

计算机语言函数应用丛书

Delphi

Delphi

计算机语言函数应用

• 袁晓君 刘宏立 袁晓文 编著



科学出版社

计算机语言函数应用丛书

Delphi

Delphi

计算机语言函数应用

• 袁晓君 刘宏立 袁晓文 编著

科学出版社

2000

内 容 简 介

本书是一本 Delphi 程序库的技术参考手册。书中详细介绍了运行时间库 RTL 包含的常量、变量、函数和过程,讲述了它们的定义和使用方法,给出了 Delphi 的 RTL 全面的参考资料;详细讲解了 VCL 的层次结构、继承关系、VCL 中包含的组件,以及 VCL 组件库的属性、事件、方法的定义和使用方法,剖析了 VCL 的层次结构和继承关系;还给出了相当数量的使用实例。本书是 Delphi 程序设计人员必备的参考书。

图书在版编目(CIP)数据

Delphi 计算机语言函数应用/袁晓君等编.-北京:科学出版社,2000.1

ISBN 7-03-007858-6

I. 计… II. 袁… III. Delphi 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(1999)第 39297 号

IS11/07

科学出版社 出版

北京东黄城根北街 16 号
邮政编码:100717

北京双青印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

2000 年 1 月第 一 版	开本: 787×1092 1/16
2000 年 1 月第一次印刷	印张: 29
印数: 1—5 100	字数: 889 000

定价: 40.00 元

(如有印装质量问题, 我社负责调换(环伟))

前 言

自 1995 年 3 月 Delphi 1.0 推出以来, 其友好的集成开发界面、可视化的双向开发模式、良好的数据库应用支持以及程序开发和程序运行的高效率, 得到了广大 Windows 程序开发人员的好评, 尤其是系统提供的功能强大的 VCL 组件库, 大大提高了应用程序的开发效率, 深受 Windows 程序设计人员的喜爱。

在近三年的时间里, Delphi 的应用在国内得到了极大的普及, Delphi 系统本身也由 1.0 版本升级到 4.0 版本, 从 16 位平台升级到 32 位平台, 国内许多程序员正在使用 Delphi 这一优秀的前端开发工具开发大型、关键性的应用系统。

本书是一本详细介绍 Delphi 程序库的书籍, 包含 Delphi 程序库的运行时间库 (Run-Time Library, RTL) 和可视组件库 (Visual Component Library, VCL) 两大部分的内容, 是 Delphi 程序设计人员不可缺少的参考资料。

本书的编写是基于 Delphi 4.0 Client/Server 版本完成的。

Delphi 4.0 较之以前的版本包含下列新的特征和改进。

1. 新的语言扩展

Delphi 4.0 将 Object Pascal 编程语言扩展到包括动态数组、方法过载和缺省参数等内容。

2. 工程项目管理器

新的工程项目管理器可以让你将许多工程项目组合在一起, 以便使这些工程项目集中为一个工程项目组实现运行。这一管理器允许编程者将相互依赖的工程组织到一起并加以运算, 例如将分散的级合并为多级应用或动态链接库, 该管理器负责实现集合运行。

3. 代码资源管理器 (包含类完成、模块导航及代码浏览)

代码资源管理器通过自动完成类生成的许多步骤而使创建类变得更加简单。该自动过程可在各节的分界处定义方法的原型, 在执行段类完成时产生基本框架代码, 迅速在单个文件中或在各段之间漫游。

4. 可停放工具窗口

IDE (Integrated Development Environment) 应用可停放于代码编辑器的工具窗口使自身更加结构化。可以随意地将工具窗口拖曳到编程者满意的地方。代码资源管理器和工程项目管理器也可以成为工具窗口停靠的地点。

5. 改良的调试工具

集中调试器具有许多新特点, 包括远程调试、多进程调试、CPU 监视、检查器、断点改进、特定调试器子菜单和可停靠窗口。

6. MTS 支持

应用于 MTS 的支持集中为多级数据库支持。此外, 一个新的 Wizard 使建立 MTS 服务对象变得更容易。

7.ActiveX 的改进

Delphi 4.0 提供了扩展的 ActiveX 支持。

8.VCL 的改进

改善了的 Delphi 对象层包括一个新增的支持 NT 服务性应用软件的内容。另外，一个新增设的列表内容可以令程序员集中管理菜单和工具按钮命令。改善了的 VCL 空间支持拖拉操作，以便提供针对整个窗口规划的格外控制或更多的服务。

9. 面向 2000 年维护的 RTL 支持

函数 StrToDate 和 StrToDateTime 应用全局变量 TwoDigitYearCenturyWindow 来控制对 2000 年日期转换时的诠释。

10.CORBA 支持

Delphi 客户/服务器及企业版包括面向 CORBA 客户和服务器的应用软件的支持。Wizards 使建立 CORBA 服务器变得更容易，动态调用界面 (DII) 可以让程序员以客户身份向存在的 CORBA 服务器传递信息。CORBA 的能力也被建入多层数据库支持。程序员甚至可以建立一个同时操纵 COM 客户与 CORBA 客户的服务器。

11. 改进的客户数据集

客户数据集目前可支持更加广泛的滤波器带宽，并允许对象相关场类型。数据集的改进可以使同层文件数据库建立数据集更加容易。

12. 多级支持

Delphi 4.0 为程序员提供了更多的多级应用控制，包括刷新支持和更多的对于包含在数据包中及如何升级等应用的控制。可以令程序员更加轻松地调用服务器界面。作为一个新类，TDataSetProvider 可以让程序员利用并补充数据集。原版/明细数据支持有效使用嵌套表格。客户数据集的改进使向应用软件服务器传送参数或者在数据包中存储客户信息变得简单易行。

13. 数据库的改进

Delphi 4.0 数据访问组件被改进，目的是让程序员在建立自己的表格和数据模式时，用以调整数据模式。数据访问组件和 Borland 数据库引擎 (BDE) 方面的改变允许程序员以新的数据服务器类型访问数据。可视化访问程序生成器由一种全新的智能访问程序生成器—SQL 生成器所代替。

Delphi 是一个性能卓越，功能强大的应用系统前端开发工具，其程序库包含了基本程序开发，数据库应用程序开发，DDE，OLE，多媒体应用程序开发等多方面内容。本书就其中运行时间库 (RTL) 和可视组件库 (VCL) 两大部分进行阐述，着重介绍了 RTL 的函数、过程以及 VCL 的组件，力求详尽。如遇谬误，敬请读者指正。

作者

1999. 11

目 录

前言

1 Delphi 程序库	(1)
1.1 Delphi 程序库构成	(1)
1.1.1 Delphi 运行时间库 RTL	(2)
1.1.2 Delphi 可视组件库 VCL	(2)
1.2 RTL 函数、过程常量和变量	(3)
1.2.1 RTL 常量和变量	(3)
1.2.2 RTL 函数	(4)
1.3 VCL 简介	(12)
2 RTL 详述	(30)
3 VCL 组件	(192)

1 Delphi 程序库

Delphi 是 Borland 公司出品的一个面向对象的可视化程序开发环境，系统提供的编程语言 Object Pascal 既有结构化程序语言的各种功能和特征，又具备了面向对象的程序语言（OOPL）的继承、封装、多态以及软件的重用等特征。Object Pascal 是一个高级的、需编译的、强键入式的语言，它支持结构化和面向对象实际，它支持 Delphi 组件结构及 RAD 环境。它的优点在于代码的易读性、编译的快捷性以及模块程序应用多单元的文件。

大部分程序员在 Delphi 集成开发环境（IDE）中编写并编译源代码。Delphi 可处理许多有关设置项目及源文件的详细内容，Delphi 还对程序组件加以约束，严格地说，这些程序组件并非 Object Pascal 语言规范的一个部分。例如，Delphi 执行某一特定文件——如果不在 IDE 中编写源程序并且在命令行上编译该程序，则可以不必理会程序命名的协议。

Delphi Object Pascal 不但提供了丰富的运行时间库 RTL，还提供了大量的可重用的软件模块（component，组件），本章将对 Object Pascal 中的运行时间库 RTL 和可视组件库 VCL 进行介绍。

本章介绍的主要内容包括：

- (1) Delphi 程序库组成
- (2) 构成 Delphi 程序库的程序单元及各单元中定义的内容
- (3) 运行时间库 RTL 内容列表
- (4) 可视组件库 VCL 分类列表
- (5) 可视组件库 VCL 层次结构详解

Delphi 对象由许多方法组成，并存在于许多控件、属性和事件中。属性描述了包含在对象中的数据资料。方法是对象可以执行的活动。事件则是对象与之相互作用的环境。所有的对象都是由其祖先对象 Tobject 继承而来的。

组件是可视对象，在设计时可供熟练操作。所有的组件都是由 Tcomponent 继承而来。

全局例程包括来自运行时间库和可视组件库的过程与函数。这些例程并非属于某一类，但可以直接被称为类方法或称作来源于类方法。

类型包括用于对象方法、属性、事件以及用于全局例程的返还类型和参数类型。许多时候类型作为方法、属性、时间及全局例程的入口。

变量存在于运行时间库和可视组件库中，其中一些变量可作为对象的实例。

1.1 Delphi 程序库构成

Delphi 程序库包含两部分的内容，即运行时间库 RTL 和可视组件库 VCL，Delphi

程序人员使用程序库的这两个部分构造应用程序。

1.1.1 Delphi 运行时间库 RTL

运行时间库 RTL (Run-Time Library) 是所有 Object Pascal 程序都可以调用的函数和过程的集合, 是 Delphi Object Pascal 程序库的基本组成部分。

运行时间库 RTL 由一系列常用的基础函数、过程、常量、变量定义组成, 运行时间库同时还包含处理程序运行异常 (exception) 的各种异常类。Object Pascal 运行时间库 RTL 给应用程序与可视组件库 VCL 提供了最基本的支持。

在 Object Pascal 中, 运行时间库中的函数、过程、常量和变量的定义都位于 System 单元中, 运行时间库中的各种异常类定义位于 SysUtil 单元中。

1.1.2 Delphi 可视组件库 VCL

Delphi 可视组件库 VCL (Visual Component Library) 是 Delphi Object Pascal 提供的面向对象的可重用软件模块, 是一个功能强大的程序对象库。在 VCL 中, 包含基本的程序对象, 其中大多数对象是可视化的组件, 使用这些对象和组件, 可以极大地提高编写 Windows 程序的效率。Delphi 本身, 包括集成开发环境 IDE 和常用工具, 都是使用 VCL 编写实现的。

Delphi Object Pascal 提供的可视组件库 VCL 是一个完全面向对象的程序库, 即 VCL 库是一个对象库, 类似于 Borland C++ 中的 OWL 或 Microsoft Visual C++ 的 MFC。

VCL 中的大部分对象称为组件 (component), 从总体上划分, VCL 对象可分为基础对象和组件两大部分; 在 VCL 的组件中, 许多组件在程序运行时是可见的, 这些组件也称为控件或控制 (control)。

大多数控件在运行时可以接收程序的焦点 (focus), 即可以接收 Windows 消息, 这类控件称为窗口控件 (windowed control, 基于窗口的控件), 与窗口控件对应, 那些不能接收焦点的控件称为非窗口控件 (non-windowed control), 这类控件不能接收 Windows 消息, 也不能包含其它的控件。图 1-1 简单描述了 Delphi VCL 库的组成和划分。

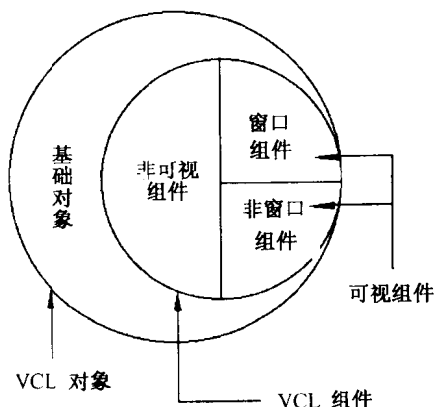


图 1-1 Delphi VCL 库组成

Delphi 程序库是 Delphi 应用程序开发的基础，下面，我们给出了一个简单的示意图描述应用程序与 RTL，VCL 以及 Object Pascal 语言的关系，如图 1-2 所示。

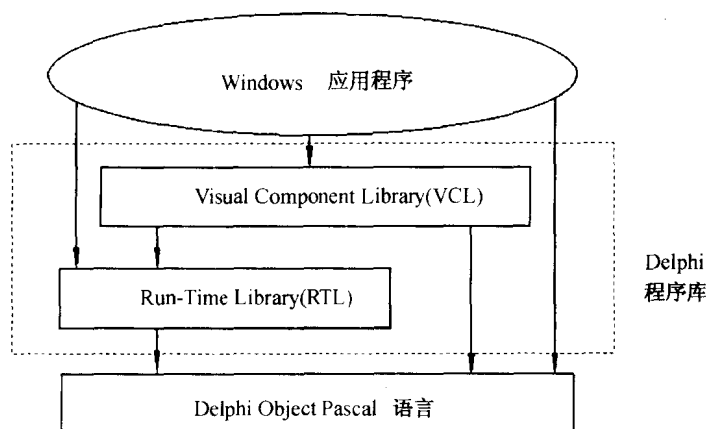


图 1-2 应用程序与 Delphi 程序库的关系

前面我们简单介绍了 Delphi Object Pascal 程序库的构成，在后面的章节中，我们将详细介绍 Delphi Object Pascal 程序库的各项内容。

1.2 RTL 函数、过程常量和变量

Delphi 运行时间库 RTL 是 Delphi 程序库的基础组成部分。RTL 包含函数、过程、常量等都是在 System 程序单元中定义的，System 单元自动被所有其它程序单元引用，所以，在任何 Object Pascal 程序单元中，我们不必在程序的 USES 语句中引用该单元，就可以自由地调用 RTL 库中的函数和过程，使用 RTL 库中的变量和常量定义。

在下一章中，我们将详细介绍各常量、变量、函数和过程的具体含义和用法，本节中，我们只给出 RTL 的内容列表。

1.2.1 RTL 常量和变量

Delphi 运行时间库 RTL 中定义的常量和变量如表 1-1 和表 1-2 所示。

表 1-1 Delphi 运行时间库中的类型常量定义

常量名称	类 型	初 值	说 明
CmdLine	Pchar	nil	命令行信息指针。
CmdShow	Integer	0	CreateWindow 函数的 CmdShow 参数值。
ErrorAddr	Pointer	nil	运行时间错误例程地址。
ErrorProc	Pointer	nil	错误处理过程。
ExceptDebugPtr	Pointer	nil	TuboDebugger 异常挂接指针。
ExceptionClass	Tclass	nil	异常处理句柄列表。
ExceptList	Word	0	未捕获异常的异常处理句柄。
ExceptProc	Pointer	nil	程序退出码。

续表

常量名称	类 型	初 值	说 明
ExitCode	Integer	0	退出处理例程。
ExitProc	Pointer	nil	文件打开模式常量。
FileMode	Byte	2	堆空间分配标志 (gmem_ Moveable)。
AllocFlags	Word	2	堆块尺寸。
HeapBlock	Word	8192	堆空间一致性检查挂接地址。
HeapCheck	Pointer	Nil	堆错误函数。
HeapError	Word	Nil	堆块空间最小尺寸值。
HeapLimit	Word	1024	堆段列表。
HeapList	Word	0	本程序当前执行副本 (instance) 标识。
Hinstance	Word	0	程序前一执行副本标识。
HprevInst	Word	0	I/O 结果状态。
InOutRes	Integer	0	程序段前缀 (PSP)。
PrefixSeg	Word	0	异常出发列表。
Raiselist	Word	0	随机数发生器初值 (种子)。
RandSeed	LongInt	0	地址选择器 (Selector) 增量。
SelectoInc	Word	0	处理器测试结果值。
Test8086	Byte	0	数据类型代码。
VtInteger	0		数据类型代码。
VtBoolean	1		数据类型代码。
VtChar	2		数据类型代码。
VtExtended	3		数据类型代码。
VtString	4		数据类型代码。
VtPointer	5		数据类型代码。
VtPChar	6		数据类型代码。
VtObject	7		数据类型代码。
VtClass	8		数据类型代码。

表 1-2 Delphi 运行时间库中的变量

变量名称	类 型	说 明
Input	Text Input	标准输入文件。
Output	Text Output	标准输入文件。

1.2.2 RTL 函数

在本节中, 我们分类给出 RTL 库的所有函数和过程。RTL 中的函数和过程分为 18 个类别, 分别列于表 1-3 到表 1-20 中, 如下所示。

1. 数学类例程 (Arithmetic routines)

RTL 提供的数学类例程如表 1-3 所示。

表 1-3 数学类例程

名 称	功能说明
Abs	取绝对值。
ArcTan	反正切三角函数。
Cos	余弦三角函数。
Exp	指数函数。
Frac	返回参数的小数部分。
Int	返回参数的整数部分。
Ln	计算自然对数。
Pi	返回圆周率 (3.1415926535897932385)。
Sin	正弦三角函数。
Sqr	求平方函数。
Sqrt	求平方根。

2. 控制台例程

RTL 提供的控制台例程如表 1-4 所示, 这些例程可用来编写界面, 类似于 DOS 程序的 Windows 应用。

表 1-4 控制台例程

名 称	功能说明
ClrEol	清除光标所在行光标后面的所有字符。
ClrScr	清除屏幕内容。
CursorTo	在虚拟屏幕中移动光标。
DoneWinCrt	关闭一个 CRT 窗口。
GotoXY	再虚拟屏幕中移动光标位置。
InitWinCrt	建立一个 CRT 窗口。
KeyPressed	测试是否有键盘按下。
ReadBuf	在 CRT 窗口中输入一行。
ReadKey	从键盘中读取字符。
ScrollTo	滚动 CRT 窗口。
TrackCursor	滚动窗口, 使得光标可见。
WhereX	返回光标的 X 坐标。
WhereY	返回光标的 Y 坐标。
WriteBuf	将一批字符写入 CRT 窗口。
WriteChar	向 CRT 窗口写入单个字符。

3. 日期/时间例程 (Date/time routines)

RTL 提供的日期/时间例程如表 1-5 所示。

表 1-5 日期/时间例程

名 称	功能说明
Date	返回当前时间。
DateTimeToStr	将日期时间值转换成字符串。
DateTimeToString	使用给定的格式化串将日期时间值转换成字符串。
DateToStr	使用 ShortDateFormat 全局变量将日期时间值转换成字符串。
DayOfWeek	DayOfWeek 返回 1—7 之间的数值表示给定日期为星期几。
DecodeDate	过程将 Date 参数分解成年、月、日值。
DecodeTime	过程将 Date 参数分解成时、分、秒值。
EncodeDate	编码一个日期值。
EncodeTime	编码一个时间值。
FormatDateTime	用给定的格式串格式化日期时间值。
Now	Now 返回当前日期和时间。
StrToDate	将字符串转换成日期格式。
StrToDateTime	将字符串转换成日期时间格式。
StrToTime	将字符串转换成时间格式。
Time	返回当前时间。
TimeToStr	将时间转换成字符串。

4. 动态内存申请例程 (Dynamic allocation routines)

RTL 提供的动态内存申请例程如表 1-6 所示。

表 1-6 动态内存申请例程

名 称	功能说明
Dispose	释放分配给一个动态变量的内存块。
FreeMem	释放给定尺寸的动态变量。
GetMem	申请内存空间。
New	申请动态变量。
MaxAvail	返回堆中最大可用的连续空间。
MemAvail	返回堆中的自由空间数量。

5. 文件管理例程 (File management routines)

RTL 提供的文件管理例程如表 1-7 所示。

表 1-7 文件管理例程

名 称	功能说明
ChangeFileExt	修改文件扩展名称。
ChDir	改变当前目录。
DateTimeToFileDate	将 Delphi 日期格式转换为 DOS 日期格式。

续表

名 称	功能说明
DeleteFile	删除文件。
DiskFree	返回磁盘剩余空间。
DiskSize	返回磁盘空间。
ExpandFileName	返回完整的路径字符串。
ExtractFileExt	返回文件扩展名称。
ExtractFileName	返回文件名称。
ExtractFilePath	返回文件的路径字符串。
FileAge	返回文件日期和时间。
FileCreate	建立文件。
FileClose	关闭文件。
FileDateToDateTime	将 DOS 格式的日期转换成 Delphi 日期。
FileExists	测试文件是否存在。
FileGetAttr	返回文件属性。
FileGetDate	返回文件的 DOS 格式的日期和时间。
FileRead	读文件。
FileSearch	在目录中查找文件。
FileSeek	文件定位（移动文件读写指针）。
FileSetAttr	设置文件属性。
FileSetDate	设置文件的日期和时间。
FileOpen	打开文件。
FileWrite	文件写入。
FindClose	关闭 FindFirst/FindNext 调用。
FindFirst	查找文件。
FindNext	查找下一个匹配的文件。
RenameFile	文件改名。

6. 浮点转换例程 (Floating point conversion routines)

RTL 提供的浮点转换例程如表 1-8 所示。

表 1-8 浮点转换例程

名 称	功能说明
FloatToDecimal	将浮点数转换成十进制表示格式。
FloatToStr	将浮点数转换成字符串格式。
FloatToStrF	将浮点数以特定格式、精度转换成字符串格式。
FloatToText	将浮点数转换成空结束字符串。
FloatToTextFmt	将浮点数转换成空结束字符串。
FormatFloat	用给定的格式化字符串格式话浮点值。
StrToFloat	将字符串转换成浮点值。
TextToFloat	从一个空结束字符串返回一个浮点值。

7. 流程控制例程 (Flow control routines)

RTL 提供的流程控制例程如表 1-9 所示。

表 1-9 流程控制例程

名 称	功能说明
Abort	终止一个进程，不报告错误。
Break	终止 for、while、repeat 语句的执行。
Continue	跳过后续语句，执行新一轮循环（在 for、while、repeat 语句中）。
Exit	立即退出本过程。
Halt	非正常结束程序执行。
RunError	结束程序执行，并激发一个运行时间错误。

8. 系统 I/O 例程 (I/O routines)

RTL 提供的系统 I/O 例程如表 1-10 所示。

表 1-10 系统 I/O 例程

名 称	功能说明
Append	以添加的方式打开现有文件。
AssignFile	将一个外部文件名与文件变量关联。
CloseFile	关闭一个文件（文件变量不再与磁盘文件相关联）。
Eof	测试文件指针是否指向文件结束位置。
Erase	删除与文件变量关联的磁盘文件。
FilePos	返回文件指针的当前位置。
FileSize	返回以字节为单位的文件长度或以记录文档为单位的记录数。
GetDir	返回磁盘的当前目录。
IOResult	返回上一次 I/O 操作后的状态值。
MkDir	建立一个子目录。
Rename	修改文件名称。
Reset	打开一个存在的文件。
Rewrite	创建一个文件并打开文件。
RmDir	删除一个空的子目录。
Seek	移动文件指针。
Truncate	删除当前文件指针位置以后的所有记录。

9. 内存管理例程 (Memory management routines)

RTL 提供的内存管理例程如表 1-11 所示。

表 1-11 内存管理例程

名 称	功能说明
AllocMem	分配一个内存块，并将块内的所有字节清零。
ReAllocMem	重新分配动态变量（扩充或压缩内存块）。

10. 杂用例程 (Miscellaneous routines)

RTL 提供的杂用例程如表 1-12 所示。

表 1-12 杂用例程

名 称	功能说明
Assigned	测试函数或过程变量是否为空。
Beep	生成一个用于计算机扬声器的标准嘟音。
CompareMem	对两个内存映象进行二进制比较。
Exclude	从集合中删除一个集合元素。
FillChar	在指定的地址处填充给定长度的字节（不做范围检查）。
Hi	返回无符号变量的高字节值（将整数视为 16 位）。
Include	为集合添加元素。
Lo	返回无符号变量的低字节值。
Move	在源和目的文件之间进行字节复制。
ParamCount	返回传递给程序的命令行中的参数个数。
ParamStr	返回特定的命令行参数。
Random	返回一个随机数。
Randomize	初始化系统内建立的随机数发生器。
SizeOf	返回参数的尺寸。
Swap	交换高低字节（用于 Integer 或 WORD 类型）。
UpCase	将字符转换成大写。

11. 序数类例程 (Ordinal routines)

RTL 提供的序数类例程如表 1-13 所示。

表 1-13 序数类例程

名 称	功能说明
Dec	将参数 X 减去 1 或 N。
Inc	将参数 X 增加 1 或 N。
Odd	测试参数 X 是否为奇数。
Pred	返回参数 X 的前一个值。
Succ	返回参数 X 的后一个值。

12. 指针与地址例程 (Pointer and address routines)

RTL 提供的指针与地址例程如表 1-14 所示。

表 1-14 指针与地址例程

名 称	功能说明
Addr	返回指定对象的地址。
CSeg	返回当前 CS 寄存器的值。
DSeg	返回当前 DS 寄存器的值。
Ofs	返回给定对象的偏移。
Ptr	将段地址和偏移转换成指针。
Seg	返回给定对象的段地址。
SPtr	返回当前 SP 寄存器的值。
SSeg	返回当前 SS 寄存器的值。

13. 字符串格式化例程 (String formatting routines)

RTL 提供的字符串格式化例程如表 1-15 所示。

表 1-15 字符串格式化例程

名 称	功能说明
FmtStr	将一系列变元格式化成字符串。
Format	格式化一系列变元, 返回 Pascal 格式的字符串。
FormatBuf	格式化数组参数, 将结果存入缓冲器。
StrFmt	格式化变元。
StrLFmt	格式化一系列变元。

14. Pascal 字符串操作例程 (String handling routines)

RTL 提供的 Pascal 字符串操作例程如表 1-16 所示。

表 1-16 Pascal 字符串操作例程

名 称	功能说明
AnsiCompareStr	基于当前 Windows 场景比较字符串并区分大小写。
AnsiCompareText	基于当前 Windows 场景比较字符串, 不区分大小写。
AnsiExtractQuotedStr	将一个引用的字符串转换为未引用字符串。
AnsiLowerCase	将字符串转换成小写形式。
AnsiPos	在某一字符串内部定义一个子字符串。
AnsiQuotedStr	返回某一字符串的引用版本。
AnsiUpperCase	将字符串转换成大写形式。
AppendStr	将字符串添加到另一字符串之后。
AssignStr	为字符串 PString 申请空间。
CompareStr	比较字符串, 区分大小写。
CompareText	依次序比较字符串, 不区分大小写。
Concat	合并两个或多个字符串成一个字符串。
Copy	返回字符串的子串或动态数组的一段。
Delete	在字符串中删除子串。
DisposeStr	释放一个字符串指针。
FmtLoadStr	从程序字符串资源表中装入字符串。
Insert	将一个字符串插入一个起始于特定指针的字符串。
IntToHex	返回一个整数的十六进制描述。
IntToStr	将整数转换成字符串。
IsValidIdent	测试字符是否为合法的 Pascal 标识符。
Length	返回字符串的字符数或数组的元素数。
LoadStr	从执行文件的资源中装入字符串。
LowerCase	将一个 ASCII 码字符串转换成小写字母。
NewStr	在堆空间中申请一个新字符串。
Pos	在给定字符串中返回特定子串首字符的索引值。
Str	格式化一个字符串并将它返回到变量。
StrToInt	将一个描述整数的字符串转换成数字。
StrToIntDef	将字符串转换成整数。
UpperCase	以大写形式返回字符串。
Val	将字符串转换成数字形式。

15. 零结束字符串操作例程 (Null-terminated string handling routines)

RTL 提供的零结束字符串操作例程如表 1-17 所示。

表 1-17 零结束字符串操作例程

名 称	功能说明
StrAlloc	为一个零结束字符串分配缓冲区，并返回首字符指针。
StrBufSize	计算字符串缓冲区中可存放的字符总数。
StrCat	将源字符串拷贝到目标字符串的尾部，进行字符串拼接。
StrComp	比较两个字符串。
StrCopy	将源字符串拷贝到目标字符串，并返回目标字符串。
StrDispose	释放字符串缓冲区。
StrECopy	复制字符串，返回指向字符串结尾的指针。
StrEnd	返回指向字符串结尾的指针。
StrlComp	比较两个字符串，不区分大小写。
StrLCat	向字符串中添加给定数目的字符。
StrLComp	比较字符串中给定个数的字符。
StrLCopy	复制字符串中给定个数的字符。
StrLen	返回字符串字符的个数（包括空结束符）。
StrLlComp	比较字符串中给定个数的字符，不区分大小写。
StrLower	将字符串转换成小写形式。
StrMove	向字符串中复制给定数目的字符。
StrNew	将字符串复制到分配空间的堆中，返回字符串指针。
StrPas	将零结束字符串转换成 Pascal 格式的字符串。
StrPCopy	将 Pascal 格式字符串复制到零结束字符串中。
StrPLCopy	将 Pascal 格式字符串中的字符复制到零结束字符串中。
StrPos	返回子串在字符串中首次出现的位置指针。
StrRScan	返回字符在字符串中最后出现的位置指针。
StrScan	返回特定字符在字符串中首次出现的位置指针。
StrUpper	以大写形式返回字符串。

16. 文本文件操作例程 (Text file routines)

RTL 提供的文本文件操作例程如表 1-18 所示。

表 1-18 文本文件操作例程

名 称	功能说明
Eoln	测试文件指针是否在行尾。
Flush	将文本文件缓冲区中的内容写入磁盘。
Read	从文件中读取数据。
Readln	从文件中读取一行文本。
SeekEof	返回文件尾状态。
SeekEoln	返回文件行尾状态。
SetTextBuf	给文本文件定义一个 I/O 缓冲区。
Write	对于类型文件，将变量写入文件单元中。 对于文本文件，将多个值写入文件中。
Writeln	向文本文件写入一个行结束符。