

## 7.3 RichTextBox 控件

RichTextBox 控件是 VB 提供的一个功能强大的文本框控件，不仅具有标准文本框的所有功能，同时还提供标准文本框控件不具有的更高级的功能。该控件能指定文本框中显示的文本格式，支持多字体、多颜色、设置左右边界和符号列表等许多功能。

RichTextBox 是一个外部控件。如果要使用该控件，需要在菜单栏上选择“工程→部件”选项，在弹出的“部件”对话框中选择  Microsoft Rich Textbox Control 6.0 (SI) 选项，单击“确定”按钮后，在工具箱中显示 RichTextBox 控件 

### 1. RichTextBox 控件的常用属性

RichTextBox 提供的一些属性，可把选取的文本改为粗体或斜体、改变字体颜色、创建上标和下标、设置左右缩进和悬挂式缩进等段落格式。

例如：

```
RichTextBox1.SelBold = True  
RichTextBox1.SelFontSize = 20
```

#### (1) 选取文本的字体属性

##### ① SelFontName 属性

返回或设置在 RichTextBox 控件中用于显示当前选定的文本、或用于显示刚从插入点所输入字符的字体，在设计时无效。

语法格式：`object.SelFontName [= string]`

说明：

- `object` 是对象表达式，其值是“应用于”列表中的一个对象。
- `String` 是字符串表达式，标识系统中已安装的一种字体。

如果所选定的文本包含有不同的字体，则 `SelFontName` 属性返回 Null 值。

##### ② SelBold、SelItalic、SelStrikethru、SelUnderline 属性

设置 RichTextBox 控件中选定文本的字体样式。字体样式包括粗体，斜体，删除线和下划线格式。这些属性在运行时有效。

语法格式：

```
object.SelBold [= value]
```

```
object.SelItalic [= value]
```

```
object.SelStrikethru [= value]
```

```
object.SelUnderline [= value]
```

说明：

- `object` 是对象表达式，其值是“应用于”列表中的一个对象。
- `value`：布尔表达式或常数，按照“设置值”中的描述决定字体样式，`value` 的设置值见表 7-3。

表 7-3 value 的设置值

设置值	描述
Null	无值。该选择或在插入点之后的字符含有多种适当字体样式的混合。
True	所选的所有字符或插入点之后的字符具有适当的字符样式。
False	(缺省) 所选的所有字符或插入点之后的字符不具有适当的字符样式。

这些属性的行为类似 Font 对象的 Bold、Italic、Strikethru 和 Underline 属性。RichTextBox 控件的 Font 属性给该控件中所有文本设置字体样式。但是，为了给选定的文本或给插入点

处输入的字符设置字体样式，应使用这些属性。要访问这些属性，可在应用程序中用一些可以分别切换这些属性的按钮，或创建工具按钮。

程序运行中读取这些属性时，为了区分 Null 和 False 的值，应用带有 If...Then...Else 语句和 IsNull 函数。例如：

```
If IsNull (RichTextBox1.SelBold) = True Then  
' 选择混合时运行的代码。  
ElseIf RichTextBox1.SelBold = False Then  
' 选择不是粗体时运行的代码。  
...  
End If
```

(2) 取文本的段落属性

①SelHangingIndent、SelIndent、SelRightIndent 属性

分别用于设置 RichTextBox 控件中段落的页边距值，该控件包括当前选定或在当前插入点添加。这些属性在运行时有效。

语法格式：

```
object.SelHangingIndent [= integer]  
object.SelIndent [= integer]  
object.SelRightIndent [= integer]
```

说明：

- object 是对象表达式，其值是“应用于”列表中的一个对象。
- Integer 是整数，决定缩放量。这些属性采用 RichTextBox 控件所在 Form 对象的刻度模式单位。

对于受影响的段落，SelIndent 属性指定 RichTextBox 控件的左边缘和选定文本的左边缘之间的距离。SelRightIndent 属性指定 RichTextBox 控件的右边缘和选定文本的右边缘之间的距离。SelHangingIndent 属性指定选取段落中文本首行左边缘与本段下面各行左边缘之间的距离。若选取跨越了具有不同页边距的多个段落，则这些属性返回零。

②SelAlignment 属性

设置 RichTextBox 控件中段落的对齐方式。该属性运行时有效。

语法格式：*object.SelAlignment* [= value]

说明：

- object 是对象表达式，其值是“应用于”列表中的一个对象。
- value 是整数或常数，按照表 7-4 的描述确定段落的对齐方式。

表 7-4 value 的设置值

常数	值	描述
	Null	无值。当前的选择跨越了几个具有不同对齐方式的段落
RtfLeft	0	(缺省) 左边。段落沿左边距对齐
RtfRight	1	右边。段落沿右边距对齐
rtfCenter	2	居中。段落居中对齐

SelAlignment 属性决定当前选定段落的对齐方式。

运行中，为了区分 Null 和 0 的值，应使用 If...Then...Else 语句和 IsNull 函数。

例如：

```
If IsNull (RichTextBox1.SelAlignment) = True Then ' 选择混合时运行的代码。  
...  
ElseIf RichTextBox1.SelAlignment = 0 Then ' 选择左对齐时运行的代码。  
...
```

End If

### (3) 上下标属性 (SelCharOffset)

设置 RichTextBox 控件中的文本是正常状态，还是作为上标出现在基线之上，或作为下标出现在基线之下。该属性在运行时有效。

语法格式：object.SelCharOffset [= offset]

说明：

- object 是对象表达式，其值是“应用于”列表中的一个对象。
- Offset 是整数，按照表 7-5 描述的方式指定当前选定字符偏离文本基线的距离。

表 7-5 offset 的设置值

设置值	描述
Null	无值。选择的字符含有不同偏移量。
0	(缺省) 正常状态。这些字符都出现在正常文本的基线上。
Positive integer	上标。这些字符出现在基线之上，偏移量由指定的缙数决定。
Negative integer	下标。这些字符出现在基线之下，偏移量由指定的缙数决定。

运行中读取该属性时，可在 If...Then...Else 语句中用 IsNull 函数区分 Null 和 0。例如：

```

If IsNull(RichTextBox1.SelCharOffset) = True Then '选择混合时运行的代码
...
ElseIf RichTextBox1.SelCharOffset = 0 Then '选择均在基线上时运行的代码
...
End If

```

## 2. RichTextBox 控件的常用方法与应用

RichTextBox 控件能以两种形式打开和保存文件：rtf 格式和普通 ASCII 文本格式。

可以用控件的方法 (LoadFile 和 SaveFile) 直接读写文件，或与 Visual Basic 文件输入/输出语句联结的、诸如 SelRTF 和 TextRTF 之类的控件属性打开和保存文件。

例如：

```

RichTextBox1.LoadFile CommonDialog1.FileName
RichTextBox1.SaveFile CommonDialog1.FileName

```

(1) LoadFile 方法。LoadFile 方法向 RichTextBox 控件加载一个 .rtf 文件或文本文件。该方法不支持命名的参数。

语法格式：object.LoadFile pathname, filetype

说明：

- object 是对象表达式，其值是“应用于”列表中的一个对象。
- Pathname 是字符串表达式，定义加载控件的文件路径和文件名。
- filetype 是整数或常数，可按表 7-6 中的描述确定装入文件的类型。

表 7-6 filetype 的设置值

常数	值	描述
RtfRTF	0	(缺省) RTF。被加载的文件必须是一个合法的 .rtf 文件
RtfText	1	文本。RichTextBox 控件可加载任一文本文件

当用 LoadFile 方法加载一个文件时，加载文件的内容将取代 RichTextBox 控件的全部内容。这将使 Text 和 RTFText 属性的值改变。

也可用 Visual Basic 的 Input 函数和 RichTextBox 控件的 TextRTF、SelRTF 属性读取 .rtf 文件。例如，可按如下方式加载一个 .rtf 文件的内容到 RichTextBox 控件中：

```

Open "mytext.rtf" For Input As #1
RichTextBox1.TextRTF = Strconv(InputB$(LOF(1), 1), vbUnicode)

```

(2) SaveFile 方法。把 RichTextBox 控件的内容存入文件。

语法格式: object.SaveFile(pathname, filetype)

说明:

- object 是对象表达式, 其值是“应用于”列表中的一个对象。
- Pathname 是字符串表达式, 定义保存控件内容文件的路径和文件名。
- Filetype 是可选项, 整数或常数, 按照表 7-7 中的描述确定加载文件的类型。

表 7-7 filetype 的设置值

常数	值	描述
RtfRTF	0	(缺省) RTF。RichTextBox 控件把它的内容存为一个 .rtf 文件。
rtfText	1	文本。RichTextBox 控件把它的内容存为一个文本文件。

可用 Visual Basic 的 Write 函数和 RichTextBox 控件的 TextRTF 属性和 SelRTF 属性写 .rtf 文件。例如, 可按以下方式把 RichTextBox 控件中突出显示的内容存为 .rtf 文件。

```
Open "mytext.rtf" For Output As 1  
Print #1, RichTextBox1.SelRTF
```

## 7.4 其他文件处理方法

### 7.4.1 文件操作概述

计算机用内存暂存数据, 而用磁盘等外部存储器长久地保存数据。计算机一般采用文件形式保存数据。

#### 1. 文件的分类

根据计算机访问文件的方式, 可以将文件分为三种类型:

(1) 顺序文件: 普通的正文文件, 读写顺序文件时, 每次读写一行, 每行的长度可以根据需要变化。适用于读写在连续块中的文本文件。对于文本文件的读写操作, 建议采用前面介绍的 FSO 对象提供的方法来实现。

(2) 随机文件: 可以按任意顺序读写的文件(随机读写)。每一行(每一个记录)的长度必须相同。适用于读写有固定长度记录结构的文本文件, 或者二进制文件。

(3) 二进制文件: 字节的集合, 数据不按某种特定的方式进行组织, 允许按所需的任何方式组织和访问。这类文件的灵活性最大, 但程序设计时的工作量和难度也最大。

#### 2. 文件处理的概念

文件处理是计算机中非常重要的概念, Visual Basic 的文件处理主要包括:

(1) 打开或建立文件: 文件必须先打开或建立后才能使用。操作时, 如果指定的文件已存在, 则打开文件; 如果指定的文件不存在, 则建立文件。

(2) 文件的读写操作: 在打开(或建立)的文件上执行所要求的输入输出操作。其中, 把数据传输到外部设备(如磁盘)并作为文件存放的操作称为写数据; 把数据文件中的数据传送到内存中的操作称为读数据。一般来说, 在内存与外设的数据传输中, 由内存传输到外设称为输出或写操作, 而由外设传输到内存称为输入或读操作。

(3) 关闭文件: 文件操作完成后, 应及时关闭文件, 保证文件数据的安全及后续操作。

#### 3. 文件的打开与关闭

(1) 打开文件。进行读写文件等操作前, 必须先打开文件; 操作完成后, 应及时关闭文件。Visual Basic 用 Open 语句打开文件。

语法格式: Open<文件名>[For<模式>][Access][lock]As[#]<文件号>[Len=<记录长度>]

说明:

(1) 语句中各参数的含义见表 7-8。

表 7-8 Open 语句各参数的含义

参数	功能与描述
文件名	必选项。字符串表达式, 文件名可包括目录、文件夹及驱动器名
<模式>	可选项, 指定文件打开方式, 有 Append、Binary、Input、Output 或 Random 方式。如果未指定方式, 则以 Random 方式打开文件。
Access (存取)	可选项, 关键字, 设置打开文件的操作属性, 包括 Read、Write 等。
Lock	可选项, 关键字, 说明能够对所打开文件进行的操作, 包括 Shared、Lock、Read、Lock Write 和 Lock Read Write 等操作。
文件号	必选项, 有效的文件号范围在 1 到 511 之间。用 FreeFile 函数可得到下一个可用的文件号
文件长度	可选项, 小于或等于 32,767 (字节) 的一个数。对于用随机访问方式打开的文件, 该值是记录长度。对于顺序文件, 该值是缓冲区字符数

(2)使用 Open 语句时, 如果指定的文件在磁盘上不存在, 且以 Append、Binary、Output 或 Random 模式打开的, 则按指定的文件名创建一个新文件。

(3) <模式>参数取值的含义见表 7-9。

表 7-9 Open 语句打开文件的模式

模式	意义
Random	随机读写模式, 是 Open 语句的默认模式
Binary	二进制读写模式, 可以用 Put 和 Get 语句读写文件中的任意字节位置的数据
Input	顺序输入模式 (即以顺序模式从文件中读取数据)
Output	顺序输出模式 (即以顺序模式向文件中写入数据)
Append	顺序输出模式。它将文件指针设置在文件的结尾, 所要写入的内容就添加在文件原有内容之后, Print#或 Write#语句可以用于这种操作

(4)若文件已由其它进程打开, 且没有指定的访问类型, 则 Open 操作失败, 且发生错误。

(2) 关闭文件。关闭文件用 Close 语句。

语法格式: Close [<文件号列表>]

说明:

(1) 语句的可选参数<文件号列表>是一个或多个文件号的列表。

(2) 若语句中不带参数, 则关闭工程中所有打开的文件。Close 语句执行后, 文件号与文件之间的联系随之结束, 可以用该文件号打开别的文件, 也可以用该文件号再次打开原文件。

③Close 语句用于以 Output 和 Append 模式打开文件时, 语句执行后将文件缓冲区的内容全部写入文件, 并释放缓冲区所占用的内存。

#### 7.4.2 顺序文件的读写

[例 7-10] 设计一个学籍管理程序, 把学生名册存入磁盘文件, 根据需要取出并显示数据文件中的学生名册。设学生名册包括编号、姓名、性别、年龄、籍贯、家庭地址和电话等数据项。

[分析]

若文件中只包含文本信息, 可以用顺序访问方式进行读写。顺序文件中的数据按字符串

存储，如一个四位数以顺序文件存取，需要 4 个字节的存储空间；作为一个整数来存储，只需要 2 个字节。因此，顺序访问方式不太适用于存储大量数据的文件。

用顺序访问方式打开一个文件时，可用以下格式的 Open 语句打开：

```
Open <文件名> For [Input|Output|Append] As <文件号> [Len=buffersize]
```

以 Input 模式打开顺序文件时，该文件必须是已存在的文件，否则会产生一个错误。但以 Output 或 Append 模式打开一个不存在的文件时，Open 语句可以先创建文件，然后再打开。

以 Input、Output 或 Append 模式打开一个顺序文件后，为了能在其它类型的操作中重新打开这类文件，必须先关闭该文件。

例如，以 Input 模式打开的文件，对文件进行修改操作后，若要将修改后的文件内容保存回文件，应先用 Close 语句关闭该文件，然后再以 Output 模式打开，最后再把文件内容写回到文件中。

[建立工程]

新建工程，按默认名称保存工程为“工程 1”，窗体名称保存为 Form1。

界面布局：在窗体上添加一个 Text1 和 label1 控件，并复制成 5 个，形成控件组，用户输入学生名册数据；添加一个文本框控件 Text2，设置多行属性 MultiLine 为 True，ScrollBars 属性为 3-Both，最后添加两个命令按钮 Command1 和 Command2，调整界面如图 7-13 所示。

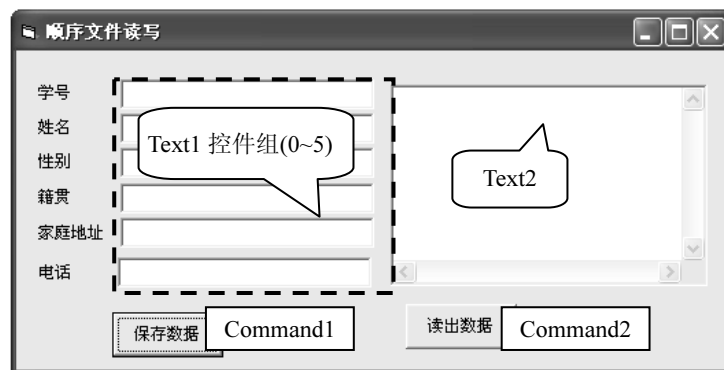


图 7-13 界面布局

[程序代码]

```
1 '保存数据到文件
2 Private Sub Command1_Click()
3     If Dir("d:\XjdaFile.txt") <> "" Then
4         Open "d:\XjdaFile.txt" For Append As #1
5     Else
6         Open "d:\XjdaFile.txt" For Output As #1
7     End If
8     Write #1, Text1(0), Text1(1), Text1(2), Text1(3), Text1(4), Text1(5)
9     Close #1
10 End Sub
11
12 '从文件中读出数据
13 Private Sub Command2_Click()
14     '判断文件是否存在
15     If Dir("d:\XjdaFile.txt") = "" Then Exit Sub
16
17     Dim dat(5) As String, tmpStr As String
18     Text2.Text = "": tmpStr = ""
19
20     Open "d:\XjdaFile.txt" For Input As #1
```

```

21
22 While Not EOF(1)
23     Input #1, dat(0), dat(1), dat(2), dat(3), dat(4), dat(5)
24
25     For i = 0 To 5
26         tmpStr = tmpStr & dat(i) & " "
27     Next
28
29     Text2.Text = Text2.Text & tmpStr & vbCrLf
30     tmpStr = ""
31 Wend
32 Close #1
33 End Sub
34
35 Private Sub Form_Load()
36     For i = 0 To 5
37         Text1(i).Text = ""
38     Next
39     Text2.Text = ""
40 End Sub

```

#### [代码说明]

行 3~9: 如果文件存在, 则以追加方式将数据保存在已有的数据后面; 如如果文件不存在, 则将建立新文件保存数据。Dir 系统函数可用于判断文件是否存在, 如果存在, 则返回文件名, 否则将返回空字符串。行 8 中依次将学生属性值分别保存为文件中的一个字段, 便于后面的读取数据操作, 也可以全部属性值连接成一个字段来保存, 不过在读取数据时, 只能作为一个字段来读取, 读写完毕必须关闭文件。

行 22~32: 依次读取所有的字段值, 并连接成字符串, 显示在 Text2 中。也可以按行来读取数据, 将行 23~30 的代码改写成如下代码:

```

While Not EOF(1)
    Line Input #1, tmpStr
    Text2.Text = Text2.Text & tmpStr & vbCrLf
Wend

```

这样代码更加简洁, 但读出的数据, 每一字段内容都将自动带双引号, 如果要单独取出每一个字段, 则需要再次处理。

[运行结果] 如图 7-14 所示。



图 7-14 运行结果

#### [总结与提高]

##### 1. 从顺序文件中读取数据

文件以 Input 模式打开后, 可以用 Input() 函数或 Line Input#、Input# 语句按顺序从文件中读取数据, 将文件内容读出来并复制到变量中。

例如：

```
Dim LinesFromFile, NextLine As String
Do Until EOF(FileNum)
    Line Input #FileNum, NextLine
    LinesFromFile = LinesFromFile + NextLine + Chr(13) + Chr(10)
Loop
```

程序中，用 **Line Input #**语句读取文件中的一行，也可以用 **Input #**语句读取文件中的一列数字或字符串表达式。

例如，要从一个邮件列表文件中读取一行，可用以下语句：

```
Input #FileNum, name, street, city, state, zip
```

以下简单介绍这几个读语句与函数的使用方法：

#### (1) Input 函数

函数格式：**Input(n, [#]<文件号>)**

功能：返回从打开的文件中读取的字符串，包括以 **Input** 或 **Binary** 方式打开的文件。

说明：

(1) **n** 是任意数值表达式，用于指定要读取字符的个数。<文件号>是任意打开的有效文件号。

(2) 与 **Input #**语句不同，**Input** 函数返回读出的所有字符，包括逗号、回车符、空白列、换行符、引号和前导空格等。

#### (2) Input # 语句

语句格式：**Input #<文件号>,<变量列表>**

功能：从已打开的顺序文件中读出数据并将数据复制给变量。

说明：

①参数<文件号>可以是任意打开的有效文件号。

②<变量列表>是用逗号分开的变量列表，从文件中读出的值将分配给这些变量；这些变量不能是一个数组或对象变量，但可以用变量描述数组元素或用户定义类型的元素。

③该语句只能读取以 **Input** 或 **Binary** 方式打开的文件。

④读出数据时，不必经过修改就可直接将标准的字符串或数值数据复制给变量。

⑤输入数据中的双引号 (“”) 将被忽略。

#### (3) Line Input 语句

语句格式：**Line Input #<文件号>,<变量名>**

功能：从已打开的顺序文件中读出一行并分配给字符串变量。

说明：

①<文件号>参数可以是任意有效的文件号。

②<变量名>可以是有效的变体类型或字符串类型的变量名。

③**Line Input #**语句一次只从文件中读出一行字符，直到遇到回车符(**Chr(13)**)或回车换行符(**Chr(13) + Chr(10)**)为止。尽管 **Line Input#**语句到达回车换行时会识别行尾，但是，把该行读入变量时并不包括回车换行符。如果要在保留回车换行符，必须在代码中给变量添加回车换行符。

## 2. 向顺序文件写入数据

要向顺序文件存入变量中的数据，应以 **Output** 或 **Append** 模式打开文件，然后用 **Print #** 或 **Write #**语句进行写入操作，如 **Print #FileNum, TheBox.Text**。

**Write #**语句可把数值或字符串表达式写入文件。写入时，自动用逗号分隔每个表达式，并且在字符串表达式端放置引号。

例如：

```
Dim AnyString As String, AnyNumber As Integer
```



```
AnyString = "AnyCharacters": AnyNumber = 23445
Write #FileNum AnyString, AnyNumber
```

以上程序段把两个表达式写入由 FileNum 指定的文件。第一个表达式包含一个字符串，第二个表达式包含数字 23445。因此，把字符（“AnyCharacters”,23445）写入到文件中（包括所有标点符号）。

以下简要介绍两个写入语句的使用。

#### (1) Print # 语句

语句格式：Print #<文件号>, [<输出项列表>]

功能：将格式化显示的数据写入顺序文件中。

说明：

①<文件号>参数可以是任意打开的有效文件号。

②<输出项列表>参数可以是表达式或要打印的表达式列表。该参数的设置格式如下：

[{Spc(n) | Tab[(n)]}] [<表达式>] [<字符插入点>]

格式中各项的含义见表 7-10。

表 7-10 <输出项列表>参数的设置项

设置	描述
Spc(n)	在输出数据中插入空白字符，n 是指要插入的空白字符个数
Tab(n)	将插入点定位在某一绝对列号上，n 是列号。无参数的 Tab 将插入点定位在下一个打印区的起始位置
表达式	要打印的数值表达式或字符串表达式
字符插入点	指定下一个字符的插入点。用分号将插入点定位在上一个显示字符之后，用 Tab(n)将插入点定位在某一绝对列号上，用无参数的 Tab 将插入点定位在下一个打印区的起始处。省略该项则在下一行打印下一个字符

③用 Line Input #或 Input 语句从文件中读出数据，用 Print #语句向文件中写入数据。若省略参数<输出项列表>，且<文件号>后只含有一个列表分隔符，则将一空白行输出到文件中。

④多个表达式之间可用一个空格或一个分号隔开，空格与分号等效。

⑤对于布尔类型的数据，写入的是 True 或 False。

⑥用操作系统能够将 Date 类型数据写入文件中。若未指定日期或时间部件，或这些部件的设置为零，则只将指定的部分写入文件中。

⑦若<输出项列表>的数据是 Empty，则不将任何数据写入文件；如果<输出项列表>的数据是 Null，则将 Null 写入文件。

⑧用 Print #语句写入的字符型数据不在字符串的两端放置引号。

#### 2. Write # 语句

语句格式：Write #<文件号>, [<输出项列表>]

功能：将数据写入顺序文件。

说明：

①<文件号>参数可以是任意有效的文件号。

②<输出项列表>参数是要写入文件的数值表达式或字符串表达式，一般用逗号将这些表达式分开。多个表达式之间也可以用空格、分号隔开。空格和分号是等效的。

③用 Input #语句从文件中读出数据，用 Write #语句向文件中写入的数据。如果省略<输出项列表>参数，并在<文件号>后加一个逗号，则将一个空白行输出到文件中。

④与 Print #语句不同，将数据写入文件时，Write #语句会在项目和标记字符串的引号之间插入逗号。没有必要在<输出项列表>中键入明确的分界符，在<输出项列表>中最后一个字

符写入文件后，自动插入一个新行的字符，即回车换行符(Chr(13)+Chr(10))。

### 7.4.3 随机文件的读写

#### 1. 随机文件的打开（建立）与读写操作

随机文件由一些相同的记录组成，每条记录包含一个或多个字段。只有一个字段的记录对应于任一标准类型，如整型或定长字符串型。具有多个字段的记录对应于用户定义类型。例如，以下定义的 **Worker** 类型可创建由三个字段组成的 19 个字节的记录。

```
Type Worker
  LastName As String * 10
  Title As String * 7
  Rank As String * 2
End Type
```

该类型的特点是每个字段都具有固定的长度，可以用随机文件来存取这些信息。随机文件由记录构成，每条记录都由定长的字段组成，因而每条记录的长度也相同。利用这一特点，可以方便地找到某条特定记录中某个字段信息。例如，以上 **Worker** 类型创建由三个字段组成的 19 个字节的记录，该随机文件中的记录和字段如图 7-15 所示。

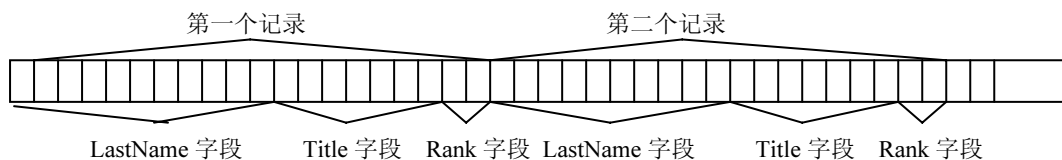


图 7-15 随机文件中的记录和字段

在应用程序打开一个文件进行随机访问前，应先声明处理该文件数据所需的变量，其中包括用户定义类型的变量，它对应于文件中的记录和其它标准类型变量，可用这些变量保存随机文件并处理相关的数据。此外，还应定义一个类型，对应该文件包含或将包含的记录。

例如，在一个职工记录文件中，可定义名为 **Person** 的用户定义数据类型，程序代码如下：

```
Type Person
  ID As Integer
  MonthlySalary As Currency
  LastReviewDate As Long
  FirstName As String * 15
  LastName As String * 15
  Title As String * 15
  ReviewComments As String * 150
End Type
```

随机文件中的所有记录都必须有相同的长度，若输入字符串比字段的定义长度少，系统在字符后面用空格（字符代码 32）填充；若输入字符串比字段的定义长度长，该字符串会被截断；若用长度可变的字符串，则任何用 **Put** 存储或用 **Get** 检索的记录总长度都不能超过 **Open** 语句的 **Len** 分句中指定的记录长度。

定义与典型记录对应的类型后，应声明程序变量，以用来处理打开的随机文件。

例如：

```
Public Employee As Person      '记录变量
Public Position As Long        '跟踪当前记录
Public LastRecord As Long      '文件最后一条记录的编号
```

作好准备工作后，可以用以下格式的 **Open** 语句打开随机文件：

**Open** <文件名> [**For Random**] **As** <文件名> **Len** = <记录长度>

默认关键字 **For Random** 表示以随机访问方式打开文件。表达式 **Len=<记录长度>**参数

指定每条记录的长度，是必选参数，即打开随机文件时必须指定记录的长度。若指定的<记录长度>比记录的实际长度短，则会产生错误，若比实际长度长，则记录可写入，但会浪费一些磁盘空间。部分数据类型的记录长度如表 7-11 所示。

表 7-11 部分数据类型的记录长度

数据类型	记录长度
Integer	2
Long Integer	4
Single	4
Double	8
Currency	8

不能直接确定字符串类型数据的长度，因此，Visual Basic 6.0 允许声明定长字符串，方法是在声明变量时直接在 String 关键词后加 “\* n”，其中，n 即所定义字符串的长度。

随机文件打开后，若要编辑随机文件，应先把记录从文件读到变量，然后改变变量的值，再把变量的值写回该文件。

随机文件的读写方法：

(1) 用 Get 语句把记录复制到变量。

语句格式：Get [#]<文件号>， [<记录号>]， <变量名>

例如，若要把一个记录从职工记录文件复制到 Employee 变量，可用以下语句：

Get FileNum, Position, Employee

其中，FileNum 包含 Open 语句（用于打开文件）的编号；Position 包含要复制的记录号；而 Employee 声明为用户定义类型 Person，用来接收记录的内容。

(2) 用 Put 语句把记录添加或替换到随机文件中。

语句格式：Put [#]<文件号>,[<记录号>],<变量名>

若要替换记录，必须使用 Put 语句指定想要替换的记录位置。例如：

Put #FileNum, Position, Employee

该语句用 Employee 变量中的数据替换由 Position 指定编号的记录。

## 2. 在随机文件中添加记录

要向随机文件添加新记录，应使用 Put 语句，并把 Position 变量的值设置为比文件中的记录数多 1。例如，要在一个包含 5 条记录的文件中添加一条记录，则应设 Position 为 6。

以下语句把一个记录添加到文件的末尾：

LastRecord = LastRecord + 1

Put #FileNum, LastRecord, Employee

### 7.4.4 二进制文件的读写

[例 7-11] 编写一个程序，将例 7-10 所建立文件的前 128 个字节以二进制方式读取并显示出来，运行结果如图 7-16 所示。

[程序代码]

```
Dim tmpStr As String, aa As Integer, A As String
```

```
Open "C:\dat1.dat" For Binary As #1
```

```
Seek #1, 1
```

```
For i = 1 To 208
```

```
Get #1, i, aa
```

```

A = Right(Hex(aa), 2)
If i Mod 16 <> 0 Then
    If Len(A) = 1 Then A = "0" + A
    tmpStr = tmpStr & A & " "
Else
    If Len(A) = 1 Then A = "0" + A
    tmpStr = tmpStr & A & " " & vbCrLf
End If
Next i
Close #1
MsgBox tmpStr

```

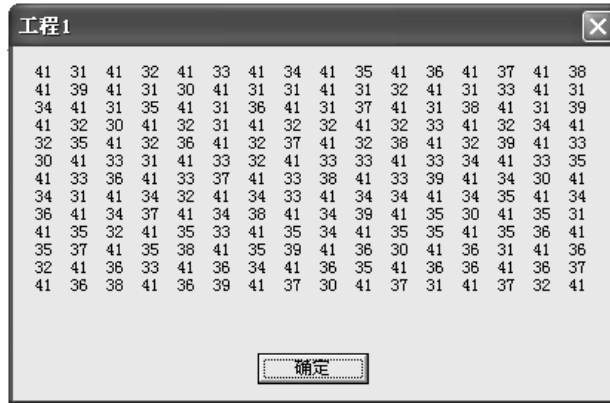


图 7-16 用二进制文件显示文件内容

#### [代码说明]

以上程序先把文件作为二进制文件打开，然后从中取出一个两个字节的整型数。由于每个整型数的排列都是高位字节在前，低位字节在后，因此，文件的第一个字节实际上是整型数的后一个字节。把整型数转换为十六进制后，取每一个数的后一个字节，即十六进制数的后两位数，就可以得到二进制文件中的各个字节。另外，该程序用 16 取模实现每行打开 16 个字节。

[运行结果] 如图 7-16 所示。

#### [总结与提高]

二进制文件是字节的集合，所存储的数据不需要按某种方式进行组织，允许程序按所需的任何方式组织和访问，这种存取方式最为灵活。当要保存的文件很小时，应该用二进制文件。

实际上，二进制文件并不是一种文件类型，它只是对文件的一种处理方式。可以将二进制文件看作记录长度为 1 的随机文件，读写随机文件的 Put、Get 语句也可以用于读写二进制文件。一个文件是随机文件还是二进制文件由打开文件的模式所决定。

#### 1. 打开二进制文件

打开二进制文件的 Open 语句格式如下：

**Open <文件名> For Binary As <文件号>**

其中，参数<文件名>是任意存在的文件名，<文件号>指定所打开的文件号，Binary 是打开二进制文件模式的关键词。

打开二进制文件的 Open 语句与打开随机文件时不同，它没有 Len=<记录长度>短语，实际上，二进制文件与随机文件可以用相同的方式进行处理，在打开二进制文件的 Open 语句中即使包括记录长度，也会被自动忽略。

#### 2. 在长度可变的字段中保存信息

为了分析二进制文件与随机文件之间的区别，设有一个职工记录文件，使用随机文件时，

文件中用长度固定的记录和字段来存储职工信息。定义 **Person** 类型如下：

```
Type Person
    ID As Integer
    MonthlySalary As Currency
    LastReviewDate As Long
    FirstName As String * 15
    LastName As String * 15
    Title As String * 15
    ReviewComments As String * 150
End Type
```

以随机方式存取时，无论字段的实际内容如何，每条记录都占用 209 个字节。而在用二进制文件时由于不需要固定长度的字段，类型声明语句中可以省略字符串长度参数，这样就可以根据实际内容使文件占用的磁盘空间降到最小。

由于各字段长度是可变的，在职工记录文件中，每个职工记录只存储实际需要的字节长度。用长度可变字段进行二进制输入/输出的缺点是不能随机地访问记录，而必须用顺序方式访问记录以了解每条记录的长度。虽然可以直接查看文件中指定字节的位置，但是，如果记录的长度可变，则无法直接知道某条记录在文件中具体的字节位置。

### 3. 文件指针

在二进制文件中，可以把文件指针移到文件的任意位置，因此，文件指针对二进制文件存取很重要。文件指针的定位通过 **Seek** 语句来实现。

语句格式：**Seek** [#]<文件号>,<位置>

功能：设定二进制文件中下一个访问操作发生的位置。

说明：<文件号>参数是任意有效的文件号，<位置>参数是一个数值表达式，来指定下一个要读写操作的发生的位置，取值范围是介于  $1 \sim (2^{31}-1)$  之间的数字。

对于以 **Binary**、**Input**、**Output** 或 **Append** 方式打开的文件，指示位置的数值表达式从文件开头到所指字符位置为止的字节数，即执行下一个操作的地址，文件的第一个字节的位置是 1。对于随机方式打开的文件，位置是一个记录号。

在 **Get** 及 **Put** 语句中，记录号优先于由 **Seek** 语句确定的位置。若在文件结束后进行 **Seek** 操作，则文件的写入操作会把文件扩大。若试图对一个位置为负数或零的文件进行 **Seek** 操作，将会导致一个“错误记录号”的错误发生。

**Seek** 函数用于返回文件当前的位置，函数的格式为：**Seek**(<文件号>)。

若<文件号>所指的文件以随机方式打开，**Seek** 语句把文件指针移到指定的字节位置，**Seek** 函数则返回下一个要读取或写入的记录号；若文件以 **Binary**、**Input**、**Output** 或 **Append** 模式打开，**Seek** 语句把文件指针移到指定的字节位置，**Seek** 函数返回发生下一个操作的字节位置。

[例 7-12] 编写一个建立二进制文件的程序。

[程序代码]

```
Dim tmpStr As String
Dim FileName As String

FileName = "C:\dat1.dat"
Open FileName For Binary As #1
For i = 1 To 100
    tmpStr = "A" & i
    Put #1, , tmpStr
Next
Close #1
```

[例 7-14] 用随机文件创建一个简单的个人通信录管理程序，要求具有添加、删除及浏览的功能。

[建立工程]

1. 创建标准 EXE 类型的工程，构造用户界面，如图 7-18 所示。

(1) 在窗体上放置的四个文本框，分别用于录入和显示姓名、电话、呼机号码及住址；三个命令按钮用来实现添加、删除和退出功能；一个列表框 List1 用于浏览通信录内容。

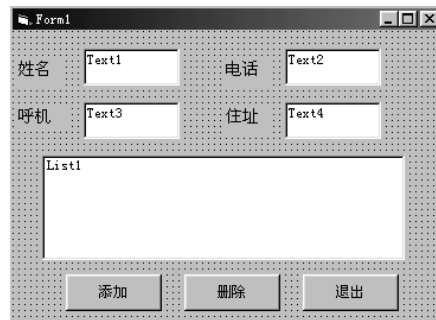


图 7-18 个人通信录用户界面

(2) 对窗体和各个控件进行属性设置，见表 8-5。

表 8-5 个人通信录属性设置

对象	属性名称	属性值	备注
Form1	Name	Form1	
	Caption	个人通信录管理程序	
TxtName	Text		初始化文本框内容为空
TxtTel	Text		初始化文本框内容为空
TxtBP	Text		初始化文本框内容为空
TxtAddr	Text		初始化文本框内容为空
CmdAdd	Caption	添加	
CmdDel	Caption	删除	
CmdExit	Caption	退出	
List1	MultiSelect	0	不允许多重选择

[程序代码]

假设数据存放在“C:\通信录.dat”文件中。

(1) 在工程中添加一个模块，用以定义通信录的记录结构。在该模块的通用声明段中编写以下代码，创建一个用户自定义类型 People，包含 name、tel、bp 和 address 四个成员，分别表示姓名、电话、呼机和住址。

```
Type People
    Name As String * 9
    Tel As Strine * 15
    Bp As String * 15
    Address AS String * 40
End Type
```

(2) 在窗体模块的通用声明段定义记录变量 rec:

```
Dim rec As People
```

(3) 启动程序后，在窗体列表框中自动读取并显示通信录文件“C:\通信录.dat”的内容，

程序代码如下:

```
Private Sub Form_Activate( )
Dim fileno , I As Integer
Dim info As String
List1.Clear                \清除列表框
TxtName.Text = ""         \清空各文本框
TxtTel.Text = ""
TxtBP.Text = ""
TxtAddr.Text = ""
Fileno = freefile          \取下一个可用文件号
Open "c:\通信录.dat" For Random As #fileno Len = Len(rec)
If LOF(fileno) = 0 Then    \文件为空则提示用户并返回
    MsgBox"当前通信录中无记录! ",, "提示信息"
    Close #fileno
    Exit Sub
Else
For I=1 to LOF(fileno)/Len(rec)
Get #fileno, i, rec
If I=1 Then
    TxtName.Text=rec.name
    TxtTel.Text=rec.tel
    TxtBP.Text=rec.bp
    TxtAddr.Text=rec.address
End If
Info = Trim(rec.name)+Space(5)+(rec.tel)+Space(5)
Info = Info + Trim(rec.bp)+Space(5)+Trim(rec.address)
List1.AddItem info
End If
Close #fileno
End Sub
```

(4) 单击“添加”按钮, 将各文本框中编辑好的内容以一条新记录追加到通信录文件中, 同时在列表框中选中该记录, 程序代码如下:

```
Private Sub CmdAdd_Click()
Dim fileno, I As Integer
If TxtName.Text="" And TxtTel.Text="" And TxtBP.Text="" And TxtAddr.Text=""
Then
MsgBox"请输入个人信息!",, "提示信息"
Exit Sub
End If
Fileno = FreeFile
Rec.name = TxtName.Text
Rec.tel = TxtTel.Text
Rec.bp = TxtBP.Text
Rec.address = TxtAddr.Text
Open "c:\通信录.dat" For Random As #fileno Len=len(rec)
Put #fileno, LOF(fileno)/Len(rec)+1, rec
Close #fileno
List1.AddItem TxtName.Text+Space(5)+TxtTel.Text+Space(5)+
TxtBP.Text+Space(5)+TxtAddr.Text
List1.Selected(List1.ListCount-1) = True
TxtName.SetFocus
End Sub
```

(5) 在列表框中选中一条记录后, 单击“删除”按钮将从通信录文件中删除该记录。

清除随机文件中删除记录的方法: 先创建一个新文件, 然后把有用的所有记录从原文件复制到新文件, 再关闭原文件并用 Kill 语句删除, 最后用 Name 语句把新文件以原文件的名字重新命名。

程序代码如下：

```
Private Sub CmdDel_Click()
For I = 0 to List1.ListCount-1
    If List1.Selected(i) = True Then    '确定要删除的记录号
        Delno = I + 1
        Exit For
    End If
Next i
Open "c:\通信录.daf" For Random As #1 Len = Len(rec)
Open "c:\通信录.tmp" For Random As #2 Len = Len(rec)
For I=1 To LOF(1)/Len(rec)
If i<>delno Then
    Get #1,I,rec
    Put #2,,rec
EndIf
Next I
    Close #2
    Close #1
    Kill "c:\通信录.daf"
    Name "c:\通信录.tmp" As "c:\通信录.dat"
    List1.RemoveItem delno - 1
End Sub
```

(6) 单击“退出”按钮，确认后结束程序运行，程序代码如下：

```
Private Sub CmdExit_Click()
    Dim Var As Integer
    Var=MsgBox("确实要退出个人通信录管理程序吗?", 256+32+1, "提示信息")
    If Var=1 Then End
End Sub
```

(7) 用户在列表框中任意选定一条记录时，同时将其内容显示在各文本框中，程序代码如下：

```
Private Sub List1_Chck()
    Dim fileno, I As Integer
    Dim info As String
    Fileno = freefile
Open "c:\通信录.daf" For Random As #fileno Len=Len(rec)
If LOF(fileno) = 0 Then    '若文件为空，则各文本框亦为空
TxtName.Text = ""
TxtTel.Text = ""
TxtBP.Text = ""
TxtAddr.Text = ""
Close #1 fileno
End If
Get #fileno, List1.ListIndex + 1,rec
TxtName.Text = rec.name
TxtTel.Text = rec.tel
TxtBP.Text = rec.bp
TxtAddr.Text = rec.address
    Close #fileno
EndSub
```

[运行结果] 程序运行结果如图 8-19 所示。



Form1

姓名: 王杰      电话: 010-64872345

呼机: 95951-12345      住址: 青年路56号

许静	0351-4133333	301
王杰	010-64872345	9595

添加      删除      退出

图 8-19 个人通信录程序运行界面