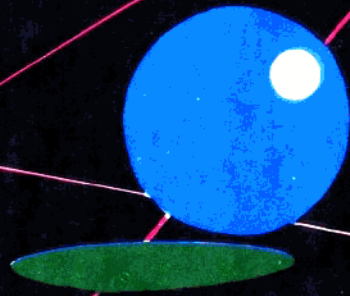
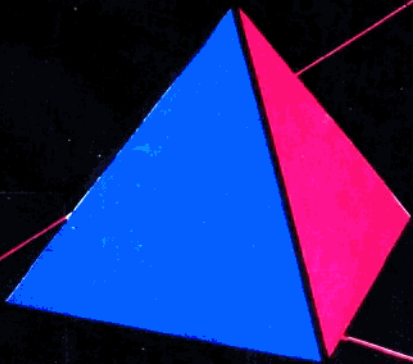


天津科学技术出版社

AutoCAD

...12

... 深入编程



C语言编程!

前 言

如果你已经掌握了 AutoCAD 的基础知识,正在寻求提高绘图效率的方法,从而使 CAD 进程更快、更容易,那么这本书正适合你。此节将告诉你如何根据你自己的绘图风格来定制独特的 AutoCAD 版本,从而满足你的需要。

一、本书适合的对象

即使你不是一位 AutoCAD 专家也可以使用这本书;仅仅需要你理解其基本的命令结构和具备一些使用本软件的经验。这本书提供了许多 AutoCAD 的命令,但不解释它们的基本用法。如果你完全是个新手,就应稍后再读本书。

将在第九章讨论的 AutoCAD 的高级开发系统(ADS),除了要求你懂得 AutoCAD 外还需要懂 C 程序设计,这并非要求你是位计算机专家或程序员,除了 ADS 这一部份外,仅仅是在教你 AutoCAD 定制技术时才会涉及到计算机程序设计的概念。

此书设想你是一名专业绘图员,旨在扩大你的 AutoCAD 应用面,而不是使你成为一名专业计算机程序员。因此,你不能将此书看作是一本程序设计指南。如果此书使你学到了更多的有关计算机程序设计方面的知识,当然更好。深入研究程序设计的好书可以在图书馆或计算机书店找到。

定制是值得做的。如果仅使用 AutoCAD 的现有版本,可以说是一种浪费,因为 AutoCAD 是特意设计成为可定制的。现有的版本仅是一个基础,你能据此定制版本,绘出美妙图形。

学会定制一个 AutoCAD 软件并不比学习 AutoCAD 的基本知识更难。你需要学习一些新的术语,研究一些实例,但最重要的在于实践与经验。

定制确实是在时间与耐心上的一项投资。这项投资是非常值得的,因为定制能给你带来巨大的收益。这将使你及其他用户更容易地使用 AutoCAD,如果坚持这样做,你甚至可以开发第三方 AutoCAD 应用程序来出售给其他用户。

二、如何使用本书

使用本书的最好方法是一边阅读,一边上机操作 AutoCAD。这样你可以自己反复研究例题,并尽快、尽可能完整地吸收那些基本概念。

这本书应从前至后地阅读。所有的章节都是循序渐进的,由简单到复杂。每个人都必须先读第一章“概论”,在此之后,你可以首先浏览你特别感兴趣的部份。但是,如果你突然发现自己不太看得明白了,那就该回头去看前面的章节了。

虽然此书中的例子是独立的,你可以将它们组合使用。但这还不是它们的全部作用。它们的主要目的是启迪你的想象力,让你考虑你自己的需要,并一步步创造原始特性、命令以及程序。仅仅推敲例子都会很有益,但是如果你肯花时间去读懂那些论述和解释,那么收益更大。

三、本书的内容

本书可视为一座桥梁,用以连接 AutoCAD 文档中的定制概念和用户——既想了解定制概念又受限于 AutoCAD 文档。

你不要将此书作 AutoCAD 文档的完全替代品。看完此书后,你会希望重阅文档,你将发现它更易于理解。本书的内容包括以下方面:

- 第一章给出了 AutoCAD 定制特性的概论。
- 第二、三章告诉你如何扩展与修改一些标准的 AutoCAD 特性:线型、影线图案以及形。
- 第四、五章提供了以下技术:AutoCAD 屏幕菜单与图形输入板选择单的重构,可以让你做全新的菜单;通过定义宏使常用的 AutoCAD 命令自动执行。
- 第六章介绍了开发 AutoLISP 和 ADS 源程序的基本技术。如果你是程序设计方面的新手,尤有必要读这一章。
- 第七章向你介绍了 AutoCAD 的内嵌语言 AutoLISP。这章介绍了如何定义新的命令、特性、函数,来提高 AutoCAD 的性能,更好为你服务。
- 第八章通过示例提高你对 AutoLISP 的理解力,教你如何连接 AutoCAD 的基本数据库;将你的特定信息与 AutoCAD 绘图实例联系起来。
- 第九章介绍了高级开发系统(ADS)。本章详细介绍了使用 C 程序设计语言编程和编译、执行定制的 AutoCAD 应用程序。

这本书有两个附录。附录 A 总结并比较了 AutoCAD 中的 AutoLISP 和 ADS 的内部函数;附录 B 为你提供了用以提高工作效率及作进一步研究的例子函数。

四、你应使用的 AutoCAD 版本

如果你还没确定的话,我推荐你使用 AutoCAD 的最新版(本书写作时是 12 版)。每个 AutoCAD 新版本都有显著的改进和新特征,并增强了现有的命令。12 版有许多新的和改进的特色,包括:整体速度的提高;绘图功能的提高;易于选择实体;可编程的对话框;易于与外部数据库相连;ADS 和 AutoLISP 的显著增强。

在这本书中描述的特性:影线图案、形文件、线型文件,大部份菜单宏以及及众多的 AutoLISP 内部函数都向上兼容,直至 2.5 版。这本书中罗列的每个 AutoLISP 函数都包括一个如何兼容 AutoCAD 早期版本的说明。一些程序例显示了 12 版的独有特征,它只能在 12 版中运行。

这本书的所有 AutoLISP 代码都是向上兼容的,这意味着在早期版本中编写的程序可以在后期版本中运行,直到 12 版。

目 录

第一章 概述	(1)
§ 1.1 “定制”包含的内容	(1)
§ 1.2 为何引入“定制”?	(1)
§ 1.3 ASCII 文件介绍	(2)
§ 1.4 AutoCAD 中可修改的源程序文件	(2)
§ 1.4.1 *.DCL	(2)
§ 1.4.2 *.EXE 和 *.EXP	(3)
§ 1.4.3 *.LIN	(3)
§ 1.4.4 *.LSP	(3)
§ 1.4.5 *.MNL	(3)
§ 1.4.6 *.MNU	(3)
§ 1.4.7 *.SHP	(4)
§ 1.4.8 ACAD.PAT	(4)
§ 1.4.9 ACAD.PGP	(4)
§ 1.5 设定 DOS 环境	(4)
§ 1.5.1 进入多个绘图目录	(5)
§ 1.6 AutoCAD 的参数	(5)
§ 1.6.1 使用 AUTOEXEC.BAT	(6)
§ 1.7 建立多 AutoCAD 配置(11 版及更早版本)	(6)
§ 1.8 建立批处理文件	(7)
§ 1.9 在 AutoCAD 中使用文本编辑器	(7)
§ 1.9.1 修改 ACAD.PGP	(8)
§ 1.10 提示项中空格的功能	(9)
§ 1.11 不得已情况下使用的批处理文件	(9)
§ 1.12 关于试验	(10)
第二章 使用定制线型和影线图案	(11)
§ 2.1 定制线型	(11)
§ 2.1.1 使用图形编辑器	(11)
§ 2.1.2 使用字处理器	(13)
§ 2.2 定制影线图案	(14)
§ 2.2.1 影线图案定义的结构	(15)
§ 2.2.2 建立新的影线图案	(19)

第三章 使用形和字型	(22)
§ 3.1 形文件.....	(22)
§ 3.1.1 形文件的编辑	(22)
§ 3.1.2 用于描述形的元素	(23)
§ 3.1.3 建立一个形描述	(25)
§ 3.1.4 复杂的形描述	(27)
§ 3.1.5 使用非标准线和弧的形	(35)
§ 3.2 字型文件.....	(41)
§ 3.2.1 比较字型文件和形文件	(42)
§ 3.2.2 字型文件中的特殊形描述	(43)
§ 3.2.3 设计字符的形	(43)
§ 3.2.4 双向字型文件	(44)
§ 3.3 在 12 版中读一个 Post Script 字型	(45)
§ 3.4 将字型实体转化为线段实体.....	(46)
§ 3.5 编辑已编译过的形和字型文件.....	(47)
第四章 建立定制菜单	(48)
§ 4.1 建立定制菜单.....	(48)
§ 4.1.1 了解你的需要	(48)
§ 4.1.2 为 AutoCAD 标准菜单制作副本	(49)
§ 4.1.3 装入定制菜单 II	(49)
§ 4.2 屏幕菜单结构.....	(50)
§ 4.2.1 主要部份	(50)
§ 4.2.2 菜单子部份	(50)
§ 4.2.3 句法	(51)
§ 4.3 阅读 AutoCAD 的标准菜单	(54)
§ 4.4 CUSTOM 菜单的第一次改变	(58)
§ 4.5 利用数字化仪的菜单.....	(60)
§ 4.5.1 设计模板	(62)
§ 4.5.2 构造模板	(62)
§ 4.5.3 将命令置于数字化平板上	(62)
§ 4.6 建立一新的菜单子部分.....	(64)
§ 4.7 建立新的下拉菜单.....	(65)
§ 4.7.1 下拉式菜单的结构	(65)
§ 4.7.2 控制下拉菜单的显示	(66)
§ 4.7.3 下拉菜单的其它特性	(70)
§ 4.8 建立图符菜单.....	(70)
§ 4.8.1 建立 AutoCAD 幻灯文件	(71)
§ 4.8.2 建立幻灯库文件	(71)

§ 4.8.3	在 ACAD.MNU 中建立图符菜单	(72)
§ 4.8.4	在屏幕上显示图符菜单	(73)
第五章	定制菜单宏	(75)
§ 5.1	实用的宏	(75)
§ 5.1.1	层宏	(75)
§ 5.1.2	保存键盘输入的菜单子部分	(77)
§ 5.1.3	快速 Undo	(77)
§ 5.1.4	快速缩放	(78)
§ 5.1.5	快速清屏	(78)
§ 5.1.6	旋转十字准线	(78)
§ 5.1.7	字处理宏	(79)
§ 5.2	辅助编辑宏	(79)
§ 5.2.1	快选宏	(79)
§ 5.2.2	结合绘图和编辑命令的宏	(81)
§ 5.2.3	结合编辑与显示命令的宏	(82)
§ 5.3	块操作宏	(87)
§ 5.3.1	插入块	(87)
§ 5.3.2	插入带有属性的块	(89)
§ 5.3.3	编辑插入块周围的线段	(91)
§ 5.3.4	其综的编辑和显示宏	(92)
第六章	AutoLISP 程序设计语言	(94)
§ 6.1	设计一个 AutoLISP 程序	(94)
§ 6.2	句法规则	(95)
§ 6.2.1	函数	(95)
§ 6.2.2	内存变量	(96)
§ 6.2.3	嵌套	(98)
§ 6.2.4	系统变量	(98)
§ 6.2.5	弧度	(99)
§ 6.3	基本的 AutoLISP 数据类型	(99)
§ 6.3.1	实体名字	(99)
§ 6.3.2	文件描述符	(99)
§ 6.3.3	整型和实型数	(99)
§ 6.3.4	表	(100)
§ 6.3.5	字符串	(100)
§ 6.3.6	符号	(100)
§ 6.3.7	选择集	(101)
§ 6.4	在 AutoCAD 中管理 AutoLISP 程序	(101)
§ 6.4.1	装入 AutoLISP 程序	(101)

§ 6.4.2	内存管理	(101)
§ 6.5	提高 LISP 文件的可读性	(104)
§ 6.5.1	注释	(104)
§ 6.5.1	缩格	(104)
§ 6.6	摆脱困境	(105)
§ 6.6.1	用 Ctrl-C 实现控制台中断	(105)
§ 6.6.2	函数不平衡时的提示	(105)
第七章	建立 AutoLISP 程序	(106)
§ 7.1	Defun	(106)
§ 7.1.1	全局和局部变量	(107)
§ 7.2	使用参数的程序设计	(108)
§ 7.3	一个基本的 LISP 程序	(109)
§ 7.4	描述问题	(109)
§ 7.5	编写伪代码	(110)
§ 7.5.1	GEAR.LSP 的伪代码	(110)
§ 7.6	GEAR.LSP 程序	(111)
§ 7.6.1	记住缺省值	(115)
§ 7.7	全局函数	(116)
§ 7.7.1	G:DTR	(116)
§ 7.7.2	G:STR	(116)
§ 7.7.3	G:INT	(117)
§ 7.7.4	G:DIST	(118)
§ 7.7.5	G:PT	(118)
§ 7.7.6	GETP	(119)
§ 7.7.7	CALCP	(120)
§ 7.7.8	DRAW	(120)
§ 7.7.9	NOTCH	(121)
§ 7.7.10	C:GEAR	(121)
§ 7.8	增强 GEAR.LSP	(121)
§ 7.9	GEAR.LSP 中的错误陷阱	(122)
§ 7.10	使用对话框输入参数(仅在 12 版中)	(123)
§ 7.10.1	设计一个对话框	(124)
§ 7.10.2	通过一个对话框控制数据	(128)
§ 7.11	在一个定制菜单中改进 LISP 程序	(137)
§ 7.12	共享函数——ACAD.LSP	(137)
第八章	实体联结表	(138)
§ 8.1	实体名和联结表	(138)
§ 8.2	实体标识符	(139)

§ 8.3	实体存取函数	(139)
§ 8.3.1	用 Entmake 建立实体	(140)
§ 8.3.2	从数据库中访问实体	(144)
§ 8.4	WALLS.LSP	(147)
§ 8.4.1	画面	(158)
§ 8.4.2	管理变量名	(158)
§ 8.4.3	全局函数	(159)
§ 8.5	WALLS.LSP 中的实体访问函数	(160)
§ 8.5.1	MakeLine	(160)
§ 8.5.2	MakeFace	(160)
§ 8.5.3	Modify	(161)
§ 8.5.4	Accum	(161)
§ 8.5.5	Addset	(162)
§ 8.5.6	Rdwsef	(162)
§ 8.6	Walls.LSP 中的其他函数	(163)
§ 8.6.1	WALLS	(163)
§ 8.6.2	Wdraw	(163)
§ 8.6.3	Wedit	(163)
§ 8.6.4	Wallpt	(163)
§ 8.6.5	Getli	(163)
§ 8.6.6	Getwindow	(164)
§ 8.6.7	D_Puall 和 D_Wwall	(164)
§ 8.6.8	Getoffset	(164)
§ 8.6.9	Closeup	(164)
§ 8.7	用 Command 函数直接访问实体	(164)
§ 8.8	使用 WALLS.LSP	(164)

第九章 AutoCAD 高级开发系统 ADS

§ 9.1	本章写作对象	(166)
§ 9.2	ADS 特征	(166)
§ 9.3	为效率、效益和兴趣进行程序设计	(167)
§ 9.4	ADS 开发的要求	(167)
§ 9.5	关于程序设计语言	(168)
§ 9.6	ADS 的支持编译器	(168)
§ 9.7	编译 AutoCAD 的样板 ADS 程序	(168)
§ 9.8	ADS 开发过程——基本步骤	(169)
§ 9.9	开始建立一个 ADS 程序——FPROMPT.C	(170)
§ 9.10	在 ADS 中执行函数	(172)
§ 9.11	TEMPATE.C	(173)
§ 9.12	测试函数	(176)

§ 9.13	CUTOUT.C	(176)
§ 9.14	CUTOUT.C 的伪代码	(177)
§ 9.15	函数表	(189)
§ 9.16	宏和常量	(189)
§ 9.17	CUTOT.C 的函数	(190)
§ 9.18	CUTOUT.C 中的外部变量	(190)
§ 9.19	使用 ads_ssfree() 和 ads_relrb()	(191)
§ 9.20	控制程序行为	(191)
§ 9.21	点处理	(191)
§ 9.22	建立和恢复 AutoCAD 环境	(192)
§ 9.23	使用 UNDO	(193)
§ 9.24	在 ADS 程序中创建实体	(193)
§ 9.25	对实体进行操作	(194)
§ 9.26	处理选择集	(195)
§ 9.27	搜索块表	(197)
§ 9.28	更进一步	(197)
附录 A AutoLISP/ADS 的函数参考		(199)
附录 A.1	用作函数的符号	(199)
附录 A.2	按字母顺序排列的函数表	(202)
附录 B AutoLISP 程序和利于学习的函数		(267)
附录 B.1	双字符宏	(267)
附录 B.2	使用文件	(271)
附录 B.3	输出 AutoCAD 文本到一个 ASCII 文件中	(275)
附录 B.4	浏览联结表	(277)
附录 B.5	初步调试	(278)
附录 B.6	切换系统变量	(279)
附录 B.7	另一种倒圆方法	(280)
附录 B.8	拷贝和旋转	(282)
附录 B.9	绕任意轴旋转三维实体	(283)
附录 B.10	计算形中的凸弧	(286)
附录 B.11	用于块的程序	(287)

第一章 概 述

AutoCAD 之所以能作为 PC 机的 CAD 标准,主要原因在于它的适应性。用户可以对 AutoCAD 进行修改,使它符合制图员与设计者的实际需要与规定。学习修改这个软件是用户能够胜任的。大多数用户是在严格限制下绘图的专业人员,要求在很短的时间内掌握定制与重构计算机软件的技术。与 AutoCAD 定制有关的基本知识简单易学,学习这些知识的时间投资会带来很大收益。

这本书说明了本软件的哪些部分可以定制。它演示了设计及开发 AutoCAD 版本的必要技术,这个特定的版本可以极大地提高绘图效率。

为了从本书中获益,你应具备 AutoCAD 的基础知识。如果你能够在 AutoCAD 中绘制并修改绘图实体,那么你将能够理解我们提供的资料。为了学习和利用高级开发系统(ADS)来开发定制应用软件,你还应该具备 C 语言方面的工作经验。

§ 1.1 “定制” 包含的内容

定制 AutoCAD 有各种不同的形式,从绘制简单的定制图形实体开始,到建立定制屏幕菜单和命令宏,直至开发基于 AutoCAD 内嵌语言 AutoLISP 与 ADS 的特殊的 AutoCAD 命令。

第二章介绍了如何建立自己的定制线型和影线图案库。第三章介绍了怎样建立和编译包括你的图形实体和符号的形文件。这些资源可以使你避免重复建立同样的图案和符号,并避免保存大量的外部块写文件。

第四和第五章演示了 AutoCAD 的屏幕菜单功能以及如何改变它以适应你的绘图需要。你还将学习如何将 AutoCAD 命令组织成宏,这样通过一个屏幕菜单选择就可以完成一系列任务。

第六、七和八章介绍如何使用 AutoLISP。利用 AutoCAD 的内部程序设计语言,你可以建立 AutoCAD 命令以完成计算和分析或产生图形实体;你也可以利用它来开发具有特殊功能的命令,以简化和加速绘图进程。

第九章指导你利用 ADS 进行一个 AutoCAD 定制软件的编程、编译和执行。

§ 1.2 为何引入“定制”?

AutoCAD 用户要花费大量时间从菜单进行选择或以正确的顺序从键盘上输入命令,再选择该命令的正确参数。对于有经验的 AutoCAD 用户来说,定制可以允许你将一长串的键盘输入组合起来,从而提高绘图进程的效率。此外,如果一个图形的建立过程中包括计算,定制能将这些计算正确地包括在绘图命令中。

毫无疑问,AutoCAD 绘图过程中的某些部分是用来调整软件的——改变层和并开关它们,调整点阵网格、设置绘图参数等等。AutoCAD 的定制版本将大量减少用于调整软件的时间,使用户集中精力于绘图与设计。

一个定制的 AutoCAD 一旦开发安装成功,不熟练的用户通过经常使用也同样受益。友好的用户界面减少了学习时间,并提供了最多的机会使他们画出有用的图形。

为什么 AutoCAD 没有首先提供这些东西呢?实际上,它是提供了的。AutoCAD 不仅仅是一个绘图软件,它还是一个大而全面的工具箱,包含了许多绘图工具。这些工具是程序设计的基础,可供各种不同类型的绘图员使用。毫无疑问,有些工具对你来说可能比对别人更有用,因此定制 AutoCAD 的过程是用一种方法将这些工具组合起来,使之对你更有效。将常用的工具组合起来,用现有的工具创造你自己的绘图工具。

§ 1.3 ASCII 文件介绍

大多数字处理器和一些文本编辑器在你的编辑文件中加入了特殊的代码,这些嵌入的代码处理打印和格式输出任务,诸如下划线、黑体、隐含和段落重排。一个 ASCII 文件是一个不含任何特殊控制符的文件,ASCII 文件只包括字母表中的字母,数字和标准的标点符号。

定制 AutoCAD 包括建立和编辑 ASCII 文件,AutoCAD 将在执行过程中随时调用它。DOS 的文本编辑器 EDIT 或 EDLIN,能用于建立和编辑 ASCII 文件,但最有效的方法是使用字处理器或文本编辑器。

计算机指令被称为源代码。这样,包括计算机指令的 ASCII 文件被称为源代码文件。AutoCAD 处理这种代码有两种方法:或者直接读取源程序代码,按文件的顺序解释每一行;或者把源代码编译成一个机器可读的形式。每种处理方法都各有其优点,但是两种方法结果一样。当你给 AutoCAD 一个命令,使它读一个源代码文件,那么它会做以下两件事之一:

- 在磁盘上定位到正确的文件,将它的整个内容复制到一个内存区以实现目标。把文件内容放入内存的过程称为装载文件。一旦文件被装载,AutoCAD 就能高效地执行这些内容。例如 AutoLISP 文件和 AutoCAD 形文件都以这种方式装载。
- 在磁盘上正确定位文件,仅把指令所指到的文件部分拷入内存。当 ASCII 文件很长时,比如说文件包括标准的 AutoCAD 影线图案,或一个文件包括线型定义,那么这种方法更有效。

§ 1.4 AutoCAD 中可修改的源程序文件

AutoCAD 通过文件的扩展名来识别一个源代码文件,扩展名是放在文件名之后的一组三个字母,由一个点将它与前面的文件名隔开。AutoCAD 可识别的文件扩展名有 DCL、EXE、EXP、LIN、LSP、MNL、MNU、PAT、PGP 和 SHP。以下是对每种类型文件的简介,每种类型的文件可以有不同的文件名,我们用星号来代表文件名。

§ 1.4.1 *.DCL

具有这种扩展名的 ASCII 文件,包括了 12 版中设计定制对话框的指令。你可以使用 Autodesk 的对话框控制语言(DCL)来建立一个新的对话框,DCL 是一种专门管理对话框的特殊语言。第七章中有 DCL 的详细介绍。

§ 1.4.2 *.EXE 和 .EXP

编译过的 ADS 具有这种扩展名,它们是根据 ADS 参数来编译的,这些参数是编译软件所设定的。AutoCAD 本身并不包括编译软件,必须单独购买编译软件。比如 MetaWare 的 High C 编译器,在这本书中用来编译 ADS 程序。许多文件都有这种扩展名,只有那些同 AutoCAD 的 ADS 一起使用且经过特殊编译的文件,才可能被装入图形编辑器。这些 ADS 源代码文件,通常的扩展名为 C,如 MYAPP.C 将被编译成 MYADD.EXP (ADS 的扩展名模式),或 MYAPP.EXE (ADS 的实际模式)。第九章和你的编译手册将更详细地介绍这些文件的建立。

§ 1.4.3 *.LIN

一个具有这种扩展名的 ASCII 文件,包含了 AutoCAD 画不同线型时所需要的信息。A-CAD.LIN 包括了 AutoCAD 的标准线型。你可以在这个文件中增加你自己需要的线型,或在不同的文件中建立线型库,但必须保证文件名各异,文件扩展名为 LIN。

在 LIN 文件中各种线型应给以不同的名字来区别它们,当你希望在绘图过程中使用一种线型时,必须从 LIN 文件中找到它的信息并将它存入图形数据库中去。在 AutoCAD 中用以装载一个新线型的命令是 LINETYPE。当你发出此命令时,AutoCAD 首先提示你输入线型的名字,然后再输入包含线型信息的文件名。当 AutoCAD 找到有关信息,它会显示一条结果信息。

通过 AutoCAD 的图形编辑器,可以将新的线型加入到 LIN 文件中去,这样,你能够在绘图过程中自由地创造新的线型。

§ 1.4.4 *.LSP

具有此扩展名的 ASCII 文件包含用 AutoLISP (AutoCAD 的内部程序设计语言)写成的源代码。定制 AutoCAD 文件的安装通常要用到许多不同的 AutoLISP 文件。包含 AutoLISP 源代码的文件可以用任意有效的名字,但通常它们的缺省扩展名是 LSP,这种缺省扩展名使得 AutoCAD 很容易识别并解释它们。AutoCAD 并不编译这些文件,而是将它们直接装入内存,顺序解释它们。

§ 1.4.5 *.MNL

具有此扩展名的 ASCII 文件包含与 12 版菜单文件有关的 AutoLISP 代码。在 12 版中当你装载一个菜单文件时,AutoCAD 就在磁盘上寻找一个名字同菜单文件名相同,扩展名为 MNL 的文件。如果 AutoCAD 找到一个这样的文件,那么在装入新菜单之后就装入此文件。

§ 1.4.6 *.MNU

这个文件扩展名是一个 AutoCAD 菜单文件的标志,它包括屏幕菜单提示和命令——当你用指示器选中一个提示项时就会执行。你可以做任意数量的 AutoCAD 菜单,每个菜单的菜单名字各异,而扩展名为 MNU。

早期的 AutoCAD 版本与后来的版本对这种文件的处理是不同的。2.18 版本和更早的版本直接读取这个文件并将其显示在屏幕上;2.5 版和以后的版本则编译菜单文件后再显示。AutoCAD 第一次在图形编辑器中装入 MNU 文件时,菜单文件中的编译过程将自动进行。通过编译菜单文件(通常很大),AutoCAD 在绘图期间可节约大量的装载时间和菜单显示时间。

一旦编译完成,菜单文件的机器可读代码被包含在另一个单独文件中,此文件具有同样的名字和一个新的扩展名—*.MNX。其后,只有编译过的菜单代码才能被装载和显示。源代码文件仍然保留,以防你还需修改。

编译过的 AutoCAD 的标准菜单译本包含在文件 ACAD.MNX 中。标准菜单的源代码包含在文件 ACAD.MNU 中,通常 ACAD.MNU 安装在你的硬盘的子目录\ACAD\SUPPORT 中。如果你修改了这个文件,必须让 AutoCAD 在这个子目录下找到这个文件,并将它重新编译成 MNX 文件。

用不同名字拷贝一份 ACAD.MUN 是明智之举,然后用 AutoCAD 的 MENU 命令修改和装载这个副本,这样 AutoCAD 就避免了因为失误造成的改动覆盖原来的标准菜单文件,详见第四章。

§ 1.4.7 *.SHP

具有这种扩展名的 ASCII 文件包含绘图符号、形或实体(类似于块,可以插入到图形中)的定义。AutoCAD 将这种文件编译成别的文件,扩展名为 SHX,这种文件称为形文件。你可以用 AutoCAD 的“LOAD”命令将形文件装入图形编辑器,装入形文件后你可以使用 AutoCAD 的 SHAPE 命令将形插入到图形中去。形的插入比块的插入快,但它通常用于相对简单的图形符号,对于复杂符号或一次性插入,多采用块插入的方法。

字型文件是一种特殊的形文件,它包含了对应字型的形,还有其它特殊形定义(表示了字符的比例和方向)。例如,AutoCAD 的标准字型的源代码存贮于下面一些文件中:TEXT.SHP、SIMPLEX.SHP、COMPLEX.SHP、ITALIC.SHP、MONOTXT.SHP,和 VERTICAL.SHP。这些文件通常安装在硬盘的\ACAD\SOURCE 子目录下。

§ 1.4.8 ACAD.PAT

这个文件包含了 AutoCAD 的影线图案。同前面的扩展名不同,PAT 扩展名不能被其它文件名使用,AutoCAD 只认识 ACAD.PAT。使用文本编辑器,你可以将自己的影线图案定义加入到这个文件的标准图案表之后。

§ 1.4.9 ACAD.PGP

ACAD.PGP 文件包含了特殊的 AutoCAD 命令,可以在图形编辑器中使用。这是能够使用扩展名 PGP 的唯一文件,此文件通常安装在\ACAD\SUPPORT 子目录下。你可以通过修改这个文件来定制 AutoCAD。经过修改,你可以在 AutoCAD 及文本编辑状态之间很快地来回切换。

§ 1.5 设定 DOS 环境

在运行 AutoCAD 之前执行特定 DOS 命令,你就可以不必对相应的 ASCII 码文件作很多改变就可以有效地设定环境。这些命令并不直接改变 AutoCAD 的执行,它们的重要作用在于改变 AutoCAD 运行的 DOS 环境,这样你的 CAD 工作会更容易和迅速。

§ 1.5.1 进入多个绘图目录

对于许多用户来说,将不同的图形文件保存在不同硬盘子目录中非常必要。把相关的图形文件组织到一个子目录下,用户可以简化寻找它们的过程或备份的过程。一旦你已经用这种方式组织了你的图形文件,你就可以利用以下的技巧来实现这个目的。

把 AutoCAD 系统子目录加入到 DOS 的搜索路径中是很好的方法。在许多系统中,这个子目录是 \ACAD。例如,在 DOS 提示符下发出以下命令,在现存的搜索路径 C:\DOS 后加入子目录 C:\ACAD:

```
PATH=C:\DOS;C:\ACAD
```

这个命令改变了操作系统的配置,为了保留这个改变,可以使用文本编辑器对系统根目录下的 AutoEXEC.BAT 文件中 PATH 语句进行增加或修改。一旦 AutoCAD 系统子目录被加到 DOS 搜索路径中之后,AutoCAD 就可以在当前工作目录下被调用。例如,假定你的图形文件在两个不同的子目录 DRAWING1 和 DRAWING2 中。用 AutoCAD 编辑 DRAWING1 中的图形,启动 AutoCAD 之前,在 DOS 提示符下键入:

```
CD\DRAWING1
```

然后,从这个子目录中直接调用 AutoCAD,AutoCAD 会正常执行,因为你已经发出了 DOS PATH 命令,并告诉了 DOS 在什么地方寻找不在当前工作目录下的命令文件。现在启动 AutoCAD,图形文件将在子目录 DRAWING1 下存取。你还可以通过键入所需图形的全部路径名,来存取其它驱动器和子目录下的图形文件。

§ 1.6 AutoCAD 的参数

如果你愿意,可以为 AutoCAD 设定一个特定的搜索路径,当它在从图形编辑器中搜索文件时使用。你可以在进入 AutoCAD 之前,在 DOS 状态下设置 ACAD 参数,包括一个子目录列表。

当你安装 AutoCAD 11 版及以后的版本,需要至少用包含 AutoCAD 文件的两个子目录名来设置这个参数。

安装好 AutoCAD 之后,你可以在 ACADRI2.BAT 文件中找到下面的命令。用类似于下面的 DOS 命令来设置你的系统:

```
SET ACAD=C:\ACAD;C:\ACAD\SUPPORT
```

它让 AutoCAD 除了在当前工作目录下搜寻文件以外,还在子目录 \ACAD 和 \ACAD\SUPPORT 下寻找文件,注意上面列表中的两个子目录要用分号隔开,而且要包括一个驱动器名。你可以在此表中加上其它目录,例如:

```
SET ACAD=C:\DRAWING;C:\DRAWING2\ACAD;ACAD\SUPPORT
```

一旦你设置的 ACAD 参数包括这些子目录,当你输入图形文件名时,AutoCAD 将到这此子目录中寻找已有的图形文件。

须当心在搜索路径中加入太多目录名,也会出现毛病。太长的搜索路径会减慢搜索文件的速度。如果同名的文件放在两个不同的目录中,AutoCAD 只能装入在路径中找到的第一个文件。

而且,长的搜索路径占据太多的操作系统空间。你可以通过修改 CONFIG.SYS 文件增大

DOS 环境空间,当重引导计算机或调用“SET ACAD”命令时,如果你看到“Out of environment space”的消息,那么请查阅 DOS 文档。

§ 1.6.1 使用 AUTOEXEC. BAT

当你为 AutoCAD 正确设置了 DOS 环境,你会发现一些 DOS 命令(例如,PATH)保持不变,而与其它设置无关。如果愿意的话,你可在 AUTOEXEC. BAT 文件中加入这样的命令以代替你的系统批处理文件。DOS 在引导时搜寻 AUTOEXEC. BAT,如果找到的话,AUTOEXEC. BAT 将自动执行,这将节约时间与磁盘空间。

§ 1.7 建立多 AutoCAD 配置(11 版及更早版本)

某些 AutoCAD 用户拥有不止一套外设与 AutoCAD 配用,比如,你可能拥有不止一台绘图机。

12 版允许你给 AutoCAD 配置多个绘图设备。在绘图时,选择其中一个配置好的设备。在以前的版本中要求你使用一个特殊的 DOS 参数——ACADCFG,它指出包含配置文件 ACAD. CFG 的子目录。当 AutoCAD 启动时,它在这个文件中寻找信息配置硬件。通过把不同的配置文件放在不同的子目录下,你可以在运行 AutoCAD 之前通过改变参数 ACADCFG 的值来使用不同的配置。

下面的例子假定 AutoCAD 的系统文件放在子目录 ACAD 中。本例使用不同的显示配置来构造 AutoCAD:一个包括屏幕菜单,而另一个没有。用两个特殊的子目录来分别保存配置文件,名为 MENU 与 NOMENU。

先在硬盘上建立两个子目录,这两个子目录仅用作存放两个版本的 ACAD. CFG,而不能包含其它文件。然后在 DOS 提示状态下键入:

```
MD C:\MENU
```

```
MD C:\NOMENU
```

建立了这两个目录后,下一步就是用 DOS 的 SET 命令,它让 AutoCAD 把 ACAD. CFG 文件放在适当的目录下:

```
SET ACADCFG=C:\MENU
```

然后,调用 AutoCAD 并根据硬件重新配置它,表示你的屏幕显示包含一个屏幕菜单。如果你已正确配置了 AutoCAD,则退出系统回到 DOS 状态,键入:

```
SET ACADCFG=C:\NOMENU
```

象刚才那样启动 AutoCAD,你要再次配置 AutoCAD,重新回答配置提示,这次表示屏显不包括一个屏幕菜单。当 AutoCAD 正确配置完毕,退出,你的工作就完成了。

从现在起,如果你希望重新使用有菜单的屏幕显示,起动 AutoCAD 之前先在 DOS 提示符下键入下面的命令:

```
SET ACADCFG=C:\MENU
```

如果你希望使用无屏幕菜单的屏幕显示,在运行 AutoCAD 之前键入:

```
SET ACADCFG=C:\NOMENU
```

你可以使用任意目录名,可以根据需要有许多不同的配置,但是每个 ACAD. CFG 文件必须在它自己的子目录中。

§ 1.8 建立批处理文件

在进入 AutoCAD 之前要用的 DOS 命令并不需要每次都从键盘敲一遍。你可以建立另一种类型的 ASCII 文件,名叫批处理文件,它包括顺序的 DOS 命令,一个命令占一行。一旦你建立了一个批处理文件,你可以仅键入此文件的文件名,文件中的 DOS 命令就会自动执行。记住哪套 DOS 命令适应于哪个版本的 AutoCAD,使批处理能够方便地进行选择。

如果你对建立批处理文件的过程不熟悉,参考 DOS 文档。你可以根据需要建立众多的批处理文件,以解决图形文件子目录和硬件配置的各种组合情况。保证所建立的所有文件都有备份,以便在由于疏忽丢失了原始文件的时候能节约恢复时间。

§ 1.9 在 AutoCAD 中使用文本编辑器

定制 AutoCAD 在你的文本编辑器与 AutoCAD 之间来回切换。首先你要修改 ASCII 文件,然后到 AutoCAD 中去测试其结果。由于你可能会相当频繁地来回切换,那么不用退出 AutoCAD 的图形编辑器就能使用文本编辑器,这是最好的方法。这样做,你可以节省大量时间,因为你可以不必在每次修改 ASCII 文件之后等待 AutoCAD 重新启动。

对 ACAD.PGP 文件进行修改,就可以在图形编辑器中调用文本编辑器,这个文件通常放在 \ACAD\SUPPORT 子目录下。

在你着手修改这个文件之前,你必须弄清 AutoCAD 与你选择的文本编辑器怎样一起工作。开始时,你需要决定在 AutoCAD 系统目录下该用什么方式来激活文本编辑器。

如果你的文本编辑器很小,能与 AutoCAD 驻留在同一子目录下,那么简单地拷一个副本到子目录下就行。这种方法是可行的,这样做,可以保证文本编辑器的命令正确执行。但是如果你的文本编辑器很大,或出于某种原因你不想将它同 AutoCAD 放在同一个子目录里,你可以把它放在别的子目录里。有些文本编辑器允许你在其它子目录中启动,而不一定要在它自己的子目录之下,但是有些文本编辑器这样做会出毛病。

如果你选择将文本编辑器同 AutoCAD 放在不同子目录下,有必要测试一下你是否能从 AutoCAD 中调用文本编辑器。下面的例子将说明如何去做。

这个例子中,假定文本编辑器在子目录 TEXT 下面,AutoCAD 安装在子目录 ACAD 中。将文本编辑器的子目录加到 DOS 搜索路径中,使用下面的命令:

```
PATH=C:\DOS;C:\ACAD;C:\TEXT
```

这个 DOS 命令能保证在磁盘的任何地方都可以调用文本编辑器和 AutoCAD 软件。现在进入到 AutoCAD 的子目录中,试着调用你的文本编辑器。检查下面几条:

- 你的文本编辑器是否出现在屏幕上,也许比平时晚几秒钟?
- 在文本编辑器中,你能否调用缺省子目录中的一系列文件,它们是 ACAD 文件吗?如果不是,你能改变目录到 AutoCAD 的子目录吗?
- 当你在文本编辑器中打开一个文件进行测试,键入一些字符并存起来,退出文本编辑器,再重新进入文本编辑器,编辑同样的测试文件,是否一切正常?
- 退出文本编辑器,你是否处于 ACAD 子目录下?如果不是,你能否设置你的文本编辑器,使它可以退出到 ACAD 子目录下?

如果你配置好的文本编辑器在转换到 ACAD 子目录下时仍然功能完备;如果在退出时能进到 AutoCAD 子目录,那么你就能够在图形编辑器中使用它。

你还应了解另一方面:你使用的文本编辑器需占用计算机多少内存(RAM),这个信息可以在你的文本编辑器的文档中找到。文本编辑器通常需要 64KB 至 256KB,查到这个数据并记下来。你可以使用的最大内存是 512KB,如果你的文本编辑器需要更多的内存(不太正常),你必须使用别的编辑器或者在 AutoCAD 的外部使用它。

查阅文本编辑器的文档,看看它在编辑文件时是否会变大——即需要更多的内存。有些文本编辑器会根据你所编辑的文件大小调整内存需要。如果你的文本编辑器是这样的,就按照它的最大需要来计算内存。

§ 1.9.1 修改 ACAD.PGP

现在你可以使用文本编辑器来修改 ACAD.PGP。记住,在编辑时将此文件作为 ASCII 文件(或非文档文件,或文本编辑器用于 ASCII 文件的任意名字)。当文本编辑器找到这个文件,就会在屏幕上出现下面的内容:

```
; acad.pgp — External Command and Command Alias definitions
; Examples of External Commands for DOS
CATALOG,DIR /W, O, File specification:,O
DEL,DEL, O,File to delete:,4
DIR,DIR, O,File specification:,O
EDIT,EDLIN, O, File to edit:,4
SH,, O, * OS Command:,4
SHELL,, O, * OS Command:,4
TYPE,TYPE, O, File to list:,O
```

要保证你是从正确的子目录中调用这个文件的。检查\ACAD\SUPPORT 子目录下是否存在这个文件。

ACAD.PGP 包括一系列 DOS 命令,在此文件中,每个命令占一行,每一行包括五个独立的信息元素,用逗号隔开。这些项目是:

1. 在图形编辑器中采用的命令名;
2. 当图形编辑器的命令发出后,AutoCAD 传给 DOS 的实际命令。
3. 执行这个命令所需要的内存数量。
4. 给用户的信息提示。
5. 当返回图形编辑器时,AutoCAD 使用的特殊数字代码。

将光标移动到以“TYPE,TYPE”开头的那一行的下面,键入你的新 AutoCAD 命令——在图形编辑器中调用文本编辑器时使用。这个命令可以和你的文本编辑器命令相同,或者可以是它的缩写,任何简单的字母组合都行,但是不要使用已有的 AutoCAD 命令名。要用大写字母键入命令名。例如你使用的编辑器叫 TE,就以 TE 为命令名,在命令名后键入一个逗号。

接着,键入你要调用文本编辑器的命令,再敲一个逗号。在这个例子中,命令名是相同的,TE。

然后,输入文本编辑器需要的内存。如果你使用的是 12 版,可以简单地只键入一个零,不然的话,敲入文本编辑器中指出的内存量。在这个例子中要求内存为 128KB,正如你看到的,内存需求要全部键入,如 128000。象“128KB”这样的缩写无效。如果你的文本编辑器象在前面