**..theApp在主类中定义，代表主类对象**

**..packet\_kind表示当前包的最上层协议**

**..OnTCatch方法 ：抓到包后调用的方法(处理包)**

**..CListView的OnInitialUpdate方法：初始化视图**

**..CInfoView的m\_info成员变量：Info变量的数组**

**..Info类：header成员（指针）：包的基本信息；pkt\_data成员(指针)：包的内容；m\_Head成员（指针）：分析类的指针变量**

**..主类的m\_packInfo变量：CPackInfo类的实例化**

**..CPackInfo类：CEditView类的子类，用于在界面中显示包的详细信息**

**..CString类的Format方法：将其他类型的变量转换为CString类型**

**..pcap\_if\_t为网卡设备类**

**..主类的startCatch用于启动抓包线程开始抓包**

**问题：**

**Q: CListView的getDocument()调用失败，返回NULL，提升：读取访问权限冲突**

**A:**

**Q：字符串内存回收**

**A：delete [] number;**

**Q：vector的push\_back**

**A：当参数的类型为指针时，push\_back并不会复制一份地址指向的内容，而是直接将指针push到vector**

**Q：监听点击事件**

**A：在消息映射中添加”ON\_NOTIFY\_REFLECT(NM\_CLICK, 入口函数)“**

**Q: 界面只显示802.3类型的包**

**A：(m\_kind.Compare((LPCWSTR)"0800") 返回不符合逻辑的值，因为“0800”字符串直接强制类型转换成了LPCWSTR，而这是错误的，需要将字符串类型转换为CString,然后再通过CString默认转换为LPCWSTR**

**Q：十六进制进制打印错误，以%02X的形式打印时，有时会出现FFFFFF\*\*的格式（当第一的\*为F时）**

**A：将char强制转换为unsigned char即可，这是因为char会默认转换为int类型，而符号为会被保留到int中，所以当他以%02X的格式打印时，会被解释成一个很大的整数，因此才会多出FFFFFF**

**Q：如何将主窗口分割为更多的窗口？**

**A：创建CSplitterWnd对象，然后实现MainFrm的OnCreateClient，并再其中调用CSplitterWnd的CreateStatic和CreateView方法**

**Q: 移位运算符的使用**

**A：二进制同十进制一样，低位在右，高位在左**

**<< 代表左移，数值增大**

**>> 代表右移，数值减小**

**Q：CEdit控件自动换行：**

**A：在控件属性中设置**

1\Multiline为true、

2\Auto VScroll为true，

3\Virtical scroll为true

4\Auto HScroll为false

**Q：十六进制与十进制的转换**

1. **char** s[100] = {0};
2. sprintf(s, "%x", 15);
3. printf("%s\n", s); // f
5. **char** str[100] = "f";
6. **int** i = 0;
7. sscanf(str, "%x", &i);
8. printf("%d\n", i); // 15

**Q：MFC的一些函数**

**A：**

|  |  |  |  |
| --- | --- | --- | --- |
| **函数名** | **参数** | **返回** | **功能** |
| **GetDlgItem** | **int 控件ID** | **控件的句柄** | **将相应的控件绑定到相应的控件对象** |
| **::AfxMessageBox** | **CString 显示的内容** | **Int** | **可以在静态函数中调用的消息提示框** |
|  |  |  |  |
|  |  |  |  |

**Q：CListView的一些操作**

**A：**

|  |  |
| --- | --- |
| **GetSafeHwnd** | **获取CListView的句柄** |
| **GetListCtrl** | **返回CListCtrl对象，从而进行CListView窗口界面的修改** |
| **SetWindowLong** | **改变窗口的风格** |
| **CListCtrl的SetExtendedStyle** | **设置扩展风格** |
| **CListCtrl的SetBkColor** | **设置背景颜色** |
| **CListCtrl的SetTextBkColor** | **设置文本背景颜色** |
| **CListCtrl的SetTextColor** | **设置文本颜色** |
| **CListCtrl的InsertColumn** | **插入新的列** |
| **CListCtrl的InsertItem** | **插入行** |
| **CListCtrl的SetItemText** | **设置行的字段的值** |
| **CListCtrl的DeleteAllItem** | **删除全部的行** |

**Q：CEditView的一些操作**

**A：**

|  |  |
| --- | --- |
| **SetWindowText** | **设置窗口显示的字符串** |
| **GetEditCtrl** | **返回一个CEdit对象，用于控制当前的CEditView的属性** |
| **CEdit的SetReadOnly** | **将窗口设置为只读即，用户无法编辑** |

**Q：Edit框的使用**

**A：**

|  |  |
| --- | --- |
| **GetDlgItemText(IDC,存储变量)** | **获取编辑框的内容** |
| **SetDlgItemText** | **设置编辑框的内容** |
| **Create** | **创建窗体** |
|  |  |
|  |  |
|  |  |
|  |  |

**Q：CHECK BOX框的使用**

**A：**

|  |  |
| --- | --- |
| **IsDlgButtonChecked(IDC)** | **当返回为BST\_CHECKED时表示被选择** |
| **SetCheck** | **设置CHECK BOX的点击状态，需要配合CButton类和GetDlgItem使用** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Q：ComboBox下拉选择框的使用**

**A:**

|  |  |  |
| --- | --- | --- |
| **方法名** | **参数** | **功能** |
| **GetCurSel** |  | **返回当前的索引值（当前选中的行）** |
| **AddString** | **CString 需要添加的字符串** | **添加一行到下拉列表** |
| **GetCount** |  | **获取当前已有的行数** |
| **DeleteString** | **UINT nIndex要删除的行** | **删除指定行** |
| **ResetContent** |  | **删除所有行** |
| **InserString** | **Int nIndex 插入的位置,**  **LPCTSTR lpszItem 字符串** | **将行插入指定位置** |
| **SetCurSel** | **int nSelect 行号** | **设置当前的行号** |
| **GetWindowText** | **CString strTemp 存储** | **获取当前行的内容** |
| **GetLBText** | **Int nIndex 行号，**  **CString strTemp 存储** | **获取指定行的内容** |

**Q：WinPcap库的接口：**

**A：**

|  |  |  |
| --- | --- | --- |
| **方法名** | **参数** | **功能** |
| **pcap\_findalldevs\_ex** | **char\*source 指定为PCAP\_SRC\_IF\_STRING ,**  **pcap\_rmtauth\*auth 指定为 0 ,**  **pcap\_if\_t\*\*alldevs 指向网卡列表的指针的地址,**  **char\*errbuf 发生错误的信息保存地址** | **操作电脑的所有设备并保存到alldevs中，因此第3个参数必须指定，失败返回-1** |
| **pcap\_freealldevs** | **pcap\_if\_t\* 指向设备列表的指针** | **释放设备列表的内存等，** |
| **pcap\_open**  remote-ext.h库中声明 | **const char\*source 设备名，指定为pcap\_if\_t的name变量 ,**  **int snaplen 一般指定为65536表示抓包的最大端口?,**  **int flags 一般指定为1表示混杂模式抓住所有的经过网卡的包,**  **int read\_timeout 一般指定为1000 读取超时？,**  **pcap\_rmtauth\*auth 一般指定为NULL,**  **char\*errbuf发生错误的信息保存地址** | **开启设备，失败时返回NULL，成功时返回pcap\_t的指针** |
| **pcap\_dump\_open** | **pcap\_t\* 已开启的网卡,**  **const char\* 存储路径?,** | **打开tmp.pcap接收数据包,返回pcap\_dumper\_t\*** |
| **pcap\_dump** | **u\_char\*,**  **const pcap\_pkthdr\*,**  **const u\_char\*,** | **未知，有待测试** |
| **pcap\_loop** | **pcap\_t\* 已开启的网卡类,**  **int 一般指定为0,**  **pcap\_handler 一个函数指针,函数定义时必须为：static void MyPcapHandler(uchar\*, const pcap\_pkthdr\*, const u\_char\*) 下面有介绍,**  **u\_char\* 一般为pcap\_dumper\_t\*的强制类型转换(unsigned char\*)(pcap\_dumper\_t\*),** | **循环抓取数据包，抓到后调用pcap\_handler** |
| **static void pcap\_handler** | **u\_char\* 未知,**  **const pcap\_pkthdr\* 包的一些信息,**  **const u\_char\* 抓到的包,** | **自定义的，但要符号接口的数据包处理函数，一般在函数中启动新的线程来处理数据包** |
| **void pcap\_breakloop** | **pcap\_t\* 已开启的网卡设备** | **停止抓包** |
| **Void pcap\_dump\_close** | **Pcap\_dumper\_t\*** | **Dump相关，有待测试** |
| **pcap\_close** | **pcap\_t\* 已开启的网卡** | **关闭网卡** |

**Q：WinPcap的类**

**A：**

|  |  |
| --- | --- |
| **类名** | **说明** |
| **pcap\_t** | **已经开启了的网卡设备类，用于启动抓包等操作** |
| **pcap\_if\_t\*** | **指向网卡列表的指针,用于开启设备** |
| **pcap\_dumper\_t** | **Dump相关，有待测试** |
| **pcap\_pkthdr** | **抓到的包的一些基本信息类，如时间，大小等** |
|  |  |
|  |  |
|  |  |

**Q：WinPcap的pcap\_if\_t类的变量：**

**A：**

|  |  |
| --- | --- |
| **变量名** | **功能** |
| **next** | **返回下一个设备pcap\_if\_t\*** |
| **description** | **获取设备的描述** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Q：引入WinPcap**

**A：项目-》属性-》VC++目录项，将需要包含和头文件的路径加入到“包含目录”以及“库目录”中，同时将xxx.lib文件放入到项目根目录的“Lib/x64”目录下**

**Q:使用到的线程相关**

**A：**

|  |  |  |
| --- | --- | --- |
| **::AfxBeginThread** | **参数一：AFX\_THREADPROC pfnThreadProc 线程的入口函数，定义时必须为UINT MyThreadFunction(LPVOID pParam)的全局或静态函数**  **参数二：LPVOID pParam传入入口函数的参数,一般是一个结构体**  **参数3、4、5分别指定线程的优先级、堆栈大小、创建标识、安全属性，含义同用户界面线程。** | **启动一个新的线程（有界面线程和工作线程两种线程），注意不要在一个MFC程序中使用\_beginthreadex()或CreateThread()创建新线程** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |