsp; <%response.write "<a href='dellFileClip.asp? flieClipNamePath="&thiscurrentfcp&" 'onclick= 'return ConfirmClipDel();'>删除</a>&nbsp;" %> 

<% next %> <!--文件列表--> <% '构造当前目录相对应用程序的路径,为文件下载提 '供路径 dim currentfcpitem,fcpToApp currentfcpitem=split(currentfcp,"\") fcpToApp="userDisk" for i=3 to ubound(currentfcpitem) fcpToApp=fcpToApp&"\"&currentfcpitem(i) next '通过FileSystemObject对象 fso 获取当前目录下的文件 set fso=CreateObject("Scripting.FileSystemObject") dim FileExt,FileLianJie if fso.FolderExists(currentfcp)then Set theFolder=fso.GetFolder(currentfcp) '获取当前目录下的文件的名称、相对应用程序的路 '径、文件类型;病将相应的信息显示在表格中 For Each theFile In theFolder.Files FileLianJie=fcpToApp&"\"&theFile.Name '文件相对应用程序的路径 NumberClipOrFile=NumberClipOrFile+1 '依次列出上传文件的编号 FileExt=trim(theFile.Name) NewFileName=returnNewFileName(FileExt) FileExt=returnFileExt(FileExt) '调用函数得到用户上传文件的类型 %> <!--根据文件编号奇偶性设定表格行的颜色--> <%if (NumberClipOrFile mod 2)=1 then%> <%end if%> <%if (NumberClipOrFile mod 2)=0 then%><tr bgcolor="#FFFFCC" align="center"><%end if%> <%= NumberClipOrFile%> <!--给出对应文件类型的图标--> <img src=images/<%=FileExt%>.gif border=0> <a href="<%=FileLianJie%>" target=" blank" title="<%=theFile.Name%>"><%=NewFileName%></a> <%= FileExt%> <% call sizeshow(theFile.size) %> <%= theFile.DateLastModified%> <%response.write "<a href='delfile.asp?FileName= "&currentfcp&"\"&theFile.Name&" ' onclick='return ConfirmfileDel();'>删除</a>&nbsp;" %> 在显示当前目录下的文件夹列表时,通过调用

objFileOperater.FolderItem()获得当前目录下的文件夹的

字符串,然后用 split()函数进行分割得到当前目录下的 文件夹集合并存到一个数组变量 fileClipArray 中。对每 一个子文件夹生成了该子文件夹的路径。在显示文件编 号时,显示了文件夹数组的下标,显示文件夹名称时, 显示文件夹数组的第 k 个元素 fileClipArray(k),并通过 调用 returnFileClipNum()显示该子文件夹下的文件夹的 数目。显示文件类型项时直接显示"文件夹"。显示大 小项时,调用 sizeShow()函数对文件的大小进行的单位 转换。在显示是否删除项时,显示文件夹是否删除该文 件夹的链接。

在显示文件列表时,通过 FileSystemObject 对象 fso 获取当前目录下的文件,对每个文件,生成了文件相对 应用程序的路径、文件的编号和类型。由于要在文件列 表中提供文件的下载,要构造每个文件相对应用程序的 相 对 路 径,即: FileLianJie=fcpToApp&"\"&theFile. Name。在显示文件名称项时,给出了文件下载的路径 FileLianJie。在显示文件大小时,同样做了单位转换。 显 示 文 件 上 传 时 间 时,调用了 文 件 对 象 的 theFile.DateLastModified 属性。在显示是否删除项时,显示文件夹是否删除该文件的链接。

#### 5.6 文件上传模块

该模块使用了化境无组件上传控件。网络硬盘用户 点击上传控件按钮时,调用文件选项页面 teaupload.asp, 选定文件并点击该页面的上传按钮,调用化境无组件上 传控件进行文件类型、大小等的判断,对合乎要求的文 件上传到当前文件夹下,给出文件上传成功的信息。如 果不符合要求,给出反馈信息。本系统只是简单调用了 部分功能,有兴趣的读者可以参看相应的资料。

#### 5.7 文件夹创建模块

新建一个页面文件 createNewFileClip.asp。在该页 面上放置表单向页面提交新创建的文件夹的名称。页面 调用后,跟据文件夹名取值决定加载表单还是调用文件 操作对象 objFileOperater 进行创建。在进行文件创建时, 首先判断是否重名,如果重名给出出错信息,如果没有 重名直接创建,实现代码如下:

```
<%
dim flieClipName
flieClipName=trim(request("txtFileClipName"))
if flieClipName=""or flieClipName="请输入新文件夹名" then
%>
<!--加载表单输入文件夹的名称-->
<form name="creteNewFileClip" action=
"createNewFileClip.asp" method="Post">
cellspacing="0">
<tt>width="300" border="0" cellpadding="0"
cellspacing="0">
<tt><tt>width="104" height="50" align="center"
valign="middle">新文件夹名
```

<input name="txtFileClipName" type="text" id="txtFileClipName" onClick="if(this.value=='请输入新文件夹名') this.value="" value="请输入新文件夹名" size="20" maxlength="18"> <input name="subok" type="submit" id="subok" value="提交" style="font-size: 10pt"> </form> <% else '如果得到文件夹的名称 dim objFileOperater set objFileOperater=new FsoCls '定义 objFileOperater 为 FsoCls 的一个对象 dim established '判断用户是否输入了已经存在的文件夹名称 currentfcpSession=session("currentfcpSession") established=false if objFileOperater.IsFolderExists (currentfcpSession&"\"&flieClipName) then established=true end if if established=false then call objFileOperater.CreateFolderA (currentfcpSession,flieClipName) %> <script language="javascript"> <!-window.opener.location.href="PersonalDisk.asp? tagkk=backPost"; alert('文件夹创建成功!'); window.close() --> </script> <% else %> <script language="javascript"> <!-window.opener.location.href="PersonalDisk.asp? tagkk=backPost"; alert('该文件夹已创建! 请输入其他文件夹名称!'); window.close() \_\_> </script> <% end if end if %> 页面首先接受文本框控件 txtFileClipName 传递过

来的文件夹名称 flieClipName。如果该页面由"创建新 文件夹"按钮控件调用,接收到的 flieClipName 取值为 空,需要加载表单来让以后输入文件夹名称。如果加载 表单后没有输入有效的文件夹名称(没有改变文本框控 件的"请输入新文件夹名"就提交创建),客户端脚本 会把该控件的值置空,并提示重新加载表单。因此页面 接收到的文件夹名称是否为空或等于"请输入新文件夹 名"就要求加载表单。其他就用接收到的文件夹名称来 创建文件夹。

在进行文件夹创建时,利用了文件操作对象 objFile Operater。首先获取当前文件夹的路径 currentfcpSession =session("currentfcpSession")。调用 objFileOperater 的函 数 IsFolderExists()判断当前路径下是否有与接收到的文 件夹名称相同的文件夹,并给标志变量 established 赋值。

跟据标志变量的值决定是否创建该文件夹,并给出 反馈信息。如果没有重名的文件夹 established =false, 调用 objFileOperater.CreateFolderA()当前路径下创建一 个新文件夹,给出创建成功的信息;如果有重名的文件, 给出文件重名的信息。

#### 5.8 文件(夹)删除模块

在该模块中,首先调用客户端脚本给出是否删除信息,并调用相应页面来执行删除操作,实现代码如下:

```
(1) 客户端脚本代码
function ConfirmClipDel()/*删除文件夹*/
{
  if(confirm("确定要删除选中的文件夹吗? 删除之后将
          不能恢复"))
      return true;
  else
    return false;
}
 (2) dellFileClip.asp页面代码
<%
dim flieClipNamePath
flieClipNamePath=trim(request("flieClipNamePath"))
if flieClipNamePath="" then
response.Write("文件路径出错")
else
    dim objFileOperater
    set objFileOperater=new FsoCls
    call objFileOperater.DeleteAFolder(flieClipNamePath)
```

```
response.Redirect("PersonalDisk.asp ")
    end if
    %>
    function ConfirmfileDel()/*删除文件*/
     if(confirm("确定要删除选中的文件吗? 删除之后将不
               能恢复"))
          return true;
     else
        return false;
     (3) dellFile.asp 文件代码
    <%
    dim FileName
    FileName=trim(Request("FileName"))
    set fso=CreateObject("Scripting.FileSystemObject")
    call DelFiles()
    response.Redirect("PersonalDisk.asp")
    sub DelFiles()
     if FileName="" then exit sub
      if left(FileName,3) <> "../" and left(FileName,1) <> "/" then
          Set thisfile = fso.GetFile(FileName)
          thisfile.Delete True
     end if
    end sub
    %>
    在删除选定的文件 (夹)时,首先触发了客户端脚
本,给出是否删除相应文件(夹)消息框。用户确认后,
```

返回调用的页面和文件(夹)的路径,在调用页面中执 行具体的删除操作,并返回到网络硬盘页面。

## 6 结语

根据办公系统中网络硬盘的业务需求,在ASP环境 中给出了具体的设计和实现过程。主要使用ASP的内置 对象 Server、Session 等实现网络硬盘空间文件夹的映射 和保存当前文件夹的路径和文件夹的切换等,在文件 (夹)操作时用到 FsoCls,在文件上传时,用到无组件 上传类。目前该系统已经正式部署,运行良好。

## 4.7 在 C#中整合 Fckeditor 编辑器实现远程图片自动上传

#### 汪立欣

## 1 问题提出及解题思路

当前,网站内容管理系统均使用 HTML 编辑器来实 现网页内容排版布局功能,熟知编辑的有 Fckeditor、 cuteeditor、FreeTextbox 和 ewebeditor 等。这些编辑器以 所得即可所见的可视化编辑特点,大大地简化了网页内 容组织及排版工作,且这些编辑器各具特点,功能丰富, 笔者所负责维护的门户网站后台发布系统使用的即是 Fckeditor编辑器。在日常运维工作中,由于门户站点内 容跨站引用二级站点内容,常会因二级站点停止服务而 出现门户站点所引用的网页图片失效的问题。

如何解决这个问题呢?最初的解决方案是手工对二 级站点内容重复组织,首先将图片下载本地;随后通过 网站管理系统重新组织图文内容并将其上传到服务器 上。图1所示为跨站处理图片混编网页内容的工作流程。



图 1 手工处理跨站网页内容业务流程

但通过实践发现这种方式要求信息发布员具有较好 HTML 页面内容处理能力,同时网页内容的二次跨站处理工作非常烦琐且容易出错,如图片前后顺序以及图片宽度错误。

能否减少因手工操作出现的错误或者简化业务处 理步骤,让后台管理程序提供模块自动处理编辑器中的 远程图片呢?答案是肯定的,笔者从其他 CMS(内容 管理系统)提供的功能得到启示。针对前一个方案流程, 分析程序实现的需求:首先,需要使用正则表达式过滤 网页内容源代码从而获得内容中图片 Url 地址列表;随 后使用 XMLhttp 组件根据图片 url 列表循环获得图片数 据,同时以流的方式将图片数据上传到本地服务器上。 如图 2 所示。



图 2 程序实现自动远程处理网页内容业务流程

## 2 XMLhttp 组件和正则表达式

#### 2.1 XMLhttp 组件

XMLhttp 从字面意义上说,是传输 XML 格式数据 的超文本传输协议。这是微软提供的一种通过 http 访问 网络接受和发送 XML 数据的 API,事实上 Xmlhttp 支 持的上传和下达的指令格式还支持字符串、流以及 URL 的参数等。众所周知, XML 和 HTML 都是标记语言, 都是将内容节点化、层次化。HTML 着重数据外观显示; XML 着重数据描述体现于数据内容。严格地说 XML 更 为规范化和标准化,但 HTML 标准较 XML 标准先提出 来并得到广泛的应用,因此从某种意义上说 XML 的制 定是为了弥补 HTML 在文件处理和数据传输等方面的 不足,因此从本质上说 XMLhttp 是将 HTML 面数据序 列化为 XML 格式传输的 API。简单地讲,XMLhttp 用 来模拟 Http 的 Get 和 Post 请求,处理数据的交互操作。 目前网络上的无刷新提交数据、模拟登录、远程获取 HTML 内容等等都可以利用 XMLhttp 来实现,如常见的 小偷程序就是一个典型的例子。

- XMLhttp 的使用步骤:
- (1) 创建 XMLhttp 对象。
- (2) 初始化对象。
- (3) 远程链接服务器。
- (4) 发送指令。
- (5) 等待服务器返回处理结果。
- (6) 释放 XMLhttp 对象。

#### 2.2 正则表达式

正则表达式的本质是利用数据抽象的方法定制一 套模式去匹配一类字符串的公式。对于文本内容的处 理,正则表达式是字符串处理利器且性能卓越。有关正 则表达式的知识,由于篇幅有限,不再赘述。

通过分析远程图片 URL 地址,可以得到一个通用 的图片 URL 公式,在正则表达式,也即模式:

"http://" + 字母、数字、"/"字符组合 + "."字符 + 图片格式 (jpg、gif、jpg、png等)。

根据以上分析所得模式及正则表达式语法规则,可以书写远程图片的正则表达式:(src=)('|'|)?(http://) (.[^'\\s|']\*)(\.)(jpg|gif]png|bmp|jpeg|JPG|GIF|PNG|BMP|JP EG)('|''\\s|>)?

### 3 新建项目

### 3.1 基础工作

3.1.1 新建项目

在 Visual Studio 2005 下新建一个 C# Web 项目: AutoRomteUploadImg。

3.1.2 添加 MSXML2 引用

打开解决方案资源管理器,选择项目根目录,右键 选择"添加引用",在弹出的应用对话框中,勾选"Com" 标签下的 Microsoft XML;V3.0。

3.1.3 配置 FckEditor

(1) 复制 FckEditor 文件夹到项目根目录。

(2)将 FckEditor 控件添加到工具箱中。在左侧工 具箱空白处,右键选择"选择项",在".Net Framework 组件"选项卡中引用 FredCK.FckEditorV2.dll。

(3) 设置 FckEditor 编辑器 Base Path 属性为 "FckEditor/"。

(4) 配置用户上传路径(UserFilesPath)。

打开 web.config, 添加以下配置:

<configuration>

```
<add key="FckEditor:UserFilesPath"value="
/SiteSource/"/>
```

<appSettings>

</appSettings>

</configuration>

```
3.1.4 页面设计
```

如图3所示。



图 3 生成页面

#### 3.1.5 页面代码

<html XMLns="http://www.w3.org/1999/xhtml">

- <head runat="server">
- <title>远程远程自动上传图片</title></head>
- </neau>
- <body>
- <form id="form1" runat="server">
- <div style="margin:0 auto 0 auto; width:700px;">
- <table width="700px" cellspacing="0" cellpadding="0"

border="1" style="border-collapse:collapse;">

<asp:TextBox ID="Tb\_Title" runat="server" Width="300px"></asp:TextBox>

文章内容

<tfCKeditorV2:FCKeditor ID="FCK\_Content" runat= "server" Width="590px" Height="400px" BasePath="FCKeditor/">

- </FCKeditorV2:FCKeditor>

height:40px; line-height:40px;"> <asp:Button ID="Btn\_ Submit" runat="server" Text=" 发布文章" OnClick="Btn\_ Submit\_Click" />

<asp:CheckBox ID="CheckBox1" nunat="server" AutoPostBack= "True" Text="远程上传图片" OnCheckedChanged="CheckBox1\_ CheckedChanged" />

</div> </form> </body> </html>

3.1.6 命名空间引用

using System.IO; using System.Collections; using System.Text; using System.Text.RegularExpressions; using MSXML2;

#### 3.2 核心代码

3.2.1 远程图片上传模块

/// <summary> /// 自动远程上传图片模块 /// </summary> private void UploadRemote() ArrayList InputBoxVlaue = new ArrayList(); //获得远程图片 Url 地址列表 InputBoxVlaue = RegExpExecuteRemoteImg(FCK\_ Content.Value); //循环读取图片数据并上传 for (int i = 0; i <= InputBoxVlaue.Count - 1; i++) //读取图片数据,上传图片,返回上传图片路径。 string Newurl = ReadWriteRemoteData (InputBoxVlaue[i].ToString()); if (Newurl != "") string pattern = InputBoxVlaue[i].ToString(); //替换远程图片路径为新上传的图片路径 Regex reg = new Regex(pattern); FCK Content.Value = reg.Replace(FCK Content.Value, Newurl); } } 3.2.2 图片 URL 地址过滤模块 /// <summary> /// 分析字符串,找出远程图片路径 /// </summary> /// <param name="Str">源字符串</param> /// <returns>返回图片路径数组</returns> public ArrayList RegExpExecuteRemoteImg(string Str) { string SingleUrl = ""; ArrayList TemArryList = new ArrayList(); //定义正则表达式 string regexstr = @"(src=)('|" + System.Convert.ToChar(34) + "| )?(http://)(.[^'|" + @"\s|" + System.Convert.ToChar(34) + "]\*)" @"(\.)(jpg|gif|png|bmp|jpeg|JPG|GIF|PNG|BMP|JPEG)('|" System.Convert.ToChar(34) + "|" + @,"\s|>)?"; Regex reg = new Regex(regexstr, RegexOptions.Compiled);

MatchCollection Matchcollage = reg.Matches(Str); //匹配正则表达式 foreach (Match Matchs in Matchcollage) { SingleUrl = Matchs.Result("\$3\$4\$5\$6"); if (SingleUrl.LastIndexOf(".") != 0) TemArryList.Add(SingleUrl); } } return TemArryList; } 3.2.3 图片数据获取及图片上传模块 /// <summary> /// 远程读取单图片数据并上传到服务器 /// </summary> /// <param name="Url">图片 Url 地址</param> /// <returns>返回上传图片路径</returns> private string ReadWriteRemoteData(string Url) try //定义 XMLhttp 对象 MSXML2.XMLHTTP\_XMLhttp = new MSXML2. XMLHTTPClass(); //初始化对象 XMLhttp.open("GET", Url, false, null, null); \_XMLhttp.send(""); //定义图片上传路径 String FilePath = Server.MapPath("~/SiteSource/"); int intNumber; Random r = new Random();intNumber = Convert.ToInt32(r.Next(100, 999)); string NewFlieName = DateTime.Now.ToString("yyyy MMddHHmmss") + intNumber.ToString() + ".jpg"; //读取对象并上传 if ( XMLhttp.readyState == 4) //判断本地服务器图片是否存在 if (System.IO.File.Exists(FilePath + NewFlieName)) System.IO.File.Delete(FilePath + NewFlieName); //如存在则删除该图片定义流并初始化 System.IO.FileStream fs = new System.IO.FileStream (FilePath + NewFlieName, System.IO.FileMode.CreateNew); System.IO.BinaryWriter w = new System.IO.BinaryWriter(fs); //写入图片数据 w.Write((byte[])\_XMLhttp.responseBody); w.Close(); fs.Close(); return "/SiteSource/" + NewFlieName; } else return ""; } catch (Exception ex) ł return ""; }

## 4 结语

整合现有 HTML 编辑器实现跨站自动上传图片功 能,从信息发布者来说,进一步简化了网页内容二次组 织编辑工作,尽可能减少了因二次编辑带来的内容错误 的发生几率,同时也提高了网页内容编辑发布效率;从 维护的角度来说上,降低了信息的缺失而造成的站点内 容不完整问题。需要注意的是,该解决方案完全基于服 务器端,图片数据的处理过程涉及本地、远程服务器双 方通信,因此它必须具备一个前提:即本地服务器必须 能解析远程服务器的域名或地址。

## 4.8 QQ 聊天程序的网络通信原理及编程

蒋 智

随着计算机网络的普及,网上聊天已成了和电话一样重要且更经济的一种通信方式。QQ 是人们网上聊天的首选,在许多人的联系方式中,QQ 号是其中之一。若想自己也设计一个类似 QQ 的聊天程序,这也许是一件很难的事情,但其实并不如你所想象的复杂,可以尝试一下。

### 1 原理

QQ 聊天程序采用的是 C/S 通信模式,即客户/服务 模式, 它把一个应用划分成功能不同的两个部分, 分别 在不同的计算机上运行,其中一个为服务端程序,用来 响应和提供固定的服务,一个为客户端程序,用来向服 务端提出请求和要求某种服务。在数量关系上,通常有 一对一的(即一个服务端程序和一个客户端程序之间通 信),也有一对多的(即一个服务端程序和多个客户端程 序之间通信),也有多对多的(即多个服务端程序和多个 客户端程序之间通信)。所谓服务端程序、客户端程序也 是相对的概念,有时在一个程序中既有服务端又有客户 端的功能。QQ聊天程序分成了两个程序,一个安装在腾 讯公司的服务器上,称之为服务端程序,一个安装在QQ 用户的计算机上,称之为客户端程序。在许多介绍网络 通信编程的书籍中有关网络聊天的例子,当一个客户要 与另一个客户聊天时, 第一个客户先把聊天数据发送给 服务器,然后服务器再把聊天数据转发给第二个客户, 服务器好像一个中转站,这在客户数量比较少时,服务 器还能承受,在客户数量比较多时,服务器肯定要瘫痪。 何况客户之间还要传送文件、语音聊天、视频聊天等等, 为了减少服务器的压力,各客户端之间需要直接通信。

1.1 在服务端和客户端之间(一对多)

用于客户端程序登录,验证用户密码,获取其他在 线好友信息等。

1.2 在客户端和客户端之间(多对多)

用于在线好友间直接通信聊天。此时每个客户端程序 上既有实现服务端功能的部分,又有实现客户端功能的部 分,前者用于接收聊天数据,后者用于发送聊天数据。

在用 C/S 模式进行通信时,作为客户端在请求与服务端连接时需要知道服务端的 IP 地址,腾讯公司的服务器具有固定的公网地址,这没有问题。但是在客户端和客户端之间通信时,每个用户的 IP 地址都是 ISP 所给的临时地址,无法固定,那么某个用户要与另一个用户连接到,怎么知道对方的 IP 地址呢?当一个用户连接到 Internet 后,获得了一个临时的公网地址,当登录到QQ 时,QQ 服务端会获得该上线用户的 IP 地址,然后告知其他要与该上线用户聊天的用户,其他用户就可以连接该用户与之直接通信了。

在 Win32 平台上,对于众多的基层网络协议, Winsock 是访问它们的首选接口。如果想从头开发一个 网络通信应用程序,TCP/IP 就是首选协议之一。为了保 证可靠的数据传输,宜选择 TCP 协议。在开发工具方面, VC 是我的首选,但 VB 可能拥有更多的使用者,所以 用 VB 6.0 作为开发工具编程实现,以便更多的读者能 够理解。喜欢 VC 的读者要将其转化成 VC 下的代码也 很容易。在 VB 中有一个 Winsock 控件,它为 VB 网络 编程提供了一条便捷的途径。

本程序中用到的 Winsock 控件如表 1 所示。

表 1

控件名	是否为数组	作用	在哪个程序中
WinsockServer	是	用于和客户端通信	服务端程序
WinsockClient	否	用于和服务端通信	客户端程序
WinsockClientServer	是	用于接收在线好友的聊天信息	客户端程序
WinsockClientClient	是	用于向在线好友发送聊天信息	客户端程序

本程序中所涉及的一些其他功能如数据库访问等的 实现,限于篇幅不进行阐述,有兴趣的读者可以参考源 程序。为了阐述方便,本程序中服务端只用了一个窗体, 客户端只用了两个窗体,分别是登录窗体和聊天窗体。

接下来按照 QQ 聊天程序的所需实现的功能一步步 编程实现,其中服务端程序和客户端程序没有分开阐述,而是按照所应实现功能的先后次序一并阐述,便于 读者理解及按此顺序编程调试。

#### 2 编程

#### 2.1 登录

使用 QQ 的人都知道,用 QQ 聊天第一步是登录, 这一步看似简单,其实程序做了不少事情。当用户启动 QQ 聊天客户端程序,出现登录界面,用户输入用户名和 密码,单击"登录"按钮后,在用户机上的客户端程序 首先请求与服务端程序建立连接,服务端程序接受客户 端程序连接请求。客户端程序在确保已和服务端程序建 立连接后发送登录时用户输入的用户名和密码,服务端 程序收到后,根据用户信息数据库中数据验证。如果验 证为合法用户,客户端程序则显示聊天窗体,并从服务 端获取在线好友信息,以便直接与各在线好友逐一建立 新的通信机制,实现好友间直接通信。具体实现步骤如下:

2.1.1 监听

先在窗体上加一个 Winsock 控件,名称设为 "WinsockServer"。由于服务端需要和所有客户端通信, 需要很多 Winsock 控件,因此把刚加入的 Winsock 控件 WinsockServer 改为数组,只要将 Winsock 控件的 Index 属性设为 0 即可,此时 WinsockServer 数组只有一个元 素,以后可根据需要动态增减。当窗体载入时,只会创 建 WinsockServer(0)这一个 Winsock 控件实例。作为服 务端程序,首先要有一个 Winsock 招于监听客户端的连 接请求,自然就用 WinsockServer(0)作为监听 Winsock, 所以先要设置其通信协议为 TCP,然后设置本地端口, 以便客户端通过该端口与服务端连接,最后通过 Listen 方法监听。

Private Sub Form\_Load() ConDB '自定义函数 ConDB 用于和存储用户信息的数 '据库连接,具体实现见源程序。 WinsockServer(0).Protocol = sckTCPProtocol '设置通信协议为 TCP WinsockServer(0).LocalPort = 8888 '设置本地端口 WinsockServer(0).Listen '监听 End Sub

2.1.2 请求与服务端程序建立连接

当用户输入用户名和密码单击"登录"按钮后,在

用户机上的客户端程序所做的事情是先请求与服务端 程序建立连接,这需要先在客户端程序的登录窗体上加 一个 Winsock 控件,名称设为"WinsockClient"。然后 设置通信协议,要连接的服务器 IP 地址,远程端口号 要与服务端监听所设置的本地端口一致,最后通过 Connect 方法请求与服务端程序建立连接。

Private Sub Command1\_Click() '单击登录按钮事件 WinsockClient.Protocol = sckTCPProtocol '设置通信协议为 TCP WinsockClient.RemoteHost = "127.0.0.1" '要设置正确的服务器 IP 地址 WinsockClient.RemotePort = 8888 '远程端口 WinsockClient.Connect '请求与服务端程序建立连接 End Sub

2.1.3 接受客户端程序连接请求

当客户端请求与服务端程序建立连接时,处于监听 状态的服务端会收到消息触发 ConnectionRequest 事件, 所以服务端程序接受客户端程序连接请求的工作在 ConnectionRequest 事件方法中完成。由于 WinsockServer(0) 专用于监听,所以要新加载一个 Winsock 控件与该客户 端通信,为了保留每个在线客户的信息,我定义了一个 OnlineUserInformation 结构类型的数组 OnlineUserInfo, 它和 Winsock 控件数组相对应,比如 OnlineUserInfo(i) 中存放的是与 WinsockServer (i)通信客户端的信息,其 中 OnlineUserInfo(i).bUsed 表示 WinsockServer (i)控件是 否已被载入在与某个客户端通信使用,如未使用,则通 过 Load 方法将 WinsockServer (i)控件动态载入,随后调 用 Accept 方法接受客户端程序连接请求。其中 MaxOnlineUser是一个常量,表示最大在线用户数量。

Type OnlineUserInformation bUsed As Boolean '对应下标的 Winsock 控件是否已被载入在用 UserID As String '对应的用户 ID IPAddr As String '对应的客户机 IP 地址 bLogined As Boolean '对应的用户是否登录 End Type Public OnlineUserInfo(1 To MaxOnlineUser) As OnlineUserInformation Private Sub WinsockServer ConnectionRequest(Index As Integer, ByVal requestID As Long) If Index = 0 Then For i = 1 To MaxOnlineUser If Not OnlineUserInfo(i).bUsed Then Load WinsockServer(i) WinsockServer(i).Accept requestID OnlineUserInfo(i).bUsed = True Exit For End If Next i End If End Sub

2.1.4 发送登录的用户名和密码

客户端程序在服务端程序接受连接后会收到消息 触发 Connect 事件,对客户端而言在 Connect 事件方法 中发送登录用户名和密码是最佳时机,为了使接收数据 的一方明白收到的数据代表什么含义,每条数据最前面 有代表不同含义的数据含义标记字符串,每条数据后有 相同的数据结束标记符,每条数据中各子数据间有子数 据分割标记符。各数据含义标记字符串如下:

Public Const OnlineFriendIPID = "FriendIPID" '表示本数据 中包含在线好友的 IP 地址和 ID

Public Const Login = "User Login" '表示本数据中包含登录用户的用户 ID 和密码 Public Const NotUser = "IsNot User" '本数据含义为"不是合法用户" Public Const IsUser = "ThisIsUser" '本数据含义为"是合法用户" 为了识别数据含义方便,含义标记字符串长度一

致,这里采用的长度为 10。 Public Const HeadLength = 10

关于分割标记符和结束标记符,我采用数值1和2 转字符的方法,因为它们转成的字符不是普通字符,不 会和传送的数据如用户 ID、密码、IP 地址等字符串数 据混淆。

具体发送数据通过 SendData 方法进行,发送登录 用户名和密码的语句如下,数据的格式为:数据含义标 记字符串+用户名+分割标记符+密码+结束标记符。

Private Sub WinsockClient\_Connect()

WinsockClient.SendData Login & TextUsername.Text & Chr(1) & TextPassword.Text & Chr(2)

End Sub

2.1.5 接收收据

当有数据到达时,程序会收到消息触发 DataArrival 事件,所以不论服务端还是客户端程序接收数据的工作 在 DataArrival 事件方法中完成。具体接收数据使用 GetData 方法进行,为了防止客户端发送来的数据太多, 服务端来不及处理,定义了一个字符串数组 BufferRecv, 该数组与 WinsockServer 控件数组对应, WinsockServer (Index) 控件收到的数据添加在 BufferRecv(Index)中,然 后通过结束标记符在 BufferRecv(Index)中取出一条条数 据进行分析。如果是客户登录,则获取用户名和密码, 然后通过自定义函数 Lookfor 在数据库中验证。如果验 证为合法用户,则发送"是合法用户"数据标记,然后 将用户 ID 和用户机器的 IP 等信息置于一个在线用户信 息的结构数组中备用。然后通过自定义函数 GetFriend 在数据库中获取该用户的所有好友 ID 及好友人数,并 根据在线用户信息数组逐一判断各个好友是否在线,如 果在线,则将在线好友的 ID 及 IP 地址发送给刚登录的 客户,同时也将刚登录客户的 ID 及 IP 地址发送给在线 好友,以便他们相互之间能够通信。如果验证为非法用 户,则发送"不是合法用户"数据标记,并将与该客户 通信用的 Winsock 控件卸载。 Private Sub WinsockServer\_DataArrival(Index As Integer, ByVal bytesTotal As Long) Dim tmpstr As String Dim EndFlagLoc As Integer '结束标记所处位置变量 If Index = 0 Then Exit Sub WinsockServer(Index).GetData tmpstr, , bytesTotal '接收收据存入 tmpstr 中 BufferRecv(Index) = BufferRecv(Index) & tmpstr '将收到的数据添加在 BufferRecv(Index)中 EndFlagLoc = InStr(BufferRecv(Index), Chr(2)) '在 BufferRecv(Index)中寻找第一个结束标记符的位置 While EndFlagLoc > 0'在 BufferRecv(Index)中取第一个结束标记符左边的数据, '即第一条数据 strMsg = Left\$(BufferRecv(Index), EndFlagLoc - 1) '获取本条数据前面的数据含义标记字符串以分析本条数 '据的含义 Select Case Left\$(strMsg, HeadLength) Case Login '如果是登录数据 Dim tmpPassWord As String Dim tmpUserID As String '通过 Mid 方法取得数据含义标记字符串之后分割标记 '符之前的数据,即登录用户名 tmpUserID = Mid(strMsg, HeadLength + 1,InStr(strMsg, Chr(1)) - 1 - HeadLength) '通过 Mid 方法取得分割标记符之后的数据,即登录密码 tmpPassWord = Mid(strMsg, InStr(strMsg, Chr(1)) + 1)'通过自定义函数 Lookfor 在数据库中查询该用户名密 '码是否存在,如存在则发送含义为"是合法用户"的 '数据, Lookfor 函数的具体实现见源程序 If Look for(tmpUserID, tmpPassWord) Then WinsockServer(Index).SendData IsUser & Chr(2) DoEvents '将客户端的 IP 地址和用户 ID 等信息置于一个有关在 '线用户信息的结构数组中备用,并设置登录标志变量 'bLogined 为 True OnlineUserInfo(Index).IPAddr = WinsockServer (Index).RemoteHostIP OnlineUserInfo(Index).UserID = tmpUserID OnlineUserInfo(Index).bLogined = True Dim AllFriendID As String Dim FriendID As String Dim FriendCount As Integer Dim ToldFriendCount As Integer '然后通过自定义函数 GetFriend 在数据库中获取该用户的 '所有好友的 ID 及好友人数,分别放在参数 AllFriendID '和 FriendCount 中 If GetFriend(tmpUserID, AllFriendID, FriendCount) Then '然后根据在线用户信息数组用 InStr 函数逐一判断在线用 '户是否是该用户的好友 For i = 1 To MaxOnlineUser If OnlineUserInfo(i).bLogined Then

If InStr(AllFriendID, OnlineUserInfo(i).UserID) > 0 Then '如果是,则将在线好友的 ID 及 IP 地址发送给刚登录的 '客户。发送数据的格式为:数据含义标记字符串+IP 地址 '+分割标记符+用户 ID+结束标记符 If WinsockServer(Index).State = sckConnected Then WinsockServer(Index).SendData OnlineFriendIPID & OnlineUserInfo(i).IPAddr & Chr(1) & OnlineUserInfo(i).UserID & Chr(2) DoEvents End If '同时也将刚登录客户的 ID 及 IP 地址发送给在线好友, '以便他们相互之间能够通信 If WinsockServer(i).State = sckConnected Then WinsockServer(i).SendData OnlineFriendIPID & OnlineUserInfo(Index).IPAddr & Chr(1) & OnlineUserInfo(Index).UserID & Chr(2) DoEvents End If ToldFriendCount = ToldFriendCount + 1End If End If If ToldFriendCount = FriendCount Then Exit For Next i End If '如果验证为非法用户,则发送含义为"不是合法用户" '的数据标记,并将与该客户通信用的 Winsock 控件卸载 Else WinsockServer(Index).SendData NotUser & Chr(2) DoEvents OnlineUserInfo(Index).bUsed = False WinsockServer(Index).Close Unload WinsockServer(Index) End If '然后从 BufferRecv(Index)中删除已分析的这条数据,继续 '分析下一条数据 BufferRecv(Index) = Mid(BufferRecv(Index), EndFlagLoc + 1) EndFlagLoc = InStr(BufferRecv(Index), Chr(2)) Wend End Sub

2.1.6 客户端接收数据

客户端程序在收到数据后,对每条数据进行分析。 如果不是合法用户,则显示"你不是合法用户!";如果 是合法用户,则隐藏登录窗体,显示聊天窗体;如果是 在线好友信息,则提取该在线好友的 ID 及 IP 地址,然 后通过自定义过程 AddConnectOnlineFriend 将该在线好 友信息保存,并与该在线好友建立连接。

Private Sub WinsockClient\_DataArrival(ByVal bytesTotal As Long)

Dim tmpstr As String Dim EndFlagLoc As Integer WinsockClient.GetData tmpstr, , bytesTotal BufferCmd = BufferCmd & tmpstr EndFlagLoc = InStr(BufferCmd, Chr(2)) While EndFlagLoc > 0 strMsg = Left\$(BufferCmd, EndFlagLoc - 1) Select Case Left\$(strMsg, HeadLength) Case NotUser MsgBox "你不是合法用户!"

Case IsUser MyID = TextUsername.Text '将用户 ID 保存在 MyID 中以备后用 loginfrm.Hide Form1.Show Case OnlineFriendIPID Dim UserIP As String Dim UserID As String UserIP = Mid(strMsg, HeadLength + 1, InStr(strMsg, Chr(1)) - 1 - HeadLength) UserID = Mid(strMsg, InStr(strMsg, Chr(1)) + 1)Form1.AddConnectOnlineFriend UserID, UserIP End Select DoEvents BufferCmd = Mid(BufferCmd, EndFlagLoc + 1)EndFlagLoc = InStr(BufferCmd, Chr(2)) Wend End Sub 自定义过程 AddConnectOnlineFriend 定义在聊天窗 体中,首先将服务端发来的在线好友的 IP 地址和 ID 等 保存在一个有关在线好友信息的结构型数组 OnlineFriendInfo 中,然后加载一个 Winsock 控件 WinsockClientClient(i),用于给在线好友发送聊天信息, 加载后同样要设置协议、远程 IP 地址、远程端口, 然 后以客户端方式请求与该好友连接。 Public Sub AddConnectOnlineFriend(UserID As String, UserIP As String) For i = 1 To MaxOnlineFriend If Not OnlineFriendInfo(i).bUsed Then OnlineFriendInfo(i).bUsed = True OnlineFriendInfo(i).IPAddr = UserIP OnlineFriendInfo(i).UserID = UserID Load WinsockClientClient(i) WinsockClientClient(i).Protocol = sckTCPProtocol WinsockClientClient(i).RemotePort = 6000 WinsockClientClient(i).RemoteHost = OnlineFriendInfo(i).IPAddr If WinsockClientClient(i).State = sckClosed Then WinsockClientClient(i).Connect Exit For End If Next i End Sub 所以在客户端程序中为了以服务端方式接受在线 好友的连接,在聊天窗体加载时,要设置 Winsock ClientServer(0)这个 Winsock 控件, 以便监听在线好友的 连接。 Private Sub Form Load()

WinsockClientServer(0).Protocol = sckTCPProtocol WinsockClientServer(0).LocalPort = 6000 WinsockClientServer(0).Listen '开始监听 End Sub

2.1.7 客户端接受在线好友的连接请求

在以服务端方式接受连接请求时,根据 bUsedServer(i)判断与之对应的WinsockClientServer(i)是

232

否被使用,如未使用,则通过 Load 方法将 WinsockClient Server(i)控件动态载入,随后调用 Accept 方法接受客户 端程序连接请求。

Public bUsedServer(1 To MaxOnlineFriend) As Boolean Private Sub WinsockClientServer\_ConnectionRequest(Index As Integer, ByVal requestID As Long)

If Index < 0 Then Exit Sub

For i = 1 To MaxOnlineFriend If Not bUsedServer(i) Then Load WinsockClientServer(i) WinsockClientServer(i).Accept requestID bUsedServer(i) = True Exit For End If Next i End Sub

2.1.8 客户端显示在线好友

当在线好友接受连接后,则将该在线好友的 ID 显示在列表框 ListFriendID 中,并将与该在线好友连接的 Winsock 的下标值加入另一个隐藏的列表框 ListSockIndex 中,两个列表框对应,以便后面从 ID 列表框中选择好友发送聊天消息时,能马上知道与该好友连接的 Winsock 下标值。

Private Sub WinsockClientClient\_Connect(Index As Integer) If Index = 0 Then Exit Sub ListFriendID.AddItem OnlineFriendInfo(Index).UserID ListSockIndex.AddItem Index End Sub

#### 2.2 聊天

完全在客户端程序之间进行。

2.2.1 发送

注意聊天者发送聊天数据前先要在列表框 ListFriendID 中选择欲与之通话的好友 ID。程序根据所 选好友在 ListFriendID 列表框中的位置从与之对应的 ListSockIndex 列表框中找到 WinsockClientClient 控件的 下标值,然后用该 Winsock 发送聊天内容,同时通过自 定义过程 SetRichColorText 将聊天内容、聊天双方 ID 及 时间以不同颜色显示在发送端聊天窗体的 RichTextBox

```
中, SetRichColorText 过程的具体实现见源程序。
Private Sub CommandTcpSend_Click()
If ListFriendID.ListIndex < 0 Then Exit Sub</li>
SockIndex = ListSockIndex.List(ListFriendID.ListIndex)
If WinsockClientClient(SockIndex).State = sckConnected Then
SetRichColorText CStr(Now), MyID, ListFriendID.Text,
Text1.Text, False
WinsockClientClient(SockIndex).SendData Text1.Text & Chr(2)
DoEvents
End If
End Sub
```

### 2.2.2 接收

接收时读取每条聊天内容,并根据发送方的 IP 地址 WinsockClientServer(Index).RemoteHostIP 从 Online FriendInfo 数组中找到发送方的 ID,通过自定义过程 SetRichColorText 将聊天内容、聊天双方 ID 及时间以不 同颜色显示在接收端聊天窗体的 RichTextBox 中。

Private Sub WinsockClientServer\_DataArrival(Index As Integer, ByVal bytesTotal As Long) If Index = 0 Then Exit Sub Dim tmpstr As String Dim EndFlagLoc As Integer WinsockClientServer(Index).GetData tmpstr, , bytesTotal BufferMsg(Index) = BufferMsg(Index) & tmpstr EndFlagLoc = InStr(BufferMsg(Index), Chr(2))While EndFlagLoc > 0strMsg = Left\$(BufferMsg(Index), EndFlagLoc - 1) For i = 1 To MaxOnlineFriend If OnlineFriendInfo(i).IPAddr = WinsockClientServer (Index).RemoteHostIP Then Exit For Next i SetRichColorText CStr(Now), OnlineFriendInfo(i).UserID, MyID, strMsg, True BufferMsg(Index) = Mid(BufferMsg(Index), EndFlagLoc + 1)EndFlagLoc = InStr(BufferMsg(Index), Chr(2)) Wend End Sub

#### 2.3 离线

当一个用户离线时,作为聊天程序的服务器程序以及其他在线好友的聊天程序上与之通信的Winsock都会收到 Close 消息,此时要做一些处理工作。

2.3.1 服务器程序

清除该在线用户的信息,卸载与之通信的 WinsockServer 控件。

Private Sub WinsockServer\_Close(Index As Integer) OnlineUserInfo(Index).bUsed = False OnlineUserInfo(Index).bLogined = False OnlineUserInfo(Index).IPAddr = "" OnlineUserInfo(Index).UserID = "" BufferRecv(Index) = "" WinsockServer(Index).Close Unload WinsockServer(Index) End Sub

2.3.2 客户端程序

卸载与之通信的 WinsockClientClient 控件,从 ListFriendID 列表框中清除该在线好友的 ID,从 ListSockIndex 列表框中清除该在线好友所对应 WinsockClientClient控件的下标值,从OnlineFriendInfo 数组中清除该在线好友的信息。 Private Sub WinsockClientClient\_Close(Index As Integer) WinsockClientClient(Index).Close Unload WinsockClientClient(Index) For i = 0 To ListFriendID.ListCount - 1 If ListFriendID.List(i) = OnlineFriendInfo(Index).UserID Then ListFriendID.RemoveItem i ListSockIndex.RemoveItem i Exit For End If Next i OnlineFriendInfo(Index).bUsed = False OnlineFriendInfo(Index).IPAddr = ""

```
OnlineFriendInfo(Index).UserID = ""
BufferMsg(Index) = ""
End Sub
卸载与之通信的 WinsockClientServer 控件。
Private Sub WinsockClientServer_Close(Index As Integer)
If Index = 0 Then Exit Sub
bUsedServer(i) = False
WinsockClientServer(Index).Close
Unload WinsockClientServer(Index)
End Sub
本程序在 VB 6.0 中已通过调试。
```

位机 (PC 机) 和下位机 (PC104) 部分。上位机和下位

机通过 RS232 进行通讯。上位机负责数据处理和图形化

显示,它是在 MATLAB 7.0 环境下进行的,应用了仪器控

制工具箱中关于串行通信的函数,通过程序控制整个系

统的工作并进行数据处理。下位机负责数据采集和响应

在 Matlab 7.0 中的设备控制工具条 (instrument

control toolbox)用来负责上、下位机之间的通信。该设

备控制工具箱的特色如下: Instrument Control Toolbox

包括两大部件: M 文件函数和接口驱动适配器。这两部

件提供了 Matlab 与外设的通讯功能如图 1 所示。

主机的控制处理信号。

2 通信原理

张斌

# 4.9 MATLAB 串行通信的实现方法

MATLAB 是由美国 Mathworks 公司开发的面向理 论分析研究和工程设计处理的一套具有强大功能的软 件系统。在设计研究单位和工业部门,MATLAB 被认作 进行高效研究、开发的首选软件工具。其编程语法规则 与一般的结构化编程语言大同小异,而且使用更方便, 具有一般编程基础的用户都可以很快熟练掌握。 MATLAB 内置大量的数学算法函数,只要几个语句就能 实现诸如矩阵/行列式运算、FIR/IIR 滤波等数据分析处 理、复杂的图形显示等。在实际应用中常常要通过各种 方式将数据传入计算机,进而借助 MATLAB 的强大分 析处理能力对数据进行分析、处理和图形显示。

本文提出在上位机 MATLAB 7.0 环境下,使用 RS232 串行接口实时接收。该方法较之使用 VC/VB 高 级语言编程,极大地缩减了开发时间,提高了开发效率。

## 1 总体框图

本设计的总体原理框图如图1所示,系统主要由上



由图 1 可见,MATLAB 的两大部件提供了一条外设 与MATLAB 之间的交互通道,使用户与外设之间进行信 息获取和传送。同许多高级语言一样,MATLAB 通过调 用 M 文件函数来创建设备对象,得到设备的文件句柄, 而设备驱动程序作为操作系统中直接控制硬件的模块, 是链接操作系统内核与系统外部设备 I/O 操作的关键模 块。它将具体的硬件细节隐藏实现对外设操作的透明, 因此 MATLAB 可以像操作文件一样实现对外设的读、写 操作。同时 MATLAB 支持二进制和文本(ASCII)两种 串行通信方式,文本方式支持 SCPI(Standard Commands for Programmable Instruments)语言;支持异步通信和同 步通信;支持查询和基于事件驱动的两种串行通信方式。

## 3 MATLAB 串行通信

在 MATLAB 环境下,读取串口数据的方式可以分为两种:查询法和事件驱动法。以查询的方式进行串行通信时,如下位有大量的数据分时分批传送给 PC 机,上位机就需要不停查询串行口的缓冲区,有数据就读取;虽然编程容易,但这样无法实现对数据进行实时处理和图形显示,系统实时性不高,而且会极大地占用系统的资源和出现丢帧现象。以事件驱动的方式对串口进行控制实现串行通信,就可以实时处理下位机传送的数据;但编程相对复杂一些,需要采用 MATLAB 的事件和回调函数机制。

#### 3.1 查询法

MATLAB 使用查询法对进行串行通信的程序如下: g = serial('com1');%创建串口对象 g.baudrate=115200; %设置波特率,缺省 9600bit/s %设置校验位无奇偶校验 g.parity='none'; g.stopbits=1; %设置停止位 g.inputbuffersize=256; %设置输入缓冲区为 256b, 缺省值为 512b g.Terminator='LF';%设置输入缓冲区为256b,缺省值 recbuf=zeros(1,40); %清接收缓冲区(40) framelen=23: %帧长度(每帧 23byte) framenum=0: %清接收帧数 rectr=0: %接收计数器清零 xctr=0: %输入接收的数据帧数 recnum= input('请输入要接收的数据帧数:\n'); %打开串口设备 %打开串口设备对象 g fopen(g); fwrite(g,255);%以二进制的方式发送握手信号 0xFF %进入接收循环 while framenum<recnum recdta=fread(g,1,'uint8'); %以二进制的方式读入数据 if recdta==255 %找帧头 %接收计数器加1 rectr=rectr+1: recbuf(rectr)=recdta; %送入接收缓冲区 elseif recdta==framelen&rcbuf(1)==255 %验证接收长度和帧头 %取帧长度 framelen=recdta: rectr=rectr+1. %接收计数器加1 %送入接收缓冲区 recbuf(rectr)=recdta;

elseif rectr>1&rectr<framelen %接收数据 %接收计数器加1 rectr=rectr+1. recbuf(rectr)=recdta; %送入接收缓冲区 if rectr==framelen %如果接收完进行处理 rectr=0: %清接收计数器 framenum=framenum+1; %帧数累加 %根据具体通讯协议提取数据 %接收滤波后的数据 plot(...) %输出显示 end %未找到帧头清计数器 else rectr=0: %接收结束 end end %主循环结束 %程序结束关闭串口类 fclose(g): %关闭串口 %删除串口对象 delete(g): clear g; %清除变量

注意: 串口的读写操作有文本方式和二进制方式两种。在文本方式下,串口设备只接收可以显示的 ASCII 字符,将 0x00H~0x1FH 范围内的字符解释为控制字符。使用文本方式对串口的读写函数分别为 fprintf()(写)、fscanf()(读)。

#### 3.2 事件驱动法

在 MATLAB 环境下以事件驱动的方式进行串行通 信,类似于 VC 中的 MSCOMM 控件中 ONCOMM 事件 的实现方法。MATLAB 7.0提供了 instrcallback(obj,event) 回调函数,用户根据需要可以自行设置具体的串行通信 事件。MATLAB 常用的串行口通信中断事件有:缓冲区 有指定字节数目的数据可用事件(bytes-available event)、串口接收到的数据长时间处于非激活状态事件 (break-interrupt event)、串行口引脚状态改变事件 (pin-status event)、输出缓冲区为空事件(output empty event)等。当串口上有监视的事件发生时,MATLAB 会自动调用回调函数进行通信事件的处理。因此,事件 驱动实质上是一种中断机制,而回调函数实质上相当于 一个中断服务子程序。以下是具体的编程步骤。

```
3.2.1 主程序
```

g=serial('com1'); %创建串口对象
g.baudrate=115200; %设置波特率,缺省 9600bit/s
g.parity='none'; %设置校验位无奇偶校验
g.stopbits=1; %设置停止位
g.BytesAvaibleFcnMode='byte';
% 设置中断触发事件为 'bytes-available Event'
g.BytesAvailableFcnCount=30;
% 设置接收缓冲区每收到 30 个字节时,触发回调函数
g.BytesAvailableFcn=@instrcallback;得到回调函数句柄
fopen(g);%连接串口设备对象
fwrite(g,255);%写串口,发送握手信号 0xFF(等价于十进制下的数值 255)
3.2.2 中断调用程序

Matlab 缺省的回调函数 instrcallback(obj,event)存在

235

于 instrcallback.m 文件中。该文件类似于 VC 中的 ONCOMM 事件函数模块。需用户自行添加相应的中断 处理程序代码。

Matlab 安装目标盘下有两个 instrcallback.m 文件, 我们只需要修改 Matlab7\toolbox\matlab\iofun\@ instrument 目录下的 instrcallback.m 文件即可。注意,在 修改 instrcallback.m 文件之前,最好对进行备份。另外, 需要注意的是:程序调试过程中如果再次修改了该回调 函数,要重新启动 MATLAB,才能使得新的回调函数文 件生效。

%修改后的 instrcallback.m 文件如下

function instrcallback(obj, event)

%以下是用户自行添加的通信事件处理及通信数据处理 的中断服务程序代码



(每个数据占一个字节)

%以下程序用以数据进行滤波 inbuffer2=filter(...., Inbuffer1);

%以下程序用以对滤波后数据进行文件存储

fid = fopen('C:\MATLAB7\work\data.txt','a+'); %创建并

data.txt 文件

fprintf(fid,'%3d',inbuffer2);%将滤波后数据存储到 fid 文件 设备中

fclose(fid); %关闭 fid 文件设备 %数据计算

% 进行实时图形输出显示,如图2所示。 plot(...); end





实验证明,基于 MATLAB 事件驱动方式的实时串 行通信稳定可靠,处理数据方便,编程简单,开发效率 大大提高。

## 4 结论

本文介绍的基于 MATLAB 环境下 PC 机与 PIC单片

机串行通信的实现方法,利用 MATLAB 的 Instrument Control Tollbox 的 serial 类及 instrcallback()回调函数,实 现基于事件驱动的实时通信。使开发人员使用 MATLAB 工具箱中的现有函数,方便地实现串行通信,充分的利 用 MATLAB 中独有数据分析处理和图形显示函数,大 大简化系统上位机软件的编程工作量,使系统开发人员 可以专注于系统的算法实现和优化。

## 4.10 手机视频播放器的开发

汪永松

## 1 引言

作为任何一个拥有多媒体功能手机的人而言,如果 能够自主开发出一款定制的手机视频播放器是多么令 人有成就感的事情。如果说仅仅是在 WTK 模拟器上, 使用 MMAPI (移动媒体 API, JSR135) 就可以做出一 个简单的视频播放器来, 然而要真正应用到实机却不是 那么一帆风顺的事情, 在下文中逐步向大家说明。图 1 是该款视频播放器在 WTK 模拟器和 Nokia5310 实机上 的运行界面。



图 1 视频播放器运行界面

该款播放器在实机上支持 3gp、mp4 和 wmv 等常见 视频媒体类型以及 gif 动画。

## 2 功能分析

作为一款实用的手机视频播放器,不仅要考虑界面 的个性化,而且还要考虑较高的播放效率和手机厂商对 视频媒体类型的支持,归纳为以下3个方面:

(1)实现播放的基本功能。包括播放控制、音量 调节、扩展卡资源读取支持等。

(2)流畅的播放效果。本案例中采用播放池(Player Pool)策略,这一策略使播放过程更为流畅。造成播放 停滞的原因主要有两个方面:一方面基于 MIDlet 的安 全考虑,每当读取本地文件时手机系统都要求进行手动 确认,这样就会间断播放过程。另外一方面每次对音乐 文件的播放预处理也会造成一定的播放间隔。

(3) 完全满足手机厂商所支持的视频媒体类型, 支

持尽可能多的视频媒体类型。本案例中采用的是系统自动匹配的策略,即由手机系统本身来识别播放视频类型。

以上的要求似乎比较简单,但是由于模拟环境和实 机环境的差异,在实际开发中总会遇到一些小的问题, 接下来就和朋友们分享开发手机视频播放器的一些经 验和技巧。

## 3 设计分析

#### 3.1 PlayerMIDlet

主 MIDlet 也是整个程序入口,程序启动时显示播 放列表面板。

### 3.2 VideoPanel

播放列表面板,为程序主界面,负责播放列表和对 象池管理。

#### 3.3 PoolHelper

对象池帮助类,负责生成播放对象池。

#### 3.4 PlayCanvas

播放布,用来装载媒体播放。

## 3.5 AudioSettingPanel

#### 3.6 音量控制面板,实现对播放过程的音量控制

另外,视频播放器工程还引入了文件选取模块,用 于生成播放文件列表。图2是该工程。





## 4 实现过程

#### 4.1 PlayerMIDlet 类

PlayerMIDlet 仅仅提供了 MIDlet 框架, 在启动界面 创建视频播放面板:

public class PlayerMIDlet extends MIDlet {

//视频播放器面板 private VideoPanel panel; protected void startApp() { //进入视频播放界面 panel = new VideoPanel (this);

#### 4.2 VideoPanel 类

视频播放面板类主要负责播放列表和对象池管理。 表1是视频播放面板类的主要成员。

编号	成员	目的
1	播放列表 List	存储/管理播放列表
2	文件选取类实例	选择视频文件,生成播放列表
3	对象池帮助类实例	生成视频播放和音量控制对象池
4	播放和音量控制对象池	对象池管理
5	播放布	提供视频播放控制

表 1 视频播放面板类的主要成员

(1) 文件选取模块的调用

视频播放面板的"添加播放列表"菜单调用文件选 取类实例来生成播放列表,同时生成播放项目的播放和 音量控制对象,并提供播放入口菜单:

```
public void commandAction(Command c, Displayable d){
    ...
      else if(c == cmdAddPlayList) {//添加播放列表
        chooser = new FileChooser(this);
      3
    //结束添加列表
    public void finishAddList() {
      //显示当前界面(必须在 showSelected 之前)
      display.setCurrent(playList);
      //显示选择结果
      showSelected();
    3
    //显示选择结果
    private void showSelected() {
      //当前选取播放资源
      String uri = chooser.getCurrentFile();
      if(uri == null) {//取消选择
        return;
      if(URITable.indexOf(uri) == -1) {//该资源不存在
        //添加资源列表
        URITable.insertElementAt(uri, 0);
        //以插入方式
        playList.insert(0, getRelativeName(uri), iconHelper.
          getIconByExt(extractExt(uri)));
      else {//该资源已经存在,无需更新
        Alert alert = new Alert("Error", "This item already
          exists!!", errorImage, AlertType.ERROR);
        alert.setTimeout(Alert.FOREVER);
        Display.getDisplay(let).setCurrent(alert);
        return:
      //对象池帮助实例
      helper = new PoolHelper(this, uri);
      helper.start();
      //添加播放入口菜单
      playList.addCommand(cmdPlay);
    以上代码中, 主模块通过提供 finishAddList 方法来
获取文件选取模块的返回。当选取完毕后,对选取项目
```

还要进行重复性判断,如果存在重复则提示,如果不存 在则调用对象池帮助类生成该项目的播放和音量控制 对象,同时提供播放入口菜单。图 3 是在 WTK 模拟器 和 Nokia5310 实机中开始播放列表的界面。



图 3 开始播放列表

```
(2) 播放控制
```

视频播放面板只提供播放入口方法来启动播放布, 真正的播放控制在播放布类中: public void commandAction(Command c, Displayable d){ else if(c == cmdPlay) {//播放 //记录当前选取媒体索引 selectedIndex = playList.getSelectedIndex(); if(initPlay() == true) {//播放成功 } 3 //开始播放 private boolean initPlay() { //播放测试 if(canvas.play(selectedIndex)) { //设置显示 Display.getDisplay(let).setCurrent(canvas); return (true);

```
}
```

```
(3) 对象池的管理
```

对象池的初始化在对象池帮助类完成,在系统关闭

```
时进行对象池的释放:
```

```
public void commandAction(Command c, Displayable d){
    if(c == cmdExit) {//退出系统
        //释放播放池
        freePlayerPool();
        let.exit():
```

```
}
```

•••

```
private void freePlayerPool() {
    int playerCount = playerPool.size();
```

```
//关闭所有的连接和文件流
for(int i = 0; i < playerCount; ++i) {
    //关闭播放器
    Player player = (Player)playerPool.elementAt(i);
    if(player != null)
    {
        player.close();
    }
}</pre>
```

//播放器池和音量调节池,视频控制池 playerPool.removeAllElements(); volumePool.removeAllElements(); videoStatusPool.removeAllElements();

以上代码中关闭了所有视频播放器对象,释放视频 播放的所有相关资源。

#### 4.3 PoolHelper 类

3

对象池帮助类,主要是根据文件选取模块提供的 URI来生成播放项目的视频播放和音量控制对象池:

```
public void run() {
  Player player = null;
  VolumeControl volume = null;
  try
    //创建播放
    player = Manager.createPlayer(URI);
    player.realize();
    //设置播放侦听
    player.addPlayerListener(panel.canvas);
    //获取音量控制
    volume = (VolumeControl)player.getControl
     (VOLUME CTRL NAME);
    if(volume != null) {
    //某些时候可能会为空,该错误无法捕获
      volume.setLevel(62); //0.618
    }
  3
  catch (IOException e) {
    e.printStackTrace();
  catch(MediaException e) {
   e.printStackTrace();
  //添加到媒体文件流容器(以插入方式)
  panel.playerPool.insertElementAt(player, 0);
  panel.volumePool.insertElementAt(volume, 0);
  panel.videoStatusPool.insertElementAt(new Boolean
    (false), 0);
```

以上代码,为了避免访问本地资源所造成的资源 死锁,对象池帮助类为线程类。在 run 函数中通过文件 URI 创建播放器和音量控制对象,并插入到对象池(后 进先出)。

对于视频播放器的创建是整个工程的核心,也是开 发人员比较迷惑的地方。

(1)视频播放对象的困惑

1) 对视频媒体类型的支持。图 4 是常见的媒体类型不支持的异常画面。

📰 Sun Java(III) Wireless Toolkit 2.5.2 for CLDC - FoolAudioPlayer
File Edit Project Help
💽 New Project 📂 Open Project 🛃 Settings 🍓 Build 🊸 Bun 🥑 Clear Cons
Device: DefaultColorPhone
Project "FoolAudioPlayer" loaded
Running with storage root C1Documents and Settings\Paullj2mewtkl2.5.2\appdb\DefaultColorPhone
Running with locale: Chinese_People's Republic of China.936
Running in the identified_third_party security domain
file:///media/audio/chord.wav
file:///media/audio/leftright.wma
javax.microedition.media.MediaException: Cannot create a Player for: Unsupported content type
at javax.microedition.media.Manager.getPlayerFromType(+13)
at javax.microedition.media.Manager.createPlayer(+35)
at javax.microedition.media.Manager.createPlaver(+389)

图 4 视频类型不支持警告

对于 WTK 模拟环境(浏览使用的是 WTK2.5.2), 似乎只支持 mpeg 类型的视频媒体,播放 mp4、3gp 类 型的视频时就会抛出不支持的内容类型的异常。对于手 机实机环境,厂商承诺支持的类型程序都可以支持。在 Nokia5310 环境中,可以支持 3gp、wmv、mp4 等主流 视频媒体类型以及 gif 动画,反而不支持 mpeg 类型。

2)视频播放对方的创建方式。上段代码中并没有 使用输入流和媒体内容类型(content type)来创建播放 器对象,而是通过视频文件 URI 自动创建。

使用输入流创建播放器对象存在制约。在模拟器环 境下调试发现,使用输入流创建播放器的方式不适用较 大的媒体文件,否则将抛出内存不足的异常。实机环境 下,往手机上安装文件大小大于一定限制的 jar 文件时, 会得到一个安装包过大而提示无法安装(大小限制额度 可能因各款手机不同而存在一定差异,但是这个值应该 小于一个常规视频文件的大小)。图 5 就是在将一个文 件大小为 2M 的 jar 文件安装到手机时提示的错误。

诺基亚应	用程序安装器
8	无法将此应用程序安装到手机中。该文件太大。
	( <u>确定</u> )

图 5 安装 jar 文件的错误提示

使用输入流创建视频播放器对象适用于文件大小 较小的视频文件,这也是 J2ME 教程中常用的通过 Class 类的 getResourceAsStream 的方法在资源中获取文件流 来创建播放对对象的方式。

3) 创建视频控制对象的时机。上段代码中并没有在

生成播放对象的同时生成视频控制对象,其主要原因有两点:一方面是视频控制对象需要绑定到播放布 (Canvas),当初始化视频控制对象显示模式时,会产生 视频控制对象视图与当前界面重叠的问题。如图6所示。



图 6 初始化视频控制对象模式造成界面重叠

另外一方面,视频播放布应该是共享的显示资源, 不可能为每个视频控制器创建播放布。基于这种对应关 系,视频控制器的管理应该纳入播放布的范围。

以上的这些"机关"也正是在前文说的仅仅在 WTK 模拟上写一个简单的播放器是很简单的,而要在实机上 写一款能用的视频播放器却是比较困难的原因。

#### 4.4 PlayCanvas

播放布类是视频播放的核心,负责播放控制、播放 侦听以及视频控制对象的管理。

```
(1) 播放控制
public void commandAction(Command c, Displayable d) {
 if(c == cmdPlay) { //恢复播放
   resumePlay();
 else if(c == cmdPause) {//暂停
   if(pausePlay()) {//暂停成功
      removeCommand(cmdPause);
      addCommand(cmdPlay);
 else if(c == cmdStop) {//停止
   if(pausePlay()) {//暂停成功并切换到主界面
                  //隐藏视频显示
      hideVideo();
      //切换显示界面
      Display.getDisplay(let).setCurrent(playList.playList);
   }
 3
 else if(c == cmdSetting) {//音量控制设置
   if(volume != null) {//某些时候可能会为空
      VideoSettingPanel panel = new VideoSettingPane
       l(let, this, volume);
   }
 else if(c == cmdFullScreen) {//全屏
   try {
```

```
addCommand(cmdNormalScreen);
removeCommand(cmdFullScreen);
video.setDisplayFullScreen(true);
}
catch (MediaException screen) {
screen.printStackTrace();
}
else if(c == cmdNormalScreen) {//正常屏幕
Try {
addCommand(cmdFullScreen);
removeCommand(cmdNormalScreen);
video.setDisplayFullScreen(false);
}
catch (MediaException screen) {
screen.printStackTrace();
}
}
}
l b thEmrth twitt to twitt of the time the factors
}
}
```

以上代码中,播放布类提供了播放、暂停、全屏幕、 正常屏幕和音量控制(如图1所示的播放界面)。这里 并没有提供"完全结束播放"(close 方法)的接口,因 为对于播放"完全结束"会造成播放对象的释放,必须 重新创建播放器对象才能再执行播放行为,否则在播放 对象关闭的状态下调用播放方法会弹出异常(详细错误 内容可以参见 MMAPI 参考文档),所以只有当系统关 闭时才考虑"完全结束播放"。

```
闭时才考虑"完全结束播放"。
     (2) 视频控制
    当选择播放项目进行播放时,播放布就会获取播放
对象的视频控制对象并启动播放:
    synchronized public boolean play(final int selectedIndex) {
     //设置播放器
     player = (Player)playList.playerPool.elementAt
        (__selectedIndex);
     volume = (VolumeControl)playList.volumePool.
       elementAt( selectedIndex);
     if(player == null) {
     //播放器不能为空(音量控制器可能为空)
       return (false):
     //设置视频控制
     video = (VideoControl)player.getControl
       ("VideoControl");
     if(video == null) {//视频控制不能为空
       return (false);
```

```
//初始化状态判断, 重复初始化会抛出 mode already set 异常
Boolean isVideoInit = (Boolean)playList.videoStatusPool.
elementAt(__selectedIndex);
if(isVideoInit.booleanValue() == false) {
  video.initDisplayMode(VideoControl.USE_
  DIRECT_VIDEO, this);
  playList.videoStatusPool.setElementAt(new Boolean
  (true), __selectedIndex);
}
try {
  video.setDisplaySize(video.getDisplayWidth())
```

video.setDisplaySize(video.getDisplayWidth(), video.getDisplayHeight() );

3

}
catch (MediaException e) {
}

```
,
return (initPlay() );
```

} //开始播放

synchronized private boolean initPlay() {
try {
//设置视频控制可见

video.setVisible(true);

```
//设置居中显示
video.setDisplayLocation((getWidth()-
video.getDisplayWidth())/2, (getHeight() -
video.getDisplayHeight())/2);
```

//开始播放 player.start();

```
catch(MediaException e) {
    Alert alert = new Alert("Exception", e.getMessage(),
    errorImage, AlertType.ERROR);
    alert.setTimeout(Alert.FOREVER);
    //显示警告
    Display.getDisplay(let).setCurrent(alert);
    e.printStackTrace();
```

```
return (false);
```

} return (true);

,

}

以上代码中,播放布通过对象池中的播放对象来获 取视频控制对象(VideoControl),并初始化视频控制对 象的显示模式和区域大小,然后调整播放显示位置并开 始播放。其中有3个比较重要的地方:

(1)视频控制对象的模式初始化只能成功调用一次,如果再次调用会抛出模式已经设置的异常。所以必须对初始化状态进行管理,避免重复初始化(上段代码中使用视频状态池 videoStatusPool 来进行状态管理)。

(2) 对于部分视频,例如 gif 动画,存在没有音量 控制对象的情形,所以要对音量控制对象进行是否为空 的判断,否则会抛出指针为空的异常。

(3)为了完整监控播放状态,必须设置播放布为播放侦听(PlayerListener)。

## 4.5 AudioSettingPanel

音量控制面板由播放布调用,主要用于提供音量控制的界面,实现当前播放音量的控制:

public class AudioSettingPanel implements CommandListener, ItemStateListener

//Gauge private Gauge gauge;

//改变音量值

public void itemStateChanged(Item item)
{

vc.setLevel(gauge.getValue() );

gauge.setLabel("Volume: " + gauge.getValue() );

以上代码中,该面板类实现了 ItemStateListener, 通过一个 Guage 来实现音量的控制。实际上,各款手 机所呈现的 Guage 组件与 WTK 模拟器显示的是不同 的。图 7 是 Guage 在 WTK 模拟器和 Nokia5310 实机的 运行界面。



图 7 Guage 组件运行界面

通过以上过程可以总结出,在实现手机视频播放器 的过程中要注意以下 7 个窍门:

(1)播放控制由播放布类完成,播放主界面只需 要提供开始播放的入口。

(2)播放布同时要设置为播放侦听,这样才能对 播放状态进行全面的监控。

(3) 在播放控制中并不需要提供"完全结束播放" (close 方法)的接口,可以通过停止接口来实现播放项 目的切换(被停止的项目可以接着播放),否则在播放 器关闭的状态下再执行播放操作会抛出播放异常。

(4) 通过文件 URI 来创建播放器对象,可以避免 使用文件流来创建播放器造成的限制,同时也可以避免 对视频文件类型的管理。

(5)视频控制对象的获取在播放布初始化中进行, 而不能用对象池进行管理。

(6)视频控制对象的模式初始化只能有一次,再 次初始会抛出模式已经设置的异常。所以必须进行初始 化状态的管理。

(7) 对于 gif 动画,其音量控制对象可能为空,需 要对音量控制对象进行是否为空的判断处理。

5 验证环境

该例涉及的开发环境为: JDK 版本 1.6.0\_03, IDE 为 WTK2.5.2 和 EditPlus2.3.1; 验证环境为 WTK2.5.2 模 拟器和 Nokia 5310(配置为 MIDP2.1 和 CLDC1.1)实机。

除了选择与配置对应的 API 之外,由于手机视频播 放器涉及到 MMAPI 和 PDA 可选包的使用,所以在工

程 API 设置中还要进行相应的 API 选择。图 8 为手机视频播放器工程的 API 选择。

#### Settings for project "FoolVideoPlayer" 1000 API Selection Target Platform Custom 9 Required Profiles Configuration 20 OMIDP 1.0 O CLDC 1.0 ⊙ CLDC 1.1 Optional O MIDP 2.0 ● MIDP 2.1 A User Defined tional Mobile Media API (TSR 135) PDA Profile for J2ME (JSR e E Bluetooth/OBEX for

图 8 手机视频播放器工程的 API 设置

## 6 结语

在开发论坛中,很多读者在开发视频播放器的过 程中因为模拟器与实机对媒体类型的支持存在差异 而产生困惑,因为播放对象的创建方式导致程序无法 正常运行而灰心,这些也正是手机程序与桌面型应用 软件开发不同的地方,在模拟器上运行正常就会在实 机上运行正常,并不一定在模拟器上运行失败而在实 机上运行失败。更多的还是需要读者根据实机环境去 摸索,敢于试验,这样才能做出一款个性优秀的手机 视频播放器。

## 4.11 蓝牙协议栈 BlueZ 移植及 GPS 实现

## 郑春芳 郭秋丽

## 1 概述

蓝牙技术是一种无线数据与语音通信的开放性全 球规范,它结合了通信技术与计算机技术,为各种 3C (Computer、Communication、Consumer,即计算机、 通信和消费电子产品)设备在没有电线或电缆相互连接 的情况下,能在近距离范围内实现相互通信或操作提供 可能。它具有无线性、开放性、低功耗等特点。

Linux 操作系统下最早的蓝牙协议栈是由 Axis Communication Inc 在 1999 年发布的 OpenBT 协议栈。 其后几年 Linux 蓝牙协议得到了快速发展,到 2001 年, Qualcomm Incorporated 发布的 BlueZ 协议栈。此后, BlueZ 作为 2.6 内核的一部分,成为 Linux 的官方蓝牙 协议栈,得到了广泛应用。

本文实验平台 MP2530 是韩国美智(MagicEyes) 公司专门为嵌入式多媒体处理器 MMSP2+设计的开发 平台。MMSP2+是美智公司旗下一款为提供高性能媒体 功能和低功率损耗特点的多媒体系统芯片,它集成 ARM926EJ和 ARM946E 的双核芯片,支持多种的视频 编解码功能,同时因为它带有 2D 和 3D 的图形加速器, 能够很好地区分 UI 同各种图像内容,包括导航地图、 2D&3D 游戏等。

## 2 Linux 蓝牙协议栈

Linux 蓝牙协议栈,又叫 BlueZ,是一个开放性的协议,目前已经集成到 2.6 及其以后版本的内核

中。对于 Linux 下的各种硬件体系结构,如单处理器系统、多处理器系统及超线程系统,BlueZ都可以很好地运行。

BlueZ 采用模块化设计,其体系结构如图 1 所示。 它包含两大部分:内核模块和用户态模块。内核模块是 由主机控制接口(Host Control Interface, HCI)层、 Bluetooth协议核心、逻辑链路控制和适配协议(Logical Link Control and Adaptation Protocol, L2CAP)、SCO 音 频层、其他 Bluetooth 服务组成。用户态模块包括 BlueZ 工具集和蓝牙应用程序。

在 BlueZ 提供的服务中, BNEP (Bluetooth 网络封 装协议) 实现了 Bluetooth 上的以太网仿真,这使 TCP/IP 可以直接运行于 Bluetooth 之上。BNEP 模块 (net/bluetooth/bnep/) 和用户模式 pand 后台进程实现 了 Bluetooth 个人区域网 (PAN)。BNEP 使用 register\_netdev 将自己作为以太网设备注册到 Linux 网 络层,并使用上面为 WLAN 驱动程序描述的 netif\_rx 来填充 sk\_buffs 并将其发送到协议栈; RFCOMM (net/bluetooth/rfcomm/) 提供 Bluetooth 上的串行仿真, 这使得串行端口应用程序(如 minicom)和协议(如点 对点协议(PPP))不加更改地在 Bluetooth 上运行。 RFCOMM 模块和用户模式 dund 后台进程实现了 Bluetooth 拨号网络。

#### 3 移植 BlueZ 到 MP2530 开发板上

移植 BlueZ 到开发板上,除了要移植 BlueZ 库和 一些 BlueZ 工具集外,还需重新配置内核,增加内核 对蓝牙的支持。从 Linux 2.6 内核开始, BlueZ 协议栈 就作为内核的一部分增加到内核中。而在 2.6 以前,要 想内核支持蓝牙,必须对内核打上相应补丁。本文 PC 机内核版本是 2.6.9,开发板的内核版本是 2.6.20。为 了实现在开发板上对 USB 蓝牙设备(实验中用的是 USB 接口的蓝牙适配器)的驱动,必须在 PC 机上交 叉编译支持蓝牙的内核,移植 BlueZ 库和工具集,最 后烧写到开发板上。



图 1 BlueZ 体系结构

## 3.1 配置、编译内核,增加对蓝牙的支持

如何配置内核,不是本文的重点,在这里不再详述, 这方面具体可以参考其他有关文档。这里只介绍有关蓝 牙部分的配置。

2.6 以后的内核,已集成了 BlueZ 协议栈,只要选 上相关配置选项,就可以实现内核对蓝牙的支持,具体 操作如下:

(1)运行 make menuconfig 打开内核配置,在 networking 下选上 Bluetooth subsystem support,在 Bluetooth subsystem support下的配置如图 2 所示。

	Bluetooth subsystem support
<*>	L2CAP protocol support
<>	SCO links support
<*>	FCOM protocol support
[*]	IFCOM TTY support
<*>	ENEP protocol support
[*]	Nulticast filter support
[*]	Protocol filter support
$\langle \rangle$	HDP protocol support
	Bluetooth device drivers>

图 2 Bluetooth subsystem support 配置

相应的配置选项功能如下: L2CAP protocol support 支持逻辑链路控制和适应协议 SCO links support 支持音频传输,在这里没用到不选 RFCOMM protocol support 支持无线电连接协议, RFCOMM 是基于 ETSI-07.10 规范的串行线仿真协议

```
RFCOMM TTY support
支持 Linux 终端设备映射到蓝牙串口端口上
BNEP protocol support
支持蓝牙网络封闭协议
Multicast filter support 支持多点传送过滤
Protocol filter support 支持协议过滤
(2) 在 Bluetooth device drivers 中配置内核所支持
```

(2) 在 Bluetooth device drivers 中配直內核所又持 的硬件接口类型,本文所用的是 USB 接口,因此配置 图 3 所示。

$\diamond$	EC1	USB driver
[]	S	CO (voice) support
<>	HCI	UART driver
<>	HCI	BCM203x USB driver
<>	HCI	BPA10x USB driver
<>	HCI	BlueFRITZ! USB driver
<>	HCI	VHCl (Virtual HCl device) driver

图 3 在 Bluetooth device drivers 配置

(3) 配置完后,运行 make gk 编译内核,在 arch/arm/boot/下生成 gk 内核文件。

#### 3.2 移植 BlueZ 库和工具集

在 BlueZ 的官网上下载 BlueZ 的程序包,主要下载 bluez-libs-2.25.tar.gz、bluez-utils-2.25.tar.gz,前者是蓝牙 设备所用到的库文件,后者是管理蓝牙设备的工具集,二者都是使用蓝牙所必不可少的。

下载完后,分别解压 bluez-libs-2.25.tar.gz, bluezutils-2.25.tar.gz, 进行交叉编译。编译过程会提示缺失某 些库文件, 那是因为有些 Linux 系统不是完全安装的。 解决办法就是缺什么就再装什么, 直到没有缺失为止。

strcpy(hcid.pin\_code, "0000");

hcid.pin\_len = 4;

编译 bluez-utils-2.25 生成几个目录: bin etc man sbin va。最后把这些生成的文件或库移到开发板的根文件系统中。

编写 PIN-HELPER 程序, PIN\_HELPER 是用来指定 读取用户输入的 PIN CODE 的程序,在配置文件 hcid. conf 中默认的配置是 PIN\_HELPER=/usr/bin/bluepin,而 这个 bluepin 是作者 python 写的,在这里没有它的源码, 无法进行交叉编译。因此,需要自己写一个 PIN-HELPER 程序。通过对源码的分析,其验证流程图如图 4 所示。 事实上 hcid 进程在验证 PIN CODE 的时候开了一个 pipe 指向 PIN\_HELPER 进程的标准输出, PIN\_HELPER 进 程向用户询问 PIN CODE,如果用户输入密码 XXXX 就 以"PIN:XXXX"的形式写到标准输出中去,如果用户 reject 就写个"ERR"回去。



图 4 验证码过程

以最简单的方式实现 my-bluepin,即固定地写一个 code 回去,用脚本实现:

#!/bin/sh

Echo "PIN:0000"

把 my-bluepin 移到开发板根目录/usr/bin/下,修改 hcid.conf,使 PIN\_HELPER=/usr/bin/my-bluepin。

### 3.3 启动开发板,检查 USB 蓝牙设备运行状态

实验根文件系统用的是 NFS 文件系统,因此无需烧 写文件系统。只需烧写 UBOOT,利用 UBOOT 下载内 核文件到 MP2530 开发板上。

下载后,系统自动运行内核文件,在进入 shell 命 令提示符后,插上 USB 蓝牙适配器,系统自动加载蓝 牙各个模块协议,运行 dmesg | tail 可以看到: usbcore: registered new driver usb-storage //注册 usb-storag 驱动 Bluetooth: HCI USB driver ver 2.7 // HCI USB 驱动 usbcore: registered new driver hci\_usb //注册 hci\_usb 驱动 然后运行以下命令,启动 hci 设备和 hcid 精灵进程: # hciconfig hci0 up # hcid -f /etc/bluetooth/hcid.conf 运行 hciconfig 查看蓝牙设备状态 # hciconfig -a Type: USB hci0: //蓝牙接口类型 BD Address: 00:18:E4:0B:3E:B3 ACL MTU: 310:10 SCO MTU: 64:8 // 00:18:E4:0B:3E:B3 为本地蓝牙设备地址 UP RUNNING PSCAN ISCAN //运行状态

结果显示 USB 蓝牙设备运行正常,说明 BlueZ 蓝 牙协议栈移植完成,USB 蓝牙设备驱动安装成功。

## 4 蓝牙应用程序开发

BlueZ 协议栈移植成功后,即为蓝牙应用程序的 开发搭好环境,开发人员可以利用 BlueZ 提供的丰富 的函数库进行相关应用程序开发。本例是利用探险家 蓝牙 GPS 模块,通过在 MP2530 开发板蓝牙连接 GPS 模块,接收数据实现时定位。其原理是对模拟串口实 时读取数据。如前所述,在 BlueZ 协议栈里,有一个 RFCOMM 协议层,它位于 L2CAP 协议层和应用层协 议之间,是一个传输层协议。它提供了基于 L2CAP 协议层之上的模拟串口,完成了对 RS232 串口的仿 真。模拟串口是 rfcomm0,应用程序只是对它进行读 取数据。

实验过程如下(以下操作都是在开发板上进行)。

#### 4.1 启动进程

在开发板上插入 USB 蓝牙适配器, 开启 GPS 模块 电源, 启动 hci 设备和 hcid 精灵进程。hcid 是 hci 的守 护进程, hcid 通过读取配置文件 hcid.conf, 指定蓝牙 设备的链接模式(主或从)、链接策略、询问和扫描模 式等。

# hciconfig hci0 up
# hcid -f /etc/bluetooth/hcid.conf

#### 4.2 搜索周围可用蓝牙设备

#hcitool scan
Scanning ...
00:1E:3A:7F:35:14 N95 8G
00:0A:3A:1C:29:B8 VGPS-7OO
VGPS-7OO 就是实验用到的 GPS 模块,00:0A:
3A:1C:29:B8 是它的地址。

#### 4.3 配对连接

让USB蓝牙适配器与GPS模块通过蓝牙进行连接: #rfcomm connet /dev/rfcomm0 00:0A:3A:1C:29:B8 1 & Connected /dev/rfcomm0 to 00:0A:3A:1C:29:B8 on channel 1 Press CTRL-C for hangup

出现以上提示,说明蓝牙适配器与 GPS 模块连接成功,实验结果也正是如此。

#### 4.4 编写模拟串口的读取程序,读取 GPS 数据

因为本例的目的只是想说明通过这个蓝牙与 GPS 模块的连接,可接收到 GPS 数据,同进也为了减小开发 周期,所以这个程序以最简单的方式读取数据:按字符 一个一个把数据读出。

读取 GPS 数据的程序流程图如图 5 所示。 交叉编译 gpsreading.c,命令如下: # arm-linux-gcc gpsreading.c – ogpsreading 运行 gpsreading,读取 GPS 数据,结果如下所示:



#### 图 5 串口读取程序流程

root@ATR: /# ./gpsreading \$GPGGA,142641.510,2425.9718,N,11806.4411,E,0,00,,36.6,M,12.1,M,0000\*4A

\$GPRMC, 142641.510, V, 2425.9718, N, 11806.4411, E, ., 170209, ., N\*73

\$GPVTG, , T, , M, , N, , K, N\*2C

\$GPGGA, 142642.462, 2425.9718, N, 11806.4411, E, 0, 00, , 36.6, M, 12.1, M, 0000\*4D

\$GPRMC, 142642.462, V, 2425.9718, N, 11806.4411, E, ., 170209, ., N\*74

\$GPVIG, , T, , M, , N, , K, N\*2C

\$GPGGA, 142649.454, 2425.9718, N, 11806.4411, E, 0, 00, , 36.6, M, 12.1, M, 0000\*43

\$GP&A,A,I,,,,,,,,,,,,,,\*1E

\$GPGSV, 3, 1, 12, 20, 00, 000, , 10, 00, 000, , 25, 00, 000, , 27, 00, 000, \*79

\$GPGSV, 3, 2, 12, 22, 00, 000, , 07, 00, 000, , 21, 00, 000, , 24, 00, 000, \*79

\$CPGGA, 142650.454, 2425.9718, N, 11806.4411, E, 0, 00, , 36.6, M, 12.1, M, 0000\*4B

\$GPRMC, 142650.454, V, 2425.9718, N, 11806.4411, E, ., 170209, ., N\*72

\$GPVTG, , T, , M, , N, , K, N\*2C

\$GPGGA,142051.402,2425.9718,N,11800.4411,E,0,00,,30.0,M,12.1,M,0000\*4F

一个简单的蓝牙应用程序基本完成,如果想把这个 应用程序做得详细一点,可以对串口读取到的数据进行 细分,提取出当前位置的经度、纬度、高度、时间等, 以人机界面的形式展现。除此之外,还可以开发通过蓝 牙手机拨号上网、车载导航、文件传输以及把开发板作 为服务器提供像 FTP、HTTP 服务等。 BlueZ 成功移植到 MP2530 开发板上,搭建好了嵌入式 蓝牙应用开发平台,并以 GPS 的实现为例阐述了基于 BlueZ 开发蓝牙应用程序的基本过程。随着移动电子设 备的日益发展,BlueZ 的成功移植势必使得具有无线性、 开放性、低功耗等特点的蓝牙得到进一步的广泛应用。 本文中所阐述的 BlueZ 移植方法及 GPS 实现均在实践中 均得到验证。

#### 5 结语

本文在对蓝牙协议栈 BlueZ 分析的基础上,将