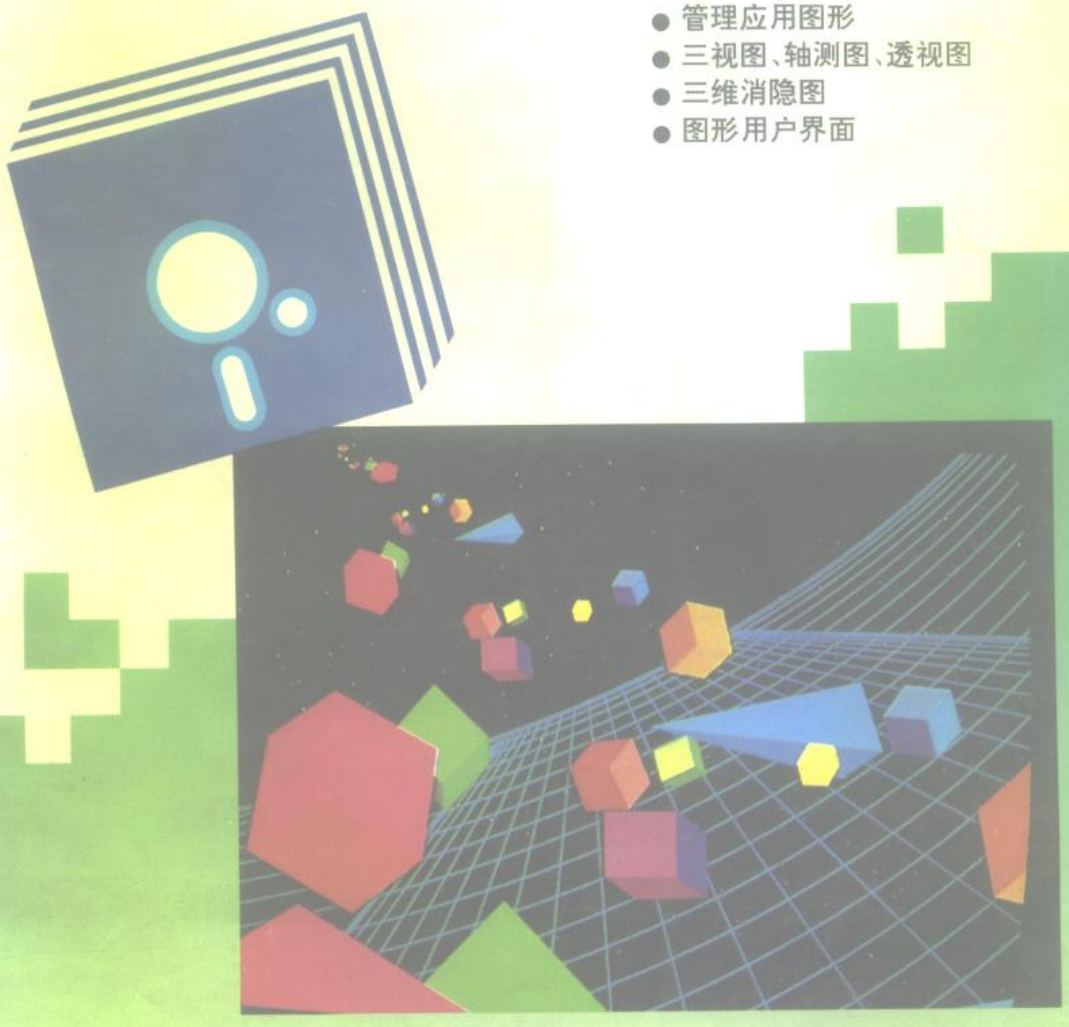
计算机绘图（软件）应用丛书

王永生　主编

最新C语言实用绘图程序与图形

- Turbo C 绘图基础
- 绘图程序设计方法、技巧
- 二维三维图形变换
- 分形及其应用
- 管理应用图形
- 三视图、轴测图、透视图
- 三维消隐图
- 图形用户界面



科学出版社

计算机绘图(软件)应用丛书

最新C语言实用绘图程序与图形

王永生　主编

科学出版社

1996

（京）新登字 092 号

内 容 简 介

本书全面系统地介绍了用 Turbo C 语言进行计算机绘图的基础知识及二维、三维、分数维图形变换原理，绘图程序设计，图形用户界面设计，二维和三维图形绘制的方法与技巧。书中文字、程序、图形并茂，内容新颖，实用性强。全书精选数十个完整的 Turbo C 绘图源程序（均上机通过），约 200 幅计算机绘制的精美图形和一个有立体感的图形用户界面。书中提供的程序有不少是多用途程序，开发难度较大，它可以解决一类计算机绘图问题；有经验者对书中程序稍加修改就能得到满足自己需要的绘图程序；书中的图形用户界面亦十分有用，可根据需要改成实际开发软件系统的用户界面。

全书共分八章。主要内容包括：图形输入输出设备、图形显示、绘图函数、绘图程序设计、绘图变换、三维图形消隐、花瓣图、螺旋图、圆形图、分数维图、管理图、展开图、工程图、曲面图、映射图、多个多面体图和用户界面等。

本书既可作为大专院校有关专业师生学习计算机绘图的教学和参考用书，亦可供工程技术人员从事计算机绘图工作参考使用。作者可提供书中全套程序软盘，需要者可与作者联系。

计算机绘图（软件）应用丛书

最新 C 语言

实用绘图程序与图形

王永生 主编

责任编辑 王淑兰

科学出版社出版

北京东黄城根北街 16 号

邮政编码：100717

中国科学院印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1996 年 1 月第 一 版 开本：787×1092 1/16

1996 年 1 月第一次印刷 印张：28 3/4

印数：1-4200 字数：935 000

ISBN 7-03-004594-7/TP · 416

定价：37.50 元

前　言

C 语言现在已成为人们使用最多的一种程序设计语言。特别是，Turbo C 以其功能齐备的集成开发环境和无与伦比的编译速度在各种版本的 C 语言中独占鳌头，风靡世界。Turbo C 语言操作十分简单、灵活；功能非常齐全、强大；编程调试格外方便、有效。不论对初次涉及 C 语言的编程者，还是对编写大型复杂软件的程序员，都会收到事半功倍的效果。一旦使用其集成开发环境，就不想在别的开发环境下工作。C 语言似乎有一统天下的趋势。

当前计算机绘图发展迅速，特别是 Turbo C，Turbo Pascal，Fortran，Basic 等程序设计语言都先后增加了绘图功能。这大大地促进了计算机绘图的开展和应用。Turbo C 不仅是面向图形的结构化程序，而且当使用 Borland 图形接口(BGI)时，就如同一颗图形巨星一样发挥其作用。国外用C 语言进行绘图或进行绘图系统设计、开发已经十分流行。国内亦同，计算机绘图已成为热门，越来越多的人对它发生兴趣，并且开展了广泛的实际应用工作；用C 语言进行计算机绘图正朝着广度与深度发展，创造着手工绘图无法比拟的效率和质量。尽管介绍计算机绘图的书已经面世，但是全面系统地讨论、提供 Turbo C 实用绘图程序与图形的书籍还几乎没有。为适应和促进国内计算机绘图的科研、教学和应用工作的开展，我们编写了这本《最新 C 语言实用绘图程序与图形》，以满足这方面读者的迫切需要。

本书具有以下特点：

1. 内容编排层次清晰，通俗易懂，由浅入深，非常适合自学。

2. 作者本着“加强基础、重视实际、立足应用、着眼发展”的原则，全面系统地讨论了 Turbo C 绘图的基础、原理、方法和技巧。全书内容实用，取材新颖，程序好读，图形精美。

3. 一般计算机绘图类图书，注重讲理论、算法，实际计算机绘图的程序、图形不多，有的甚至没有什么程序与图形。因此，初学者看完这类书后仍然感到不会用计算机绘制图形。本书不仅重视绘图程序设计的理论及算法介绍，同时又给出大量计算机绘图程序与图形。此外还注意程序思路清晰、逻辑性强、可读性好、模块结构和书写风格的培养。

本书将告诉读者如何用 Turbo C 语言进行计算机绘图，包括绘图程序编写、修改、调试以及绘图程序设计方法、技巧、风格等。无论读者在计算机绘图方面是没经验的新手，还是有经验的老手，书中的内容都会使读者发生浓厚的兴趣。如果读者是新手，使用的计算机设备支持 Turbo C(1.5 以上)程序设计语言，我们极力鼓励读者输入并运行书中的每个程序，并把屏幕的输出结果与书中给的图形相比较。这样做，读者会通过仿效、练习，从而学会用 Turbo C 语言进行计算机绘图，最后达到开发绘图程序(或图形系统)的较高水平。甚至可以随心所欲地创造性地绘制图形，走入美妙的计算机图形世界！如果读者是老手，本书的绘图程序将有益于读者的工作，事半功倍，或许还能从中受到启发，开阔视野，拓宽思路，因此同样很有参考、引用价值。总之，本书会使读者颇有“一旦拥有，别无所求”之感。

本书总计提供数十个完整的 Turbo C 绘图源程序。它们在 PC 及其兼容机上(不管用哪一种类型的视频适配器，如 CGA，MCGA，EGA，VGA 和 TVGA 等)都可在 Turbo C 的集成开发环境中运行。每个程序的图形初始化部分自动检测计算机的视频适配器与视频模式，并选择二者的最佳组合来初始化显示器屏幕。所有程序均在微计算机上调试、运行、通过，并且都有详细的程序分析、说明，以及附有程序运行结果打印输出的屏幕图形。图形输出相当漂

亮，书上仅取其中一部分，大约有 200 个图形。

全书共分八章。第一章概述 Turbo C 绘图基础，主要包括图形输入输出设备、图形显示、绘图函数和图形文本。第二章讲解绘图程序设计步骤、结构和方法（源程序风格、技巧等）。第三章讨论绘图变换，首先论述二维变换即二维几何变换、用户坐标到屏幕坐标的变换以及线段裁剪与多边形裁剪；然后阐述三维变换及三维几何变换和投影变换，并绘制了三视图、轴测图和透视图。第四章介绍二维图形花瓣图、螺旋图、圆形图、摆线图、展开图以及一般平面曲线图的绘制。第五章研究了近些年来出现的分数维图形即龙图、C 图、H 图、曼德勃罗特图、朱莉娅图等。这些图形新颖、有趣，引起很多人对它进行研究并加以应用。现在它们越来越多地用在造型环境中，用来设计壁纸、地板革图案和纺织品花样。本章将分数维图形映射到器皿上进行图案设计，效果极佳。第六章讨论了各种管理图形扇形图、直方图、甘特图、条形图、折线图等。第七章介绍三维图形绘制，其中包括图形数据结构、浮动水平线算法和画家算法；参数曲面、函数曲面和多个多面体图形的绘制等。第八章论述图形用户界面，连同前面章节的所有程序可构成一个实用图形软件系统。

全书由王永生主编。其中第二、四、七章及第三、五、六章大部分由王永生编写；第三、五章部分内容由刘静华编写；第八章由曹玉编写；第一章由李晓方编写；第六章第四节由张泽虹徐莉编写。另外，参加编写工作的还有翟桂英、刘佐英、胡允林、王询、薛春龙和周小伟。本书程序与图形均由所在章节的作者编写、调试、通过，并运行程序在计算机屏幕上绘出图形，然后打印得之。图中汉字显示均采用第八章汉字显示技术制作。书中全部插图由刘静华绘制，文字录入由李晓方完成，全书最后由王永生统稿。

本书包含了作者多年的工作与研究成果，是一本非常实用的计算机绘图教学和参考用书。在此，奉献给在校学生和广大科技人员。为便于读者对本书的学习和使用，我们可向读者提供全套程序软盘，需要者可与我们联系。

由于我们的水平有限，书中难免存在一些缺点与错误，恳请读者批评指正。

作　者

1995 年 6 月

目　录

第一章　Turbo C 绘图基础

第一节　图形输入输出设备

在一个计算机图形系统中，除了计算机之外，各种图形输入输出设备是必不可少的。近代的电子计算机是利用电信号来传输信息的，而图形输入设备是将用户的图形数据、各种命令等转换成电信号传送给计算机。同样，图形输出设备则将计算机处理好的结果转换成可见的图形呈现在用户面前。

这一节将分别介绍这些图形设备的功能及简单工作原理。

一. 图形输入设备

1. 键盘

键盘不仅是字符输入设备，它还是一种简单而又便宜的图形输入设备。键盘是由许多按键与相应的微动开关以及键盘编码器、寄存器等逻辑电路组成。键盘的盘面按照英文打字机设计，其操作与英文打字机相似，这就给使用者带来方便。使用者通过按动键盘上的按键，使相应的开关接通，便可向主机发出图形命令，也可输入图形数据。常用键盘形式如图 1-1 所示。

键盘上的键按其用途可分为三类。

(1)字符键

用以产生字符编码。字符编码是按 ASCII 编码(或其它的编码形式)进行的。

(2)功能键

包括程序功能键、编辑功能键及图形功能键。功能键被按动时，将产生相应的控制命令传送给计算机以完成程序控制功能、文本编辑功能、图形编辑及变换等功能。

(3)控制键

用于对系统的运行进行人工控制以完成各种中断操作、屏幕硬拷贝等操作。

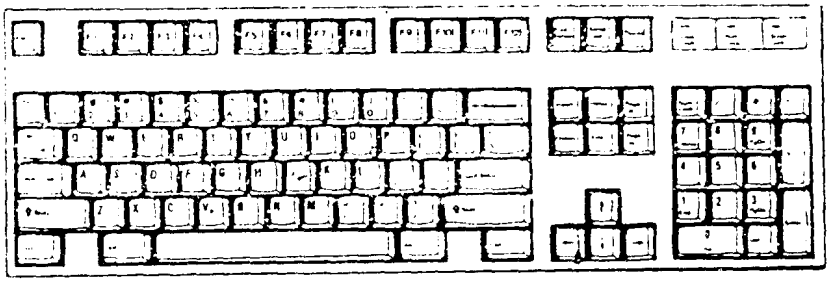


图 1-1　键盘

2. 鼠标器和电磁笔

鼠标器和电磁笔的形式见图 1-2 所示。其中有单按钮，三按钮和四按钮的鼠标器。鼠标器的位置对应屏幕上光标，移动鼠标器，并按一个适当的按钮，就会把光标在屏幕上的准确位置送入计算机中，用这种方法可实现选择、定位，重复这些操作即可在屏幕上产生一幅图形。鼠标器上的按钮可用于各种目的，这完全取决于应用程序。鼠标器工作起来要比键盘方便得多，但鼠标器要在内存中装入其驱动程序后才能工作。在购买鼠标器时，会得到该鼠标器的驱动程序 mouse.com 与 mouse.sys。要装入鼠标器驱动程序，只要将 mouse.com 放入 DOS 系统文件 autoexec.bat 中或者将 mouse.sys 放入 config.sys 中即可。每次启动计算机系统时，DOS 就会自动执行 config.sys 和 autoexec.bat 中的每个文件，从而将鼠标器驱动程序调入内存并驻留内存中。应用程序可以使用软件中断 Int33h 来与鼠标器通信，通过该中断，应用程序可以得到鼠标器光标位置、

按钮状态。鼠标器光标的形状可以改变，也可以可见或不可见。电磁笔的工作方式类似于鼠标器只是它用笔尖来实现定位和选择，这相当于单按钮鼠标器，但移动起来非常灵活，如同使用通常的笔一样方便。

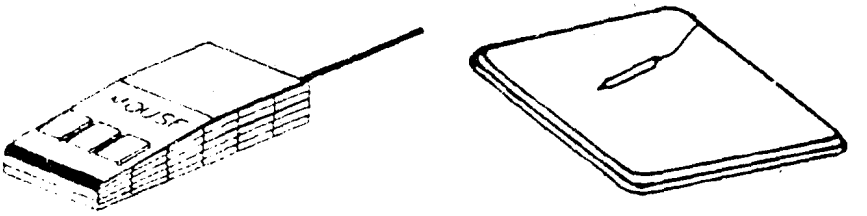


图 1-2　鼠标器和电磁笔

3. 图形输入板

图形输入板大部分是利用磁感应原理，在图形输入板下沿 x 和 y 方向上有许多平行的印制线。这样就将图形输入板划分成极小的一块块小方块，每一小方块就是一个像素。图形输入板的分辨率就是指一个像素或小方块的距离。图形输入板一般配有游标或电磁笔，用以输入字符，定位或拾取。当游标在板内移动时，通过按下游标上的功能开关，它在板上的绝对位置坐标就被送给计算机。若将图纸贴在图形输入板上，即可利用游标将图形上的点送入计算机，即图形的输入。还可以通过图形输入板上的菜单区将用户的专用图形库中的图形（如建筑中的门、窗、墙、卫生设备等，电子行业中的各种元件等）直接输入计算机系统，这样就大大地提高了输入效率。图形输入板的大小和分辨率多种多样，其有效幅面，小的是 $228\times304mm^2$，大的可达 $910\times1220mm^2$，可根据要输入图形的大小适当加以选择。利用图形输入板来输入图形数据十分方便，它是微机图形系统中理想的输入设备。常见的图形输入板如图 1-3 所示。

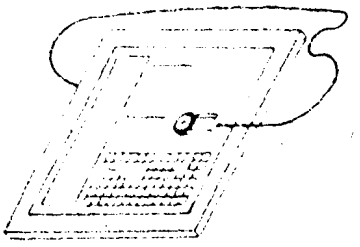


图 1-3　图形输入板

二. 图形输出设备

常用的图形输出设备一般可分为二大类。一类是与图形输入设备相结合，构成具有交互功能的可以快速生成和删改图形的显示系统；一类是在纸或其它介质上输出可以永久保存的图形的绘图系统。下面分别简单地加以介绍。

1. 显示设备

(1)随机扫描的图形显示器

这种显示器又称之为直接画线器或轨迹扫描设备。它的基本工作过程是：从显示文件存储器中取出画线指令或显示字符指令，方式指令（如亮度、线型等），送到显示控制器，由显示控制器控制电子束的偏转，轰击荧光屏上的荧光粉，从而出现一条发亮的图形轨迹，由于这种显示器一般使用的都是低余辉的荧光粉，因此这个过程需要每秒至少 30 次的频率重复进行，否则图形就会出现闪烁。

(2)存储管式的图形显示器

由上面知，随机扫描的显示器使用了一个独立的存储器来存储图形信息，然后不断地取出这些信息来刷新屏幕，这样带来了显示稳定图形时画线长度有限且造价较高等缺点。针对这些缺点，发展了利用管子本身存储信息的技术、这就是存储管技术。表面上看存储管的特性象是一个有极长余辉的阴极射线管，一条线一旦写在屏幕上，在一小时之内都将是可见的；从内部结构看，存储管也类似于阴极射线管，因为它们都使用类似的电子枪、聚焦和偏转系统，在屏幕上都有某些类似的荧光涂层。然而这种显示器的电子束不是直接打在荧光屏上，而是先用写入枪将图形信息“写”在一个细网栅格上，栅格上涂有绝缘材料，栅网装在靠近屏幕的后面，其上有由写入枪画出的正电荷图形。还有一个独立的读出电子枪，有时称作泛流枪，它发出连续的低能电子流把存储栅网上的图形“重写”在屏幕上。

(3)光栅扫描的图形显示器

这种图形显示器是基于电视的工作原理。在阴极射线管中装的水平（行）偏转线圈和垂直（场）偏转线圈，

前者产生一个水平磁场，后者产生一个垂直磁场，在其中通以线性变化的电流，因而产生的磁场也是线性变化的。电子束在水平偏转线圈产生的水平磁场作用下，沿着水平方向扫描，称为“水平扫描”或“行扫描”；在垂直偏转线圈所产生的垂直的磁场作用下，电子束沿着垂直方向扫描，称为“垂直扫描”或“场扫描”。通常电视机每幅图象扫描 512×512 个点，为了使图象不闪烁，要求一秒钟产生 30 幅(帧)图象，即每秒要扫描 30×512×512 个点，每一个点通常称为一个象素。每一个象素可以具有不同的灰度及颜色等属性，这些属性被存储起来并提供了修改的方便。这样存储器的容量是十分庞大的，假如一个象素只有亮或暗二种属性的话，那么每个象素至少要一个二进制位来标志它的亮、暗状态，整个屏幕就要 512×512 位的存储空间(即 256K 位)。实际上，许多光栅扫描的显示器设定 8 位或更多位来表示象素的灰度，即一个象素可以选择 $2^8=256$ 种灰度等级来显示。颜色常选择 $2^6=64$ 种。这样，要保留屏幕上所有点的属性就需要一个相当大的存储器，通常称之为帧缓冲存储器。在存储器的价格相当昂贵时，使用光栅扫描的图形显示设备几乎没有实际的可能。随着集成电路技术的迅速发展，性能可靠的大容量半导体存储器的价格大幅度下降，这才使得光栅扫描的显示器真正投入实用，并越来越普遍。光栅扫描的显示器具有随机扫描的显示器和存储管式的显示器所没有的优点，突出的一点是它不仅可以显示物体的轮廓线，特征线等所谓的线图形，而且可以显示被多种灰度和色调的象素所填充的所谓的面图形，这就使得输出具有真实感的立体图形成为可能。目前微机上所配备的显示器几乎全部为光栅扫描的图形显示器。

2. 绘图设备

(1)滚筒式绘图仪

这种绘图仪是用两只电机分别带动绘图纸和绘图笔运动，从而产生图形轨迹。绘图纸卷在筒上，由电机带动沿着一个方向(±x 方向)运动，而绘图笔架在纸筒上方，由另外的电机带动，沿垂直于纸运动的方向(±y 方向)运动。笔架上带有几种颜色的笔，从而可以画出不同颜色的图形。这种绘图仪结构简单、价格便宜、易于操作，但是精度、速度不可能太高，画出线条会呈锯齿状，而且要求用标准纸。

(2)平板式绘图仪

这种绘图仪的特点是绘图纸平铺在绘图平台上，平台板面从 $200\times300mm^2$ 到 $1800\times5500mm^2$，甚至长达十几米不等。平板式绘图仪通常可以分为二种，一种是步进马达驱动、机械传动的绘图仪，一种是平面电机驱动的绘图仪。机械传动的绘图仪的传动装置一般用钢线绳或齿轮齿条箱，需在 x、y 两个方向上用步进电机拖动笔架运动。笔架装在一根笨重的横梁上，因而这种绘图仪的速度和加速度都比较低。国产的这类绘图仪的速度，为每分钟几米，国外的这类绘图仪可以达到十几米。由于使用机械传动，这类绘图仪的精度容易下降，寿命较短，但价格相对比较便宜。平面电机驱动的绘图仪采用了两轴同时驱动的单向脉冲电机，电机的可动部分(动子或 x-y 马达)与不动部分(定子或天花板)均为平板，因此习惯上称之为平面电机。这种绘图仪的可动部分(动子)重量比较轻，动子和定子之间采用空气轴承(即运动时动子和定子互相不接触，其间留有十几微米的气垫间隙)，因此可以产生比较高的速度和加速度，大大缩短了绘图时间。由于绘图笔架或刻图刀架直接装在动子上，省去了机械传动机构，可以减少由于机械传动引起的误差及烦琐的维修工作，又因它长期工作也不致引起太大的精度降低，延长了机器的使用寿命，绘图纸一般以真空吸附或静电吸附的方式固定在绘图台板上，对绘图纸没有特殊的要求。但是这种绘图仪与滚筒式绘图仪相比，价格相对要昂贵一些。

(3)静电绘图仪

这种绘图仪是打印机和给图仪的结合，它的运动部分很少，只有供纸和调色盒是机械运动，其余都是电子线路。它的工作原理是，当程序控制的电压作阵列式输出，并作用在管头的管针尖上时，被选中的针尖就在管头下面通过的纸上产生极小的静电点，然后纸暴露在液态的调色盒下，产生图形或字符。这种设备的绘图速度与图形的复杂性无关，但绘图过程对计算机是一个很重的负载，因此常常采用脱机方式工作。这种绘图仪能够输出具有明暗度的面图形，分辨率较高，而且可靠安静，但线条有锯齿状，用纸特殊且贵。

3. 图形打印机

从屏幕上看到的计算机绘出的图形是非常漂亮的，若不能将如此漂亮的图形在纸上打印出来的话，那就太令人失望了。为此下面介绍二种常用的图形打印机。

(1)点阵打印机

点阵打印机是与微计算机配合使用最多的打印机。它的工作方式是，以针击打色带，从而在打印纸上得

到一系列的点。如果将计算机处理过的图形信息送到打印机的控制部分来控制打印机针头的运动，便可将图形打印在纸上。但由于打印机打印的点的密度有限，所以打印出的图形光滑程度不够理想；同时纸的幅面也受到限制，故点阵打印机打印图形只适用于对输出图形质量要求不高的情形。

(2)激光打印机

激光打印机是一种高性能的字符、图形输出设备。它利用光学原理，空间分辨率特别高，在每平方英寸①的范围内可达到 600×600 点的密度。激光打印机可产生达到印刷质量的图形，它打印出的图形几乎和绘图笔画的图形一样光滑。一般屏幕图形输出到激光打印机上和输出到点阵打印机上差不多，但激光打印机的性能和处理方式略有些不同。激光打印机比起点阵打印机来机械约束少，打印出的图形质量高。大多数激光打印机支持四个分辨率，不同的分辨率并不影响输出速度。目前，激光打印机广泛用于印刷及文字处理行业。由于激光打印机输出的幅面不能太大，所以在输出图形时只用于较小范围。本书程序的图形均为激光打印机和点阵打印机输出的图形。

第二节　字符屏幕

标准 C 语言没有定义字符屏幕和图形函数，而 Turbo C 提供了这些函数。这就大大地方便了编程，因为它补充了相当多的功能以适应任何用途。实际上，完整的字符屏幕和图形功能函数是最成功的商用软件所必须的。字符屏幕函数的核心是窗口(Window)，它是屏幕的活动部分、字符输出或显示在活动窗口中进行，窗口在缺省时候，就是整个屏幕。窗口可以根据需要，由用户指定其大小。同样，对图形函数的操作，Turbo C 提供了视口(Viewpoint)，也就是说图形函数的操作都是在视口上进行。图形视口与字符窗口具有相同的特性，用户可以在屏幕上定义大小不同的视口，若不定义视口大小，它就是整个屏幕。

窗口是在字符屏幕状态下的概念，只有字符才能在窗口中显示出来；这时，用户可访问的最小单位为一个字符。视口是在图形屏显状态下的概念，文本与图形都可以在视口上显示，用户可访问的最小单位是一个象素(象素这一术语最初用来指显示器上最小的、单独的荧光体单元，它可以被扫描束单独激发。然而现在，其含义拓宽为指图形显示器上的最小可访问点)。

字符和图形状态下，屏幕上的位置都是由它们的行与列所决定的。有一点须指出：字符状态左上角坐标为(1,1)，但图形状态左上角坐标为(0,0)。

了解字符屏幕和图形函数与窗口和视口的关系是很重要的。例如，字符屏幕光标位置函数 gotoxy( ) 将光标移到窗口的 x,y 位置上，这未必是相对于整个屏幕。下面分别介绍常用的几类字符屏幕函数的功能用途、操作方法及其例行程序。

一. 屏幕操作函数

编写程序绘图经常会需要对字符屏幕进行操作。例如，在往屏幕上写字符之前，首先要将屏幕清除干净。又如，有时需要在屏幕上多处写上同样的字符内容，这时最好用屏幕拷贝来高效率地完成这一任务。对这些操作，Turbo C 为我们提供了一系列字符屏幕操作函数来实现。下面介绍之。

1. clrscr()清除字符窗口函数

函数 clrscr()清除整个当前字符窗口，并且把光标定位于左上角(1,1)处。

此函数调用方式为

```
void clrscr(void);
```

这里括号中 void 表示无参数。该函数相应的头部文件为 conio.h。使用这个函数的例子见下面 clreol() 函数的实例中。

Turbo C 的许多函数要与自己专门类型的数据和变量一起工作，用户程序也必须访问它们。这些变量和数据类型由编译程序提供的头部文件定义。在任何一个使用这些函数的文件中，涉及这些函数的头部文件必须包含进该文件(用#include 预处理指令)。此外，头部文件中有标准函数的原型定义，以提供更强的类型检

① 1 平方英寸=$6.451600\times10^{-4}m^2$

查手段。把用户程序中使用的标准函数与对应的头部文件连接起来。可以查出潜在的类型不匹配错误。如调试运行程序时，相应的头部文件在程序中没有包括进来，那么程序运行会出错(可能是浮点域错误或发生"死循环"等)，而 Turbo C 语言本身不会报告这类潜在错误是源于需要头部文件。因此可能反复检查程序语句是否有毛病，也许会猜想是语言版本不符，甚至怀疑计算机有病毒等。其实不然，一旦头部文件被包括进程序，这种潜在的错误就消失了，程序运行就正常了。自然，正确的结果可能就获得了。这时会对头部文件的体会、理解更深。Turbo C 的标准头部文件在表 1-1 里给出。本章每个函数注明其相应的头部文件。

表 1-1　标准头部文件

头部文件	用　途
alloc. h	动态地址分配函数
assert. h	定义 assert()宏
bios. h	ROM 基本输入输出函数
conio. h	屏幕操作函数
ctype. h	字符操作函数
dir. h	目录操作函数
dos. h	DOS 接口函数
errno. h	定义出错代码
fcntl. h	定义 open()使用的常数
float. h	定义从属于环境工具的浮点值
graphics. h	图形函数
jo. h	UNIX 型 I/O 函数
limits. h	定义从属于环境工具的各种限定
math. h	数学库使用的各种定义
mem. h	内存操作函数
process. h	spawn()和 exec()函数
setjmp. h	非局部跳转
share. h	文件共享
signal. h	定义信号值
stdarg. h	变量长度参数表
def. h	定义一些常用常数
stdio. h	以流为基础的 I/O 函数
stdlib. h	其它说明
string. h	字符串函数
time. h	系统时间函数
values. h	从属于机器的常数

2, window() 字符窗口函数

函数 window()用于在指定位置建立一个字符窗口、其调用方式为

```
void window(int left, int top, int right, int bottom);
```

函数中参数 left，top 为窗口左上角坐标；right，bottom 为其右下角坐标。若有一个坐标是无效的，则 window()函数不起作用。一旦该函数调用成功，那么所有定位坐标都是相对于窗口的，而不是相对于整个屏幕。但是建立窗口所用的坐标总是相对整个屏幕的绝对坐标，而不是相对当前窗口的相对坐标。这样用户就可以根据各种需要建立多个互不嵌套的窗口。

此函数相应的头部文件为 conio. h。调用这个函数的实例见下面 gotoxy()函数的例中。

3. gotoxy()光标定位函数

函数 gotoxy()将字符屏幕上的光标移到当前窗口指定的位置上。这是字符状态(有时称文本状态)下最常用的函数，其调用方式为

```
void gotoxy(int x, int y);
```

这里 x,y 是光标要定位的坐标。如果其中有一个坐标值无效(如坐标超出界),那么光标不会移动。此函数相应的头部文件为 conio. h。

[例] 下面程序建立两个窗口,然后在窗口里显示字符,字符的位置是调用该函数确定的。

```
# include "conio. h"
void border(int startx, int starty,
            int endx, int endy)
{
  register int i;
gotoxy(1,1);
for (i=o; i<=endx-startx; i++)
    putch('-');
gotoxy(1, endy-strty);
for (i=o; i<=endx-startx; i++)
    putch('-');
for (i=2; i,endy-starty; i++){
    gotoxy(1, i);
    putch('1');
    gotoxy(endx-startx+1, i);
    putch('1');}
}

main()
{
void border(int, int, int, int,);
clrscr();
window(6,8,38,12);
border(6,8,38,12);
gotoxy(2,2);
printf("window1");
window(8,16,40,24);
border(8,16,40,24);
gotoxy(3,2);
printf("window 2");
getche();
}
```

4. clreol()清除光标行尾字符函数

函数 clreol()在当前字符窗口中,清除从光标位置到行尾的所有字符,而光标位置保持不变、它的调用方式为

```
void clreol(void);
```

此函数相应的头部文件为 conio. h。

[例] 下面程序中使用了函数 clreol()和 clrscr()。

```
#   include "conio. h"
main()
{
register int i;
gotoxy(6,8);
printf("This is a test of the clreol()
```

```
        function.");
getch();
gotoxy(6,8);
clreol();
for (i=0; i<20; i++)
printf("Hello\n");
getch();
clrscr();  /* clear the screen */
}
```

5. insline()插入空行函数

函数insline()插入一空行到当前光标所在行上，同时光标以下的所有行都向下顺移一行。该函数只用于文本模式，并且在前当字符窗口才有效。此函数的调用方式为

```
void insline(void);
```

这个函数对应的头部文件是 conio.h 。

[例] 下面程序给出了insline()函数的用法。

```
#  include "conio.h"
main()
{
register int i;
clrscr();
for (i=1; i<24; i++)
{
gotoxy(1,i);
printf("This i line %d\n",i);
}
getch();
gotoxy(1,10);
inslin();
getch();
}
```

6. delline()删除一行函数

函数delline()删除当前窗口内光标所在行，同时把该行下面的所有行都上移一行。注意，如果当前窗口小于整个屏幕，那么该函数只影响到窗口内的字符。

此函数调用方式为

```
void delline(void);
```

这个函数相应的头部文件是 coino.h 。

[例] 下面程序先在屏幕上显示16行文字，然后删除第4行。

```
#  include"conio.h"
    main()
    {
    register int i ;
    clrscr();
    for (i=0; i<16; i++) printf("line %d\n",i);
    getch();
    gotoxy(1,4);
```

```
delline();
getch();
}
```

7. gettext()拷进文字函数

函数gettext()用于文本状态下将屏幕上矩形域内的文字拷进内存。其调用方式为

```
int gettext(int left, int top, int right,
int bottom, void * buffer);
```

函数中参数left，top为矩形域的左上角坐标;right,bottom为其右下角坐标。这些坐标是屏幕的绝对坐标,不是窗口的相对坐标。buffer指针必须指向一个足够保存该矩形域内文字的内存。所用内存大小按下式计算:

$$占用字节数=矩形域的行数\times矩形域的列数\times 2$$

这里将行数乘以列数再乘以2的原因是保存屏幕上每个字符要用两个字节存储单元,一个字节存储单元存放字符本身,而另一个存放其属性。若函数调用成功,则返回1,否则返回0。

此函数相应的头部文件是conio.h。

[例] 下面程序语句把屏幕上左上角点(1,1)和右下角点(10,10)的区域拷贝到buf指向的内存中去。

```
buf=(char * )malloc(10 * 10 * 2);
if(! buf) gettext(1,1,10,10,buf);
```

8. puttext()拷出文字函数

函数puttest()把先前由gettest()保存到buffer指向的内存中的文字拷出到屏幕上一个矩形区域中。

此函数调用方式为

```
int puttext(int left, int top, int right,
        int bottom, void *buffer);
```

这里(left,top)为给出的屏幕上矩形区域的左上角点;(right,bottom)为其右下角点。其坐标是用屏幕的绝对坐标,而不是用窗口的相对坐标。函数调用成功,返回值为1,否则为0。

该函数相应的头部文件为conio.h。

[例] 下面一段程序把屏幕上某个区域内容拷进buf指向的内存中,然后又将这些文字拷出到屏幕上新位置。

```
buf=(char *)malloc(10 * 10 * 2);
gettext(1,1,10,10,buf);
puttext(16,16,30,30,buf);
```

9. movetext()移动文字函数

函数movetext()将屏幕上一个矩形区域的文字移动到另一个区域上。该函数调用方式为

```
int movetext(int left, int top, int right,
    int bottom, int newleft, int newtop);
```

上面left,top为矩形区域左上角坐标;right,bottom为其右下角坐标;newleft,newtop为移动到区域左上角坐标。这些坐标是屏幕的绝对坐标,不是窗口的相对坐标。如果有一个以上坐标无效,那么函数返回0,否则返回1。

若要把屏幕上一段文字移到屏幕的另一位置,那么使用movetext()函数比用gettext()然后再用puttext()效率更高。

此函数相应的头部文件是conio.h。

[例] 下面程序语句把屏幕上左上角点(8,8),右下角点(20,20)的矩形区域文字移动到左上角点(10,10)的位置上。

```
movetext(8,8,20,20,10,10);
```