

VB 中屏幕共享系统的设计与实现^①

Design and Implementation of Screen Sharing System in Visual Basic

张宗飞 (台州职业技术学院 计算机工程系 浙江 台州 318000)

摘要: 分析了常用的几种屏幕共享技术后,提出了一种运用 VB 和 SQL Server 技术设计屏幕共享系统的方案,实现了从主机到客户机的屏幕共享及客户机可回放共享过程的功能。详细分析了屏幕图像的截取、保存、显示及回放技术的实现方法。系统构架简单、扩展性好,适合进行二次开发以满足各种具体应用。

关键词: 系统 屏幕共享 图像 VB SQL Server

1 引言

屏幕共享是指网络环境中的一台主机(共享机)将其屏幕的界面内容在允许的时间延迟内重现于客户机屏幕,从而提供给客户机用户一种如坐在主机旁一样的虚拟界面环境。屏幕共享是数据共享中的一种重要应用,已在视频会议、远程教学、远程监控和股票分析系统中得到了很好的应用。目前,常见的基于屏幕共享技术的产品主要有 Windows 2000 的远程终端服务、NetMeeting、PCAnywhere、VNC 等。

最早由苹果公司提出的图形用户界面 GUI (Graphic User Interface) 为屏幕共享技术的产生和发展奠定了基础。此后随着网络技术和多媒体技术的发展,屏幕共享逐渐成为网络通信领域的关键技术之一。常见的屏幕共享技术通常采用以下两种方式来实现:

(1) 利用操作系统底层的 GUI 矢量指令实现。例如 Windows 2000 提供的远程终端服务,用户通过使用客户端软件登陆到服务器,同时服务器端为每个客户端开辟一个虚拟屏幕(Virtual Desktop),并将该用户的操作结果显示重定向到他所在的屏幕,这些显示操作首先被分解成若干 Windows GDI(Graphic Device Interface)指令,然后传输到客户端进行解码显示。

(2) 利用屏幕拷贝和图像压缩技术实现。屏幕显示的内容不是通过拆解成具体的绘图命令来实现,而是通过先拷贝屏幕图像,然后将拷贝得到的屏幕图像

进行压缩后传送到客户端,最后由客户端解码并显示。

本文所设计的屏幕共享系统是采用基于屏幕拷贝和图像压缩的方式,以 Visual Basic 为开发工具、SQL Server 为后台数据库设计并开发实现的。

2 设计方案

2.1 屏幕图像处理

在 Windows 中产生的任何资源,如图标、光标、窗口、加载到内存运行中的应用程序,操作系统每产生一个这样的资源时,都要将它放入相应的内存,并为其指定一个唯一的标志号,这个标志号即为该资源的句柄,操作系统通过该句柄来管理和操作这一资源。在本系统中,当主机屏幕开始共享时,此时主机屏幕就是一个窗口,该窗口是通过操作系统在内存中指定一个句柄来管理的,因此只要得到此句柄,便可得到主机屏幕所对应的位图图像。本系统使用 GetDesktopWindow() 函数来获取主机屏幕的句柄。

2.2 图像压缩

实现图像压缩的方法很多,考虑到屏幕图像大多具有简单的 GUI 界面,纹理和色彩具有单一、规则的特点,同时出于对较低 CPU 占用率的要求,在选择屏幕图像压缩算法时,放弃使用一些压缩效率比较高但

^① 台州职业技术学院校级科研项目(2006YB13)

CPU 占用率高的压缩算法,而采用 JPEG 标准,对主机屏幕图像截取后所生成的 BMP 位图文件进行压缩,形成 JPG 文件。

2.3 网络通信

屏幕共享中的网络通信发生在主机与客户机之间,是一对多的通信模式。本文所设计的屏幕共享通过 UDP 下的组播模式实现通信。虽然 UDP 下的广播模式也能实现主机向客户机发送信息的能力,但效率不高,因为网络中不相关的客户机也同时要接收此信息,而且广播不能跨路由。组播模式不但能跨路由传输,而且方便用户权限的管理,没有权限的用户不能加入该组。本系统中当截取主机屏幕图像并保存到 SQL Server 中后,便组播消息通知指定的客户机到 SQL Server 中读取图像数据。

3 实现流程

本系统主要由服务器、主机(共享机)和客户机三部分组成,服务器中运行 SQL Server 2000 数据库系统。系统程序包含两个模块:主机程序与服务器程序编写在同一模块中,即服务器模块;客户机程序独立编写为客户机模块。

本系统是在 VB 环境中开发的,运用 VB 通信技术与数据库技术实现屏幕共享,系统实现的基本流程是:

- (1) 按年月日时分秒毫秒截取主机屏幕图像;
- (2) 将截取的屏幕图像压缩后写到 SQL Server 中;
- (3) 组播消息通知客户机到 SQL Server 中读取保存的图像数据;
- (4) 读取图像数据到客户机并保存到文件里,然后解压并显示图像文件。

系统实现流程图如图 1、图 2 所示。

4 关键技术的实现

4.1 屏幕图像截取

本系统需要实现屏幕共享功能,因此主机屏幕图像的截取是整个系统实现的第一步。在系统开发过程中,采用如下方法实现主机屏幕图像的截取。

- (1) 设置目标设备场景的 ID 号;

(2) 用 CreateCompatibleDC() 函数创建一个与设备场景号一致的内存设备场景;

(3) 再用 CreateCompatibleBimap() 函数创建一幅与设备有关的位图,与指定的设备场景兼容;

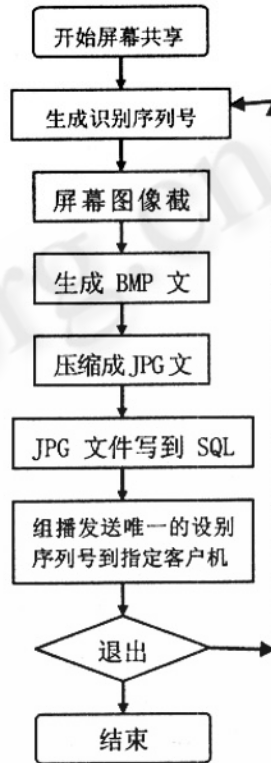


图1 服务器端流程图

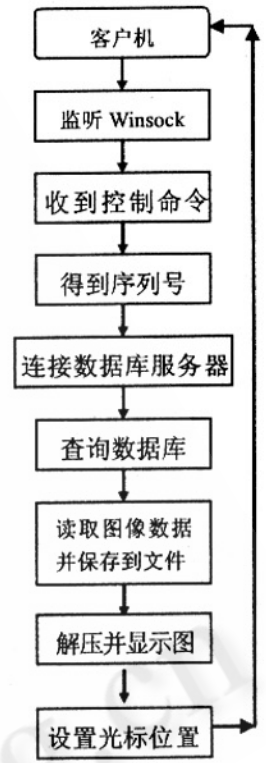


图2 客户端流程图

(4) 把与设备有关的位图装入内存场景中;

(5) 利用 BitBlt() 函数进行图像块的传送,将屏幕图像传送到新建的 BMP 文件中。

本功能模块的主要代码如下:

```

Public Function CaptureWindow ( ByVal hWndSrc As Long, ByVal Client As Boolean, ByVal LeftSrc As Long, ByVal TopSrc As Long, ByVal WidthSrc As Long, ByVal HeightSrc As Long)

```

```

    If Client Then 设置目标设备场景号

```

```

        hDCSrc = GetDC (hWndSrc)

```

```

    Else

```

```

        hDCSrc = GetWindowDC (hWndSrc)

```

```

    End If

```

```

        hDCMemory = CreateCompatibleDC (hDCSrc)

```

创建一个与 hDCSrc 设备场景号一致的内存设备场景

```

        hBmp = CreateCompatibleBimap (hDCSrc, Width-

```

Src, HeightSrc) 创建一幅与设备有关位图,它与指定的设备场景兼容

```
hBmpPrev = SelectObject(hDCMemory, hBmp)
```

选定目标装入场景

```
RasterCapsScrn = GetDeviceCaps(hDCSrc, RAS-  
TERCAPS)
```

```
HasPaletteScrn = RasterCapsScrn And RC_PAL-  
LETTE
```

```
PaletteSizeScrn = GetDeviceCaps(hDCSrc, SIZE-  
PALETTE)
```

```
If HasPaletteScrn And (PaletteSizeScrn = 256)
```

Then 设置位图的格式

```
LogPal.palVersion = &H300
```

```
LogPal.palNumEntries = 256
```

```
r = GetSystemPaletteEntries(hDCSrc, 0, 256,
```

```
LogPal.palPalEntry(0))
```

```
hPal = CreatePalette(LogPal)
```

```
hPalPrev = SelectPalette(hDCMemory, hPal,
```

```
0)
```

```
r = RealizePalette(hDCMemory)
```

```
End If
```

```
r = BiBit(hDCMemory, 0, 0, WidthSrc,
```

```
HeightSrc, hDCSrc, LeftSrc, TopSrc, bSrcCopy)
```

将屏幕图像信息传送到内存位图中

```
hBmp = SelectObject(hDCMemory, hBmpPrev)
```

```
If HasPaletteScrn And (PaletteSizeScrn =
```

```
256) Then
```

```
hPal = SelectPalette(hDCMemory, hPalPrev, 0)
```

为内存位图选择调色板

```
End If
```

```
r = DeleteDC(hDCMemory)
```

```
r = ReleaseDC(hWndSrc, hDCSrc)
```

```
Set FMAI.ImageMan1.VBPicture = CreateBimap-  
Picture
```

(hBmp, hPal) 按照规定的格式创建一幅与设备有关的位图并将位置直接传送给 ImageMan 控件

```
End Function
```

4.2 图像保存

本系统是利用数据库作为中间保存方式来实现

图像数据传输的,因此当截取到主机屏幕图像并压缩后,应保存到 SQL Server 数据库中,等待客户机来读取。在具体实现时,先在 SQL Server 的数据库文件 R1 中新增一条空记录并指定序列号 ID,然后建立一个二进制流对象,将压缩后的屏幕图像文件加载到新建的流对象中,最后将流对象中的图像数据写到 R1 中的 ID 记录中。

本功能模块的主要代码如下:

```
DB.BeginTrans 开始一个事务
```

R1.AddNew 增加一条数据库记录,将屏幕截取的图像增加到数据库中去

```
R1! 序号 = S2 S2 为生成的唯一序列号 ID
```

Set Pstream = New ADODB.Stream '新建一个流对象

Pstream.Type = adTypeBinary 指定流为二进制类型

```
Pstream.Open 将数据获取到 Stream 对象中
```

Pstream.LoadFromFile(s1) 将选择的文件加载到打开的流对象中,s1 为采集到的桌面图像数据

```
R1! 图像 = Pstream.Read 从流对象中读取数据
```

```
Pstream.Close
```

```
R1! X = P1.X
```

```
R1! Y = P1.Y
```

```
R1.Update
```

DB.CommitTrans 完成一个事务

4.3 图像显示

当把截取的屏幕图像保存到数据库服务器中后,服务器就等待客户机来读取图像数据。服务器是如何把要显示的图像信息告诉客户机呢,什么时候告诉呢。系统通过在图像保存后向客户机组播一个控制命令及序列号 ID 参数。

```
FMAI.Winsock1.SendData "新图@@~" + 唯一  
序列号 ID
```

向客户机发送了一个唯一序列号 ID 后,客户机通过数据库引擎到数据库服务器中查询指定序列号 ID 的数据库记录,找到后将该记录数据写到新建的二进制流对象中,然后将流对象中的数据保存到本机硬盘,生成图像文件,解压后通过 VB 的 Image 控件显示图像。

本功能模块的主要代码如下:

```

S2 = Mid $(S1, 1 + 4)
R1. Close
R1. CursorLocation = adUseClient
R1. Open "Select * From 图像 where 序号 = "
+ S2 + "'", DB, adOpenKeyset, adLockOptimistic 到
数据库服务器中去查询指定序列号的数据库记录
Dim Pstream As ADODB. Stream
Dim X As Long
Dim Y As Long
Set Pstream = New ADODB. Stream 新建一个流
对象
Pstream. Type = adTypeBinary 指定流为二进
制类型
Pstream. Open
Pstream. Write R1! 图像 把图像数据写到二
进制流对象中
S1 = App. Path + "\Sys\temp01. tmp"
Pstream. SaveToFile S1, adSaveCreateOver-
Write 将 Stream 对象中的数据保存到硬盘
Pstream. Close
X = R1! X 获取光标位置 X 向
Y = R1! Y 获取光标位置 Y 向
SetCursorPos X, Y 设置本机的光标位置
Form2. Image1. Picture = LoadPicture(S1) 显
示图像
Form2. Image1. Refresh 刷新显示

```

4.4 图像回放

上述的屏幕共享过程是由主机通过网络控制客户机来完成整个共享过程的。本系统的另一功能是当共享过程完成后,如果某客户机需要重新显示共享内容,可以回放图像。具体实现方法是到 SQL Server 的数据库文件 R1 中读取记录,从首记录开始读取记录并保存到本机硬盘,形成图像文件,然后解压显示,当读完最后一条记录并显示后,即完成了整个回放过程。

本功能模块的主要代码如下:

```

Dim X, Y As Long
StopBool = False
Do While R1. EOF = False
If StopBool = False Then

```

```

Dim Pstream As ADODB. Stream
Set Pstream = New ADODB. Stream
Pstream. Type = adTypeBinary 指定流是二进
制类型
Pstream. Open
Pstream. Write R1! 图像 写图像到文件
S1 = App. Path + "\Sys\temp01. bmp"
Pstream. SaveToFile S1, adSaveCreateOverWrite
Pstream. Close
Form1. Image1. VBPicture = LoadPicture(S1)
Form1. Image1. Refresh
X = R1! X
Y = R1! Y
R1. MoveNext
SetCursorPos X, Y 设置光标位置
Sleep VI 延时处理
End If
DoEvents 控制权转让
Loop
MsgBox "回放完成!"

```

5 系统特点

系统采用 SQL Server 数据库来保存共享屏幕的图像数据,因此系统不仅实现了从主机到客户机的屏幕共享,而且当共享过程结束后,客户机还可以重新从数据库中读取保存的图像数据,回放共享过程。系统中屏幕图像的压缩与解压缩是调用独立的压缩算法来实现的,便于系统的修改与扩展。

6 结束语

屏幕共享的应用越来越广泛,本文的贡献在于提出了一种结构简单、设计容易的实现方案。本文所设计的系统已在本院部分教师的教学课件中使用,效果良好。本系统也存在不完善之处,系统的屏幕图像是采用定时截取机制,而在实际应用中,屏幕图像可能在相当一段时间内没有更新,因此采用定时截取屏幕图像机制将会对本地资源与网络带宽造成浪费,这是系统今后需要改善的地方。本系统的下一步目标是在此基础上开发客户机对主机的控制功能,实现在必要时客户机也可以操作主机。 (下转第 111 页)

参考文献

- 1 郭代,晏坚,王文江. 新型屏幕共享系统设计与实现[J]. 计算机应用研究,2007,24(6):299—301.
- 2 张丽萍,喻占武,肖进胜,等. 一种基于拷屏和压缩技术的屏幕共享方案[J]. 中国图象图形学报,2003,8(A)(9):1095—1099.
- 3 肖任重,申瑞民. 视频会议教学系统中屏幕共享工具的设计与实现[J]. 计算机仿真,2004,21(1):136—138.
- 4 柳迎华,李定主. 基于组播的屏幕共享技术[J]. 电脑开发与应用,2003,16(6):35—36.
- 5 马绍良,李凤保,谢阅. 用 ODBC 实现 SQL Server 2000 在 VB 中的应用[J]. 计算机应用,2003,23(1):115—118.
- 6 古凌岚. 用 ADO 技术实现数据库图像字段的存取[J]. 计算机工程与设计,2004,25(8):1388—1389.
- 7 李迎春,朱诗兵. Visual Basic 6.0 网络编程[M]. 北京:希望电子出版社,2001.
- 8 高春艳,刘彬彬. Visual Basic 控件参考大全[M]. 北京:人民邮电出版社,2006.
- 9 刘韬. Visual Basic 6.0 数据库系统开发实例导航[M]. 北京:人民邮电出版社,2004.
- 10 (美)Dan Fox,骆万文等译. 深入学习:Visual Basic 6 开发与实例[M]. 北京:电子工业出版社,2000.