

VC5 的窗口静态分割

李 刚 (大庆石油学院计算机系 151400)

在一个框架(Frame)中,经常需要将窗口静态分割为多个不同类型的布局特殊的视图(View)。本文介绍如何实现分割以及各视图之间的数据交换。例程将窗口分为上一下二的三个视图。

1. 创建分割窗口

静态分割窗口的一般步骤是:在父框架中申明 CSplitterWnd 类型的成员变量;重置父框架的 CMainFrame::OnCreateClient 成员函数;在 OnCreateClient 函数中调用 CSplitterWnd::CreateStatic 函数。

如果在向导的第四步选取了 Use Split Window 选项,向导会限制你选择视图的基类型。对单文档应用而言,在 CMainFrame 类中实现分割,对多文档在 CChildFrame 类中实现分割。以下以单文档为例,应用程序名为 Demo。实现上一下二分割的思路是设两个 CSplitterWnd 类型的变量, m_wndSplitter1 以 m_wndSplitter 的下窗格(pane)为父窗口进行分割。

变量申明:

```
public:
```

```
CSplitterWnd m_wndSplitter
```

```
CSplitterWnd m_wndSplitter1;
```

函数实现:

```
BOOL CMainFrame::OnCreateClient(LPCREATESTRUCT/* lpcs*/, CCreateContext* pContext) + { // 将主窗口分为二行一列
```

```
if (! m_wndSplitter.CreateStatic(this//以主窗口为父窗口
```

```
,2,1)) return FALSE;
```

```
CRect mRect;
```

```
m_wndSplitter.GetWindowRect(mRect);
```

```
//创建上半部分视图
```

```
if (! m_wndSplitter.CreateView(0,0,RUNTIME_CLASS(CDemoView),
```

```
//视图类可以是自定义的由 CView 或 CEditView 派生的视图,
```

```
//或从 CFormView 派生的拥有模板资源的视图,注意申明头文件
```

```
CSize(0,0),pContext)) return FALSE;
```

```
//将 m_wndSplitter 的下半部分一分为二
```

```
if (! m_wndSplitter1.CreateStatic(&m_wndSplitter, //m_wndSplitter 作为其父窗口 1,2, WS_CHILD| WS_VISIBLE| WS_BORDER, m_wndSplitter.IdFromRowCol(1,0))) return FALSE;
```

```
if (! m_wndSplitter1.CreateView(0,0, RUNTIME_CLASS(CDemoView1), CSize(0,0), pContext)) return FALSE;
```

```
if (! m_wndSplitter1.CreateView(0,1, RUNTIME_CLASS(CDemoView2), CSize(0,0), pContext)) return FALSE;
```

```
return (TRUE);
```

```
}
```

在 OnCreateClient 函数中调用 CSplitterWnd::RecalcLayout 函数控制各视图的显示尺寸没有什么效果,可以增加如下函数供运行过程中调用:

```
void CMainFrame::Layout()
```

```
{ CRect mRect;
```

```
m_wndSplitter.GetWindowRect(mRect); //取得了客户区矩形
```

```
m_wndSplitter.SetRowInfo(0,mRect.Height()/3,0); //设置了上半部分的高度
```

```
//第三个参数一般设为0,尤其是当此视图由 CForm View 派生时
```

```
m_wndSplitter1.SetColumnInfo(0,mRect.Width()/3,0); //下半部分左边的宽度
```

```
m_wndSplitter1.RecalcLayout();
```

```
m_wndSplitter1.RecalcLayout();
```

```
}
```

2. 各个视图的数据交换

(1) 文档作为视图间沟通的桥梁,当一个视图改变了文档的数据时,可以调用文档的 UpdateAllViews 函数更新其余视图。例如:

```
void CDemoView1::OnLButtonDown(UINT nFlags, CPoint point)
```

```

{
    CTranDoc * pDoc = (CTranDoc *)GetDocument();
    //更新文档数据的语句
    pDoc -> UpdateAllViews(this);
    CFormView::OnLButtonDown(nFlags, point);
}

```

(2)根据创建各视图时所确定的关系在一个视图里面获取其他视图的指针。例如: void CDragView1::OnRButtonDown(UINT nFlags, CPoint point)

```

{
    CMainFrame * pFrame = (CMainFrame *) AfxGetMainWnd();
    ASSERT(pFrame -> IsKindOf(RUNTIME_CLASS

```

```

(CMainFrame)));
    CDemoView * pDemoView = (CDemoView *)
        pFrame -> m_wndSplitter.GetPane(0,0);
    ASSERT(pDemoView -> IsKindOf(RUNTIME_CLASS(CTargeFile)));
    CDemoView2 * pDemoView2 = (CDemoView2 *)
        pFrame -> m_wndSplitter1.GetPane(0,1);
    ASSERT(pDemoView2 -> IsKindOf(RUNTIME_CLASS(CSourceFile)));
    //利用所得二指针进行数据交换
    CFormView::OnRButtonDown(nFlags, point);
}

```

(来稿时间:1999年4月)