

HOPE

(共四册)

Turbo Pascal 6.0 库函数参考指南

其它
三种书
是

- 用户指南
- 程序员指南
- Turbo Vision 指南

金子方 编译

2

中国科学院希望高级电脑技术公司

Turbo Pascal 6.0

库函数参考指南

金子方 编译

中国科学院希望高级电脑公司

一九九一年三月

版权所有

不许翻印

违者必究

■ 北京市新闻出版局

准印证号： 3314 — 90314

■ 订购单位：北京 8721 信箱资料部

■ 电 话： 2562329

■ 电 传： 01—2561057

■ 电 挂： 0755

■ 地 址： 海淀影剧院北侧

■ 乘 车： 320、332、302路海淀黄庄下车

■ 办公地点： 公司大楼 101 房间

前 言

Turbo 序列软件问世以后即风靡全球，一直在高级语言编译和调试软件中占主导地位。继 Turbo Pascal 5.5 之后，Borland 公司于一九九零年底又推出了最新版本的 Pascal 编译调试器—Turbo Pascal 6.0。

一种新版本软件的诞生，总伴随着对原版本功能和性能的较大改进，Turbo Pascal 6.0 也不例外。对于初用 Turbo Pascal 的用户而言，完全不必重温 Pascal 的早期版本。内容翔实、刻划细腻、循序渐近的联机示教和各种介绍资料会使您学来如浴春风，得心应手。而 Turbo Pascal 的老用户也许会对新增加的功能更感兴趣。除了与早期版本完全兼容之外，Turbo Pascal 6.0 还提供了一种全新的集成开发环境(IDE)，它支持鼠标、多文件编辑和多重叠窗口，并配备了增强型调试工具和功能完备的嵌入式汇编器。Turbo Pascal 6.0 还支持 286 代码生成、远程过程调用，并能等同对待过程与函数。

为了适应程序设计风格的新潮流，Turbo Pascal 6.0 还在 5.5 版的基础上扩展了面向对象的程序设计(OOP)，允许使用私有域和私有方法，并支持在未知源级代码的情况下扩展目标代码的功能。此外，面向对象的应用工具 Turbo Vision 也是 Turbo Pascal 6.0 的一大新特色，它可以为您构造窗口应用程序，实现用户自己的 IDE。

为了满足广大用户的需要，我们根据有关资料编译了这套丛书。全套丛书包括《Turbo Pascal 6.0 用户指南》、《Turbo Pascal 6.0 程序员指南》、《Turbo Pascal 6.0 库函数参考指南》和《Turbo Vision 指南》。各书都具有自说明性，并侧重于 Turbo Pascal 的某个方面。

内容准确、版面清晰是我们所追求的目标。为了使读者更好地把握原文思想，我们对书中经常出现的术语进行了讨论、统一，尽可能准确地反映原编者的本意。相信这套丛书将成为读者的有力工具，我们将因此而感到欣慰。

本套丛书的出版得到了中科院希望高级电脑公司资料部秦人华同志的大力支持，在此表示真诚的谢意。

在本套丛书的编译过程中，虽然我们作出了很大的努力，尽量使文笔通顺、概念清晰，但仍存在不少错误，恳请广大读者谅解并指正。

编译者

一九九一年二月

于北京

目 录

第〇章 简介	(1)
本书内容	(1)
第一章 运行库函数	(2)
A	
Abs 函数	(2)
Addr 函数	(2)
Append 过程	(3)
Arc 过程	Graph 单元 (3)
ArcTan 函数	(4)
Assign 过程	(4)
AssignCrt 过程	Crt 单元 (5)
B	
Bar 过程	Graph 单元 (5)
Bar3d 过程	Graph 单元 (6)
BlockRead 过程	(7)
BlockWrite 过程	(8)
C	
ChDir 过程	(8)
Chr 函数	(9)
Circle 过程	Graph 单元 (9)
ClearDevice 过程	Graph 单元 (10)
ClearViewPort 过程	Graph 单元 (10)
Close 过程	(11)
CloseGraph 过程	Graph 单元 (11)
ClrEol 过程	Crt 单元 (12)
ClrScr 过程	Graph 单元 (12)
Concat 函数	(12)
Copy 函数	(13)
Cos 函数	(13)
CSeg 函数	(14)
D	
Dec 过程	(14)
Delay 过程	Crt 单元 (14)
Delete 过程	(14)
DelLine 过程	Crt 单元 (14)
DetectGraph 过程	Graph 单元 (15)
DiskFree 函数	Dos 单元 (16)
DiskSize 函数	Dos 单元 (16)
Dispose 过程	(16)
DosExitCode 过程	Dos 单元 (17)

DosVersion 函数	Dos 单元	(17)
Drawpoly 过程	Graph 单元	(18)
Dseg 函数		(18)

E

Ellipse 过程	Graph 单元	(19)
EnvCount 函数	Dos 单元	(19)
EnvStr 函数	Dos 单元	(20)
Eof 函数(正文文件)		(20)
Eof 函数(类型、无类型文件)		(20)
Eoln 函数		(21)
Erase 过程		(21)
Exec 过程	Dos 单元	(22)
Exit 过程		(23)
Exp 函数		(23)

F

FExpand 函数	DOS 单元	(23)
FilePos 函数		(24)
FileSize 函数		(24)
FillChar 过程		(24)
FillEllipse 过程	Graph 单元	(25)
FillPoly 过程	Graph 单元	(25)
FindFirst 过程	Dos 单元	(26)
FindNext 过程	Dos 单元	(27)
FloodFill 过程	Graph 单元	(27)
Flush 过程		(28)
Frac 函数		(28)
FreeMem 过程		(29)
FSearch 函数	Dos 单元	(29)
FSplit 过程	Dos 单元	(30)

G

GetArcCoords 过程	Graph 单元	(30)
GetAspectRatio 过程	Graph 单元	(31)
GetBkColor 函数	Graph 单元	(32)
GetCBreak 过程	Dos 单元	(33)
GetColor 函数	Graph 单元	(33)
GetDate 过程	Dos 单元	(34)
GetDefaultPalette 函数	Graph 单元	(34)
GetDir 过程		(35)
GetDriverName 函数	Graph 单元	(35)
GetEnv 函数	Dos 单元	(36)
GetFAttr 过程	Dos 单元	(36)
GetFillPattern 过程	Graph 单元	(37)
GetFillSettings 过程	Graph 单元	(38)
GetFTime 过程	Dos 单元	(38)
GetGraphMode 函数	Graph 单元	(39)
GetImage 过程	Graph 单元	(40)
GetIntVec 过程	Dos 单元	(41)

GetLineSettings 过程	Graph 单元	(41)
GetMaxColor 函数	Graph 单元	(42)
GetMaxMode 函数	Graph 单元	(42)
GetMaxX 函数	Graph 单元	(43)
GetMaxY 函数	Graph 单元	(43)
GetMem 过程		(44)
GetModeName 函数	Graph 单元	(44)
GetModeRange 过程	Graph 单元	(45)
GetPalette 过程	Graph 单元	(45)
GetPaletteSize 函数	Graph 单元	(46)
GetPixel 函数	Graph 单元	(46)
GetTextSettings 过程	Graph 单元	(47)
GetTime 过程	Dos 单元	(48)
GetVerify 过程	Dos 单元	(48)
GetViewSettings 过程	Graph 单元	(48)
GetX 函数	Graph 单元	(49)
GetY 函数	Graph 单元	(50)
GotoXY 过程	Crt 单元	(51)
GraphDefaults 过程	Graph 单元	(51)
GraphErrorMsg 函数	Graph 单元	(51)
GraphResult 函数	Graph 单元	(52)

H

Halt 过程		(53)
Hi 函数		(54)
HighVideo 过程	Crt 单元	(54)

I

ImageSize 函数	Graph 单元	(54)
Inc 过程		(55)
InitGraph 过程	Graph 单元	(55)
Insert 过程		(57)
InsLine 过程	Crt 单元	(58)
InstallUserDriver 函数	Graph 单元	(58)
InstallUserFont 函数	Graph 单元	(60)
Int 函数		(61)
Intr 过程	Dos 单元	(61)
IOResult 函数		(62)

K

Keep 过程	Dos 单元	(62)
KeyPressed 函数	Crt 单元	(62)

L

Length 函数		(63)
Line 过程	Graph 单元	(63)
LineRel 过程	Graph 单元	(64)
LineTo 过程	Graph 单元	(64)
Ln 函数		(65)
Lo 函数		(65)
LowVideo 过程	Ctr 单元	(66)

M

Mark 过程	(66)
MaxAvail 函数	(66)
McmAvail 函数	(67)
MkDir 过程	(67)
Move 过程	(68)
MoveRel 过程	Graph 单元 (68)
MoveTo 过程	Graph 单元 (69)
MsDos 过程	Dos 单元 (69)

N

New 过程	(70)
NormVideo 过程	Crt 单元 (70)
NoSound 过程	Crt 单元 (70)

O

Odd 函数	(70)
Ofs 函数	(71)
Ord 函数	(71)
OutText 过程	Graph 单元 (71)
OutTextXY 过程	Graph 单元 (73)
OvrClearBuf 过程	Overlay 单元 (73)
OvrGetBuf 函数	Overlay 单元 (74)
OvrInit 过程	Overlay 单元 (74)
OvrInitEMS 过程	Overlay 单元 (75)
OvrSetBuf 过程	Overlay 单元 (75)

P

PackTime 过程	Dos 单元 (76)
ParamCount 函数	(76)
ParamStr 函数	(77)
Pi 函数	(77)
PicSlice 过程	Graph 单元 (77)
Pos 函数	(78)
Pred 函数	(78)
Ptr 函数	(78)
PutImage 过程	Graph 单元 (79)
PutPixel 过程	Graph 单元 (81)

R

Random 函数	(81)
Randomize 过程	(82)
Read 过程(文本文件)	(82)
Read 过程(类型文件)	(83)
ReadKey 函数	Crt 单元 (83)
Readln 过程	(83)
Rectangle 过程	Graph 单元 (84)
RegisterBGIDriver 函数	(84)
RegisterBGIFont 函数	(86)
Release 过程	(88)
Rename 过程	(88)

Reset 过程	(88)
RestoreCrtMode 过程	Graph 单元 (89)
ReWrite 过程	(90)
RmDir 过程	(90)
Round 函数	(91)
RunError 过程	(91)

S

Sector 过程	Graph 单元 (91)
Seek 过程	(92)
SeekEof 函数	(92)
SeekEoln 函数	(93)
Seg 函数	(93)
SetActivePage 过程	Graph 单元 (93)
SetAllPalette 过程	Graph 单元 (94)
SetAspectRatio 过程	Graph 单元 (95)
SetBkColor 过程	Graph 单元 (96)
SetCBreak 过程	(97)
SetColor 过程	Graph 单元 (97)
SetDate 过程	Dos 单元 (98)
SetFAttr 过程	Dos 单元 (98)
SetFillPattern 过程	Graph 单元 (98)
SetFillStyle 过程	Graph 单元 (100)
SetFTime 过程	Dos 单元 (100)
SetGraphBufSize 过程	Graph 单元 (101)
SetGraphMode 过程	Graph 单元 (101)
SetIntVec 过程	Dos 单元 (103)
SetLineStyle 过程	Graph 单元 (103)
SetPalette 过程	Graph 单元 (104)
SetRGBPalette 过程	Graph 单元 (105)
SetTestBuf 过程	(107)
SetTextJustify 过程	Graph 单元 (108)
SetTextStyle 过程	Graph 单元 (109)
SetTime 过程	Dos 单元 (110)
SetUserCharSize 过程	Graph 单元 (110)
SetVerify 过程	Dos 单元 (111)
SetViewPort 过程	Graph 单元 (111)
SetVisualPage 过程	Graph 单元 (113)
SetWriteMode 过程	Graph 单元 (113)
Sin 函数	(115)
SizeOf 函数	(115)
Sound 过程	Crt 单元 (115)
Sptr 函数	(116)
Sqr 函数	(116)
Sqrt 函数	(116)
SSeg 函数	(116)
Str 过程	(116)
Succ 函数	(117)

Swap 函数	(117)
SwapVectors 过程	Dos 单元 (117)

T

TextBackground 过程	Crt 单元 (118)
TextColor 过程	Crt 单元 (118)
TextHeight 函数	Graph 单元 (119)
TextMode 过程	Crt 单元 (120)
TextWidth 函数	Graph 单元 (121)
Trunc 函数	(122)
Truncate 过程	(122)

U

UnpackTime 过程	Dos 单元 (122)
Ucase 函数	(122)

V

Val 过程	(122)
--------------	-------

W

WhereX 函数	Crt 单元 (123)
WhereY 函数	Crt 单元 (123)
Window 过程	Crt 单元 (124)
Write 过程(文本文件)	(124)
Write 过程(类型文件)	(126)
Writeln 过程	(126)

第〇章 简介

本书包含所有 Turbo Pascal 库例程的定义，并用例子来说明如何使用这些过程与函数。

如果你是初学 Pascal 程序设计，请先阅读《用户指南》。该书详细说明了 Turbo Pascal 的许多特性，并概述了本套手册四本书的内容。在《用户指南》中还将学习有关 IDE、项目管理器、编辑器以及命令行编译器的参考信息。

《程序员指南》说明了 Pascal 语言的 Turbo Pascal 实现，还讨论了一些高级的程序设计问题。运行时刻和编译时刻的错误信息在其附录 A “错误信息”中。

本书内容

本书按字母顺序列出了所有的 Turbo Pascal 库过程与函数。对每一个过程和函数都给出了语法、说明、返回值、有关的例程以及演示如何使用该例程的例子。

第一章 运行库函数

本章详细说明 Turbo Pascal 6.0 的所有过程和函数。以下样例说明了如何查找关于 Turbo Pascal 过程和函数的详细信息。

过程或函数名	所在单元
功 能	该过程或函数的作用。
用 法	该过程或过程是如何声明的。
结果类型	返回值的类型（如果为函数的话）。
说 明	关于该过程或函数的一般信息。
限 制	特殊要求或要了解的特定项。
参 见	相关的过程和函数。
例 子	说明如何使用该过程和函数的例子。

Abs 函数

功 能	返回表达式的绝对值。
用 法	Abs(x)
结果类型	和参数类型一样。
说 明	x是整型或实型表达式。结果值和x的类型一样，是x的绝对值。
例 子	

```
var
    r:real;
    i:integer;
begin
    r:=Abs(-2.3);    {2.3}
    i:=Abs(-157);   {157}
end.
```

Addr 函数

功 能	返回指定对象的地址。
用 法	Addr(x)
结果类型	指针。
说 明	x是任何变量、过程或函数标识符，结果指向x的指针。Addr结果和所有指针类型赋值兼容。
注 意	@操作和Addr产生相同结果。
参 见	Ptr,Ofs,Seg
例 子	

```
var P:pointer;
begin
```

```

P:= Addr(P); {Now point to itself}
end.

```

Apend 过程

功 能	为拼加而打开一个已存在的文件。
用 法	Append(Var f:text)
说 明	f必须是通过Assign和一个外部文件联接的正文文件变量。 Append用赋给f的名打开存在的外部文件。如果不存在给定的文件，产生错误。如果f已经存在，就先关闭再重新打开它。当前文件指针指向文件尾。 如果Ctrl-Z(ASCII码为26)在最后128字节出现，文-件指针指向块中的第一个Ctrl-Z。这样，文件信息加到以Ctrl-Z为结尾的文件中。 如果f赋以空名，比如： <pre>Assign(f,'')</pre> 调用Append后，f指向标准输出文件(标准句柄号为1)。 调用Append, f只能写。当前文件指针位于文件末尾。 用{\$I-}编译时，如果操作成功IoResult返回零，否则返回非零错误码。
参 见	Assign、Close、Reset、Rewrite。
例 子	

```

var f:text;
begin
  Assign(f,'TEST.TXT');
  Rewrite(f);    {Create new file}
  Writeln(f,'Original text');
  Close(f);      {Close file,save changes}
  Append(f);     {Add more text onto end}
  Writeln(f,'appended text');
  Close(f);      {Close file,save changes}
end.

```

Arc 过程

Graph 单元

功 能	以(x,y)作为圆心，从起始角到终止角画一条圆弧。
用 法	Arc(x,y:integer; StAngle,EndAngle,Radius:word)
说 明	以Radius为半径绕(x,y)画圆弧。弧从StAngle到EndAngle，用当前颜色画。每个驱动程序都包含被Circle、Arc和PieSlice所引用的纵横比例因子。 StAngle为0, EndAngle为360将画一完整的圆。Arc、Ellipse和PieSlice的角是逆时针方向计算的。最后一个调用的信息可用GetArcCoords查到。
限 制	在图形模式中使用。
参 见	Circle、Ellipse、FileEllipse、GetArcCoords、GetAspectRatio、PieSlice、Sector、SetAspectRatio。

例 子

```
uses Graph;
var
  Gd,Gm:integer;
  Radius:integer;
begin
  Gd:= Detect;
  InitGraph(Gd,Gm,'');
  if GraphResult <> grOK then
    Halt(1);
  for Radius:= 1 to 5 do
    Arc(100,100,0,90,Radius * 10);
  Readln;
  CloseGraph;
end.
```

ArcTan 函数

功 能 返回参数的反正切值。

用 法 ArcTan(x:real)

结果类型 实数。

说 明 x是实型表达式，结果是arctg(x)的值，以弧度为单位。

参 见 Cos,Sin

例 子

```
var r:real;
begin
  r:= ArcTan(P);
end.
```

Assign 过程

功 能 把外部文件名赋给一个文件变量。

用 法 Assign(var f; name:string)

说 明 f是一个文件类型的变量，name是字符串表达式。所有关于f的操作将在以 name 为名的外部文件上进行。

调用Assign后，f和外部文件的联结，直到对f再次使用Assign后才脱离脱离。

文件名由多层目录和文件名组成，如：

Drive:\DirName\...\DirName\FileName

如果路径以反斜杠开始，表示从根目录开始。否则，从当前目录开始。

Drive是盘驱动器标识符(A~Z)。如果省略驱动器号和冒号，则使用缺省驱动器。

\DirName\...\DirName是到文件名的根目录和子目录串，FileName由一个到八个字符组成，后跟由可选的句点和三个字符组成的扩展名。

路径名的最大长度是79个字符。

当文件名是空串时，f和标准输入、输出文件联接。这些特殊文件允许程序利用 DOS 操作系统的输入输出重定向。如果赋以空名，调用 Reset(f)，f 和标准输出联接；调用 Rewrite(f)，f 和标准输出联接。

限制 Assign不能用于已打开的文件。
参见 Append,Close,Reset,Rewrite。
例子

```
var f:text;
begin
  Assign(f, '');      { Standard Output }
  Rewrite(f);
  Writeln(f,'Standard output ...');
  Close(f);
end.
```

AssignCrt 过程

Crt 单元

功能 把正文文件和Crt联接起来。
用法 AssignCrt(var f:Text)
说明 除没有指定文件名外，AssignCrt和标准过程Assign的作用相同，只是正文文件和 Crt 联接。
这使得比标准文件更快地输入、输出。

例子

```
uses Crt;
var f:Text;
begin
  Write('Output to Screen or Printer [S,P]?');
  if UpCase(Readkey)='P' then
    Assign(f,'PRN') { output to printer }
  else
    AssignCrt(f); { Output to screen,use fast CRT routines }
  Rewrite(f);
  Writeln(f,'Fast output via CRT routines...');
  Close(f);
end.
```

Bar 过程

Graph 单元

功能 用当前填充方式和颜色画长方条。
用法 Bar(x1,y1,x2,y2:integer)
说明 画出并填满长方形(用于条形图表)。
使用SetFillStyle和SetFillPattern设置的颜色和模式。调用深度为0的Bar3d可画出长方形轮廓。
限制 在图形模式中使用。

参 见 Bar3D、GraphResult、SetFillStyle、SetFillPattern、SetLineStyle
 例 子

```

uses Graph
var
  Gd,Gm :integer;
  I,Width :integer;
begin
  Gd:= Detect;
  InitGraph(Gd,Gm, '');
  if GraphResult <> grOK then
    Halt(1);
  Width:= 10;
  for I:= 1 to 5 do
    Bar(I * Width,I * 10,Succ(I) * Width,200);
  Readln;
  CloseGraph;
end.

```

Bar3d 过程

Graph 单元

功 能 用当前填充模式和颜色画三维长方体。

用 法 Bar3d(x1,y1,x2,y2:integer;Depth:word;Top:boolean)

说 明 使用由SetFillStyle和SetFillPattern定义的填充模式和颜色画出并填充三维的长方体。三维长方体的轮廓线由SetLineStyle和SetColor设置的当前线模式和颜色画线，Depth是三维图形的深度像素值。如果Top为真，在长方体上设置三维的顶面；如果为假，则没有三维的顶面（这可使若干个条形叠加在一起）。

典型的深度通常取长方条宽度的25%，如：

```
Bar3d(x1,y1,x2,y2,(x2-x1+1) div 4,TopOn)
```

下面是定义的常量：

```

const
  TopOn = True;
  TopOff = False;

```

限 制 在图形模式中使用。

参 见 Bar、GraphResult、SetFillStyle、SetFillPattern、SetLineStyle。

例 子

```

uses Graph;
var
  Gd,Gm:integer;
  y0,y1,y2,x1,x2:integer;
begin
  Gd:= Detect;
  InitGraph(Gd,Gm, '');

```



```

if GraphResult < > grOK then
    Halt(1);
y0:= 10;
y1:= 60;
y2:= 110;
x1:= 10;
x2:= 50;
Bar3d(x1,y0,x2,y1,10,TopOn);
Bar3d(x1,y1,x2,y2,10,TopOff);
Readln;
end.

```

BlockRead 过程

功 能 将一个和多个记录读到变量中。

用 法 BlockRead(var f:file; var buf; count:word [,var result:word])

说 明 f是无类型文件变量，buf是任意变量，count是Word型表达式，result是Word型变量。

BlockRead从文件f读取count个和少于count个记录到内存，从buf的第一个字节开始存放。在可选的result中返回实际读取的完整记录个数。如果没有指定result，并且读取的数不等于count，则产生I/O错误。

转换后的整个块至少占用count×RecSize字节，其中RecSize是文件打开时指定的记录长度（缺省为128字节），如果count×RecSize大于65535(64K),则发生错误。

result是可选的参数，下面说明它如何工作：如果传送整个块，则返回结果值等于count。否则，如果result小于count，则在传送结束前就已到达文件尾部。在这种情况下，如果文件记录长度比1大，result返回读取的完整记录个数，最后部分记录有可能不包括在result中。

当前文件指针前进result个记录位置。

在{\$I-}状态下，如果操作成功则函数IoResult返回零；否则返回非零的错误码。

限 制 文件必须打开。

参 见 BlockWrite.

例 子

```

program CopyFile;
{ Simple ,fast file copy program with No error-checking }
var
    FromF,ToF:file;
    NumRead,NumWritten:word;
    buf:array [1..2048] of char;
begin
    Assign(FromF,ParamStr(1)); { open input file }
    Reset(FromF,1); { Record size = 1 }
    Assign(ToF,Paramstr(2)); { open output file }

```