
WATCOM C/386

使用指南

王一公 刘宝海 张 璞 (第四分册)

- 图形库函数参考手册
- C 语言参考手册

西安电子科技大学出版社

WATCOM C/386 使用指南

(第四分册)

王一公 刘宝海 张 璞

●图形库函数参考手册

●C 语言参考手册

西安电子科技大学出版社

1993.8

(陕)新登字 010 号

内 容 简 介

本书是一套全描述 WATCOM C/386 编译系统的手册。全书共分四册,第一分册描述了该软件包的详细列表,如何在 IBM PC 或 PS/2 的 DOS 系统下使用 WATCOM C/386 以及开发工具, DOS32 位及 Windows 32 位应用程序编程方法。第二分册着重介绍 VEDIO 和 LINKER 工具即如何在 DOS, 386 DOS Extenders, Microsoft Windows, OS/2 和 QNX 下调试及连接应用程序。第三分册描述了标准 C 库函数(美国国家 C 标准)以及专为 IBM 个人机设计的许多库函数。第四分册描述了 WATCOM 图形库函数以及 ANSI C 编程语言和 WATCOM C/386 的支持的扩充部分。

该书可作为从事系统软件及应用软件开发人员的参考书,同时也可作为大学计算机专业高年级学生的教材及教学参考用书。

WATCOM C/386 使用指南

(第四分册)

王一公 刘宝海 张 瑛

西安电子科技大学出版社出版发行

陕西省大荔县印刷厂印刷

新华书店经销

开本 787×1092 1/16 印张 17 2/16 字数 410 千字

1993 年 8 月第 1 版 1993 年 8 月第 1 次印刷 印数 1—3 000

ISBN 7-5606-0268-1/TP·0092

全套定价:100.00 元

封面设计：傅化群

合作出版单位：
陕西电子杂志社

地址：西安市西五路16副5号
邮编：710004
电话：794344 794391
电挂：8860

ISBN 7-5606-0268-1 / TP·0092 全套定价：100.00 元

前 言

近年来,随着 C 语言开发的广泛和深入,我们越来越感觉到,目前我国软件市场较为流行的编译器如 Microsoft 6.0 及 Borland C++ 等已不能满足广大用户的需要。比如,当您想在 IBM PC 及兼容机的 DOS 环境下开发大型的应用程序,特别是 Windows 应用程序时,往往会感觉到 DOS 640K 内存限制以及运行速度的困扰。再比如,目前市场上有许多很优秀的操作系统如 OS/2、QNX 以及目标系统如 Novell 网、16 位和 32 位 Windows 以及 AutoCAD,您无法找到一种编译器来支持所有这些操作和目标系统。因此我们觉得有责任向广大用户介绍一种崭新的真正 32 位的 C 编译系统——WATCOM C8.5/386。

该编译系统是由加拿大 WATCOM 系统软件公司以最新的美国国家标准 ANSI X3.159—1989 为基础于 1992 年推出的具有 32 位编程功能的软件包。在该软件包中,WATCOM 使用了当今世界最先进的目标代码优化及速度优化技术,不仅能不受 DOS 环境的限制,全面体现 386 的所有特性,而且提供了一套极为丰富的软件开发工具以及库函数。它所包含的 32 位程序开发软件包(包括 DOS 扩展器、32 位 Windows 开发工具、Windows SDK 部件及专用调试工具)可使用户轻而易举地突破 640K 的内存限制,使在 DOS 和 Windows 下开发的应用程序不仅能访问机器中的所有内存,而且程序的执行速度也会成倍提高。同时,它还为 OS/2、Novell's 网络、QNX 应用程序的开发配备了不同的编译、连接和调试工具,极大地方便了在不同环境下 C 应用程序的开发。尤其是它对 Novell's 网和 Net BIOS 所配备的远程服务器,将成为在网络环境下开发 C 程序不可缺少的工具。其它如原码级调试器、make 工具、touch 工具、bind 工具及反汇编服务程序等可为用户在 DOS 和 OS/2 下开发出高质量的 C 应用程序提供极大的帮助。

在库函数方面,WATCOM 兼容 Microsoft 6.0 所有的库函数,还增加了 32 位 Windows C 运行时的库函数,Windows API/DOS 接口库函数。同时还为数学库及图形库增加了很多新的内容,使得它能比现有的编译器支持更多的适配器及图形模式。

通过对 WATCOM C/386 的消化及使用,我们体会到:该编译系统代表了 C 语言程序开发及应用的发展方向,真正地实现了从 16 位到 32 位,从 640K 到 32M,从单系统到多系统支持的飞跃,的确是一种功能强大、使用方便的编译系统,它的应用和普及必将会使我国的 C 语言程序的开发技术上升到一个更高的阶段。因此,由西安三强技术开发公司组织,联合院校、科研所的专业软件工作人员编写了此书。该书由以下九个部分组成,分为四册编写,每一分册包含内容如下:

第一分册:

《入门必读》

《优化编译器及工具库用户手册》

《DOS 32 位应用程序编程指南》

《Windows 32 位应用程序编程指南》

2/5 104/07

第二分册:

《VEDIO(调试器)用户指南》

《LINKER(连接器)用户指南》

第三分册:

《库函数参考手册》

第四分册:

《图形库函数参考手册》

《C 语言参考手册》

本书由王一公、刘宝海、张璞同志担任主编。

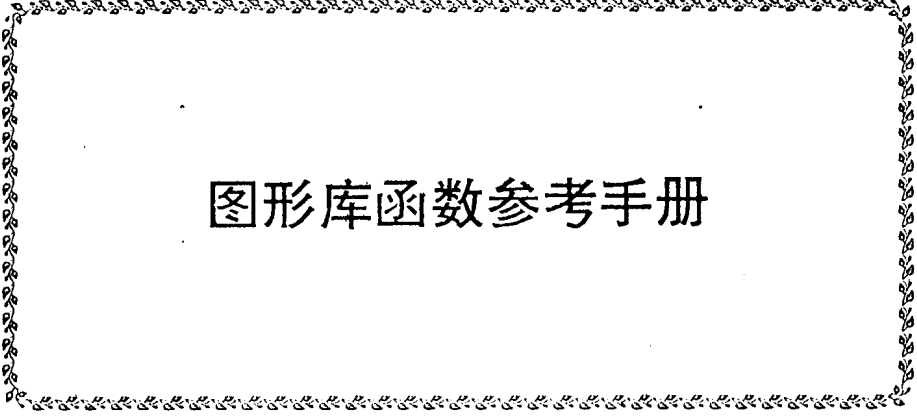
参加本书编写工作的有西安交通大学的付晓东、王爱群同志和西安石油学院的王鸿飞、阙征、王魁生、王江萍、于元华、胡洪涛同志。

西安三强技术开发公司的岳军、云振刚同志参加了本书的审定工作,在此衷心地表示感谢。

由于时间仓促,人力有限,如有缺点和不足之处,敬请广大用户给予批评。

编 者

1992 年 11 月



图形库函数参考手册

目 录

第一章 图形库.....	6
1.1 图形函数	6
1.2 图形适配器	6
1.3 坐标系统	6
1.4 颜色值	7
1.5 图像	7
1.6 文字	8
1.7 图形函数一览表	8
1.8 图形结构及常量	10
第二章 图形函数	15
_are, _are_w, _are_wxy 函数	15
_clearscreen	17
_displaycursor	18
_ellipse, _ellipse_w, _ellipse_wxy 函数	19
_floodfill, _floodfill_w 函数	20
_getactivepage	21
_getarcinfo	22
_getbkcolor	23
_getcliprgn	24
_getcolor	25
_getcurrentposition, _getcurrentposition_w 函数	26
_getfillmask	27
_getimage, _getimage_w, _getimage_wxy 函数	28
_getlinestyle	29
_getphyscoord	30
_getpixel, _getpixel_w 函数	31
_getplotaction	32
_gettextcolor	33
_gettextcursor	33
_gettextextent	34
_gettextposition	35
_gettextsettings	37
_gettextwindow	38
_getvideoconfig	39

_getviewcoord, _getviewcoord _ w, _getviewcoord _ wxy 函数	40
_getvisualpage	41
_getwindowcoord	42
_grstatus	43
_grtext, _grtext _ w 函数	44
_imagesize, _imagesize _ w, _imagesize _ wxy 函数	45
_lineto, _lineto _ w 函数	46
_moveto, _moveto _ w 函数	47
_outmem	48
_outtext	49
_pie, _pie _ w, pie _ wxy 函数	50
_polygon, _polygon _ w, _polygon _ wxy 函数	52
_putimage, _putimage _ w 函数	54
_rectangle, _rectangle _ w, _rectangle _ wxy 函数	55
_remapallpalette	56
_remappalette	58
_scrolltextwindow	59
_selectpalette	60
_setactivepage	61
_setbkcolor	62
_setcharsize, _setcharsize _ w 函数	63
_setcharspacing, _setcharspacing _ w 函数	64
_setcliprgn	65
_setcolor	66
_setfillmask	67
_setlinestyle	68
_setpixel, _setpixel _ w 函数	69
_setplotaction	71
_settextalign	72
_settextcolor	73
_settextcursor	74
_settextorient	76
_settextpath	76
_settextposition	78
_settextrows	79
_settextwindow	80
_setvideomode	81
_setvideomoderows	83
_setvieworg	84

_setviewport	84
_setvisualpage	85
_setwindow	86
_wraon	88
第三章 图形示例程序	89
3.1 显示“WATCOM C GRAPHICS”	89
3.2 画棒图形和饼图形	93

第一章 图 形 库

1.1 图 形 函 数

WATCOM C 图形库是由许多函数组成的。这些函数可用来在计算机屏幕上显示像直线和圆那样的图形画面,这些函数在图形输出的同时还可以显示文本信息。

1.2 图 形 适 配 器

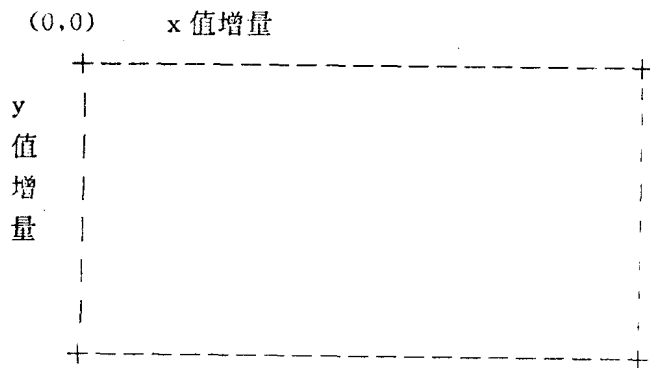
可支持具有下列图形适配器的计算机彩色或黑白显示器:

- IBM 黑白显示/打印适配器(MDPA)。
- IBM 彩色图形适配器(CGA)。
- IBM 增强型图形适配器(EGA)。
- IBM 多色图形阵列(MCGA)。
- IBM 视频图形阵列(VGA)。
- 大力神单色适配器。
- 由许多厂商提供的 super VGA 适配器(SVGA)。

1.3 坐 标 系 统

可以把显示屏当做含有许多点或称为图素(图形元素)的一个矩形。计算机的硬件决定了显示图素的行数和列数(可用_getvideoconfig 函数来了解这些数值)。

通过确定点的行坐标(x)和列坐标(y)就可访问每个图素,这些坐标一般放在圆括弧中(x,y)。按习惯,用(0,0)表示的实际坐标原点,它位于显示屏左上角的顶点处。原点右边的图素具有一个正值的 x 坐标。而原点下边的图素则有一个正值的 y 坐标。x 坐标和 y 坐标均不可能是负值。



可以通过在实际坐标系统上将原点从屏幕左上角移动到任意一个实际坐标点(参见 `_setvieworg` 函数)的方法来定义用户的观察坐标系统。在观察坐标系中就可以有负的 x 坐标和 y 坐标,观察和实际坐标的比例是完全一样的(都以图素为单位)。观察坐标值可从实际坐标中获得(见 `_getviewcoord` 函数),实际坐标值也可以从观察坐标中获得(见 `_getphyscoord` 函数)。

窗口坐标系统按照用户确定的值的范围来定义(见 `_setwindow` 函数)。这些值按比例映射到屏幕的实际坐标上,这就意味着对所有图像可不必注意其显示屏幕的分辨率(图素个数)。我们已有将窗口坐标转变为观察坐标的函数(见 `_getviewcoord` 函数),也有将观察坐标转换成窗口坐标的函数(见 `_getwindowcoord` 函数)。

1.4 颜色值

屏幕上每个点均有与其相对应的一个图素值,这个数值决定在该点上显示的颜色。可通过使用调色板(palette)来建立颜色,在理论上调色板是一个颜色表,在应用中,图素值为 n 时,就会在一个点上显示调色板中第 n 号的颜色。一般,调色板中的颜色数要比实际可能显示的颜色数要少。图形库可为每个点提供建立(见 `_setpixel` 函数)和报告(见 `_getpixel` 函数)图素值的函数。调色板的颜色可全部装入(见 `_remapallpalette` 函数)或一次只装入一种颜色(见 `_remappalette` 函数)。对视频方式来说可允许有多个调色板,用户可以选择其中任意一个调色板(见 `_selectpalette` 函数)供实际使用。

1.5 图 像

通过设置各独立点的图素值就可以产生一个高水平的图像。本库提供了画直线(见 `_lineto` 函数)、矩形(见 `_rectangle` 函数)、正多边形(见 `_polygon` 函数)、椭圆(见 `_ellipse` 函数)、椭圆弧(见 `_arc` 函数)和饼形图(见 `_pie` 函数)等功能。这些图形可绘其边框线或者使用当前颜色填充为实心图。画线和图形时使用的是已设置的当前颜色(见 `_setcolor` 函数)、当前线形(见 `_setlinestyle` 函数)和当前填充图案(见 `_setfillmask` 函数)。

画图函数要给图素设定图素值。在缺省状态下,这个设定值是用提供的图素值(通常指在调色板中当前颜色的索引号)来代替原先的图素值。另外,还可以用 `_setplotaction` 函数把原先的值与提供的值进行“与”运算(AND)、“或”运算(OR)或“异或”运算(exclusive OR)来计算图素的替换值,这称为当前进行画图的操作(currentplotting action),可在程序中用 `_getplotaction` 函数来决定这个操作。

可把图像保存起来(见 `_getimage` 函数),而后进行显示(见 `_putimage` 函数)。

有些显示方式允许有多个图形页面(graphic pages)(在图形硬件中的存储器用来存放图像信息)。可以选取其中任意一页用来在屏幕上显示(参见 `_setvisualpage` 函数),或者生成任一包含有图像的页面(见 `_setactivepage` 函数),这些页面不必是同一页面,可用 `_getvideoconfig` 函数来确定图像页面号。

1.6 文 字

用图形库可显示两类文字。一种是利用 BISO 函数进行文字输出。它可在任何显示模式下进行显示。这些文字是用 `_outtext` 和 `_outmem` 函数来显示。其文字的输出生位置是上次所显示的最后一个字符的下一个位置,也可重新设置(见 `_settextposition` 函数)。可以生成文字窗口(见 `_settextwindow` 函数),文字就在该窗口中卷屏。文字是以当前颜色来显示的(见 `_settextcolor` 函数)。

另一类文字可用 `_grtext` 函数显示。图形文字像一系列的线段那样来显示,而且可以用不同的尺寸(见 `_setcharsize` 函数)、不同的方向(见 `_textorient` 函数)及基准(见 `_settextalign` 函数)来绘出。

1.7 图形函数一览表

定义了下列函数:

<code>_arc, _arc _w, _arc _wxy</code>	画弧。
<code>_clearscreen</code>	清屏,用背景色填充屏幕。
<code>_displaycursor</code>	决定在执行完一个图形函数后是否显示光标。
<code>_ellipes, _ellipes _w, _ellipes _wxy</code>	画椭圆。
<code>_floodfill, _floodfill _w</code>	用当前颜色填充屏幕上一个区域。
<code>_getactivepage</code>	获得当前活动(操作)的图形页面号。
<code>_getarcinfo</code>	获得最近所画弧的终点坐标。
<code>_getbkcolor</code>	获得背景色。
<code>_getcliprgn</code>	获得当前限定区域的边界。
<code>_getcolor</code>	获得当前颜色。
<code>_getcurrentposition, _getcurrentposition _w</code>	获得当前输出位置的坐标。
<code>_getfillmask</code>	获得当前填充图案。
<code>_getimage, _getimage _w, _getimage _wxy</code>	把屏幕上的图像存入内存。
<code>_getlinestyle</code>	获得当前线型。
<code>_getphyscoord</code>	获得观察坐标系中某一个点的实际坐标值。
<code>_getpixel, getpixel _w</code>	获得指定位置图素的颜色值。
<code>_getplotaction</code>	获得当前进行画图的操作。
<code>_gettextcolor</code>	获得用于显示文字的颜色。
<code>_gettextcursor</code>	获得文字光标的图案。
<code>_gettextextent</code>	获得一个图形文字串边界的矩形框。
<code>_gettextposition</code>	为文字获得当前输出位置。
<code>_gettextsettings</code>	获得用于显示图形文字的当前设置的有关信息。

<code>_gettextwindow</code>	获得当前文字显示窗口的边界。
<code>_getvideoconfig</code>	获得有关图形配置的信息。
<code>_getviewcoord, _getviewcoord _w, _getviewcoord _wxy</code>	获得实际坐标中一个点在观察坐标系中的坐标。
<code>_getvisualpage</code>	获得当前可见图形页面的页号。
<code>_getwindowcoord</code>	获得观察坐标系中一个点的窗口坐标。
<code>_grstatus</code>	获得最近一次调用图形函数的状态。
<code>_grtext, _grtext _w</code>	显示图形文字。
<code>_imagesize, _imagesize _w, _imagesize _wxy</code>	获得图像屏幕的尺寸。
<code>_lineto, _lineto _w</code>	从当前位置到指定位置画一条直线。
<code>_moveto, _moveto _w</code>	设置当前输出位置。
<code>_outmem</code>	显示一个预定长度的文字串。
<code>_outtext</code>	显示一个文字串。
<code>_pie, _pie _w, _pie _wxy</code>	画一个饼形图。
<code>_polygon, _polygon _w, _polygon _wxy</code>	画一个正多边形。
<code>_putimage, _putimage _w</code>	把一个图像从内存中显示到屏幕上。
<code>_rectangle, _rectangle _w, _rectangle _wxy</code>	画一个矩形。
<code>_remapallpalette</code>	为所有图素设定颜色值。
<code>_remappalette</code>	为一个图素设定颜色值。
<code>_scrolltextwindow</code>	对文字窗口中的内容卷屏。
<code>_selectpalette</code>	选取一个调色板。
<code>_setactivepage</code>	设定活动的操作的图形页面(可将图形目标画到该页上)。
<code>_setbkcolor</code>	设定背景颜色。
<code>_setcharsize, _setcharsize _w</code>	设定用于显示图形文字的字符大小。
<code>_setcharspacing, _setcharspacing _w</code>	设定用于显示图形文字的字符间距离。
<code>_setcliprgn</code>	设定限定区域的边界。
<code>_setcolor</code>	设定当前颜色。
<code>_setfillmask</code>	设定当前填充图案。
<code>_setlinestyle</code>	设定当前线型。
<code>_setpixel, _setpixel _w</code>	在一个指定位置设定图素的颜色。
<code>_setplotaction</code>	设定当前进行画图的操作。
<code>_settextalign</code>	设定用于显示图形文字的基准。
<code>_settextcolor</code>	设定用于显示文字的颜色。
<code>_settextcursor</code>	设定文字光标的图案。

<code>_settextorient</code>	设定用于显示图形文字的方向。
<code>_settextpath</code>	设定用于显示图形文字的路径。
<code>_settextposition</code>	为文字设定输出位置。
<code>_settextrows</code>	在屏幕上设定显示文字的行号。
<code>_settextwindow</code>	设定用于显示文字区域的边界。
<code>_setvideomode</code>	选择要用的显示模式。
<code>_setvideomoderows</code>	选择要用的显示模式和文字行数。
<code>_setvieworg</code>	设定观察坐标系中用作坐标原点的位置。
<code>_setviewport</code>	设定限定区域的边界及观察坐标系中的坐标原点。
<code>_setvisualpage</code>	设定可见的图形页面(显示在屏幕上的页面)。
<code>_setwindow</code>	定义窗口坐标系统中的边界。
<code>_wrapon</code>	允许或者禁止在文字窗口中给文字加边框。

1.8 图形结构及常量

所有要用图形库的程序模块应该包含“graph.h”头文件,这个头文件中含有库中所有函数的原型以及它们所用的结构和常量。

“videoconfig”结构中含有关于当前显示模式和硬件配置的信息,这个结构中含有下列字段(所有字段均为短型):

<code>numxpixels</code>	x 轴上图素的个数。
<code>numypixels</code>	y 轴上图素的个数。
<code>numtextcols</code>	文字的列数。
<code>numtextrows</code>	文字的行数。
<code>numcolors</code>	有效颜色的个数。
<code>bitsperpixel</code>	图素值的位数。
<code>numvideopages</code>	显示页的个数。
<code>mode</code>	当前显示模式。
<code>adapter</code>	适配器类型。
<code>monitor</code>	监控器类型。
<code>memory</code>	显示存储器的千字节(1024个字符)的个数。

“xycoord”结构中含有屏幕上一个图素位置的坐标。这个结构所含字段如下(均为短型字段):

<code>xcoord</code>	位置的 x 分量。
<code>ycoord</code>	位置的 y 分量。

“_wxycord”结构中含有窗口坐标中一个位置的坐标。这个结构中所含的字段如下(均为双精度字段):

<code>wx</code>	位置的 x 分量。
<code>wy</code>	位置的 y 分量。

“rccoord”结构中含有屏幕上一个点的坐标(行和列),这个结构中所含的字段如下(均为短型字段):

row	文字在屏幕上的行号。
col	文字在屏幕上的列号。

“textsettings”结构中含有当前使用的图形文字显示的有关信息,这个结构所含字段如下(均为短型字段):

basevectorx	当前基本向量的 x 分量。
basevectory	当前基本向量的 y 分量。
path	当前文字路径。
height	当前文字高度(以图素计)。
width	当前文字宽度(以图素计)。
spacing	当前文字间距(以图素计)。
horizontal	当前文字基准的水平分量。
vertical	当前文字基准的垂直分量。

用来标识各种图形模式的常量有:

模式	说 明
_MAXRESMODE	最高分辨率的图形模式。
_MAXCOLORMODE	最多颜色的图形模式。
_DEFAULTMODE	恢复屏幕到初始状态。
_TEXTBW40	40×25文字,16级灰度。
_TEXTC40	40×25文字,16/8种颜色。
_TEXTBW80	80×25文字,16级灰度。
_TEXTC80	80×25文字,16/8种颜色。
_MRES4COLR	320×200,4种颜色。
_MRESNOCOLOR	320×200,4级灰度。
_HRESBW	640×200,2种颜色。
_TEXTMONO	80×25文字,黑白(BW)。
_HERCMONO	720×350,2种颜色。
_MRES16COLOR	320×200,16种颜色。
_HRES16COLOR	640×200,16种颜色。
_ERESNOCOLOR	640×350,4种颜色。
_ERESCOLOR	640×350,4或16种颜色。
_VRES2COLOR	640×480,2种颜色。
_VRES16COLOR	640×480,16种颜色。
_MRES256COLOR	320×200,256种颜色。
_URES256COLOR	640×400,256种颜色。
_VRES256COLOR	640×480,256种颜色。
_SVRES16COLOR	800×600,16种颜色。
_SVRES256COLOR	800×600,256种颜色。