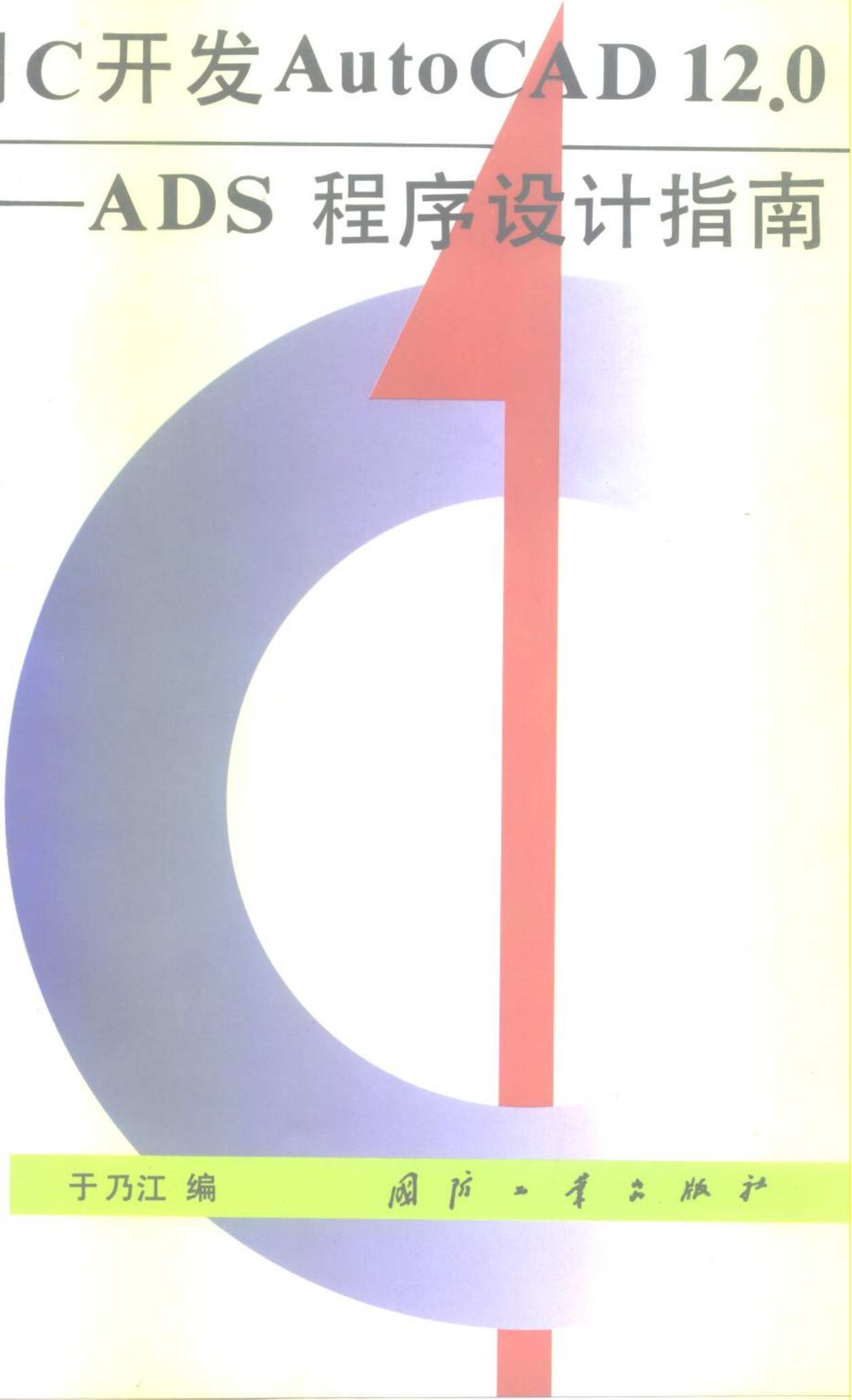
用C开发AutoCAD 12.0

——ADS 程序设计指南



于乃江 编

国防工业出版社

用C开发AutoCAD 12.0
——ADS程序设计指南

于乃江 编

国防工业出版社
·北京·

图书在版编目(CIP)数据

用C开发AutoCAD 12.0-ADS程序设计指南/于乃江编.
北京:国防工业出版社,1995.9
ISBN 7-118-01390-0

Ⅰ.用… Ⅱ.于… Ⅲ.C语言-程序设计-计算机辅助设计-计算机设计自动化 Ⅳ.TP391.72

中国版本图书馆CIP数据核字(94)第15956号

国防工业出版社出版发行
(北京市海淀区紫竹院南路23号)
(邮政编码 100044)
国防工业出版社印刷厂印刷
新华书店经售
*
开本 787×1092 1/16 印张 16¼ 371千字
1995年9月第1版 1995年9月北京第1次印刷
印数:1—5 000册 定价:12.30元

前　言

美国 Autodesk 公司开发的 AutoCAD 软件包是目前在我国微机上最为流行的通用软件之一。该软件以其强大的图形功能、友好的用户界面以及便于二次开发等特点在机械、电子、建筑等领域得到广泛应用和迅速发展。自 1982 年 1.0 版问世以来，Autodesk 公司先后推出了该软件的 2.0、2.17、2.18、2.5、2.6、9.0、10.0 和 11.0 版，1992 年 7 月又推出了最新版本 12.0。

AutoCAD 软件之所以受到人们的青睐，主要原因之一是它具有便于二次开发的特点。从 2.17 版开始，AutoCAD 引入了嵌在内部的 AutoLISP 语言，用户可以利用它进行二次开发。但 AutoLISP 程序存在运行速度慢、保密性差等缺点(因为它是一种解释性语言)，所以用起来不方便。从 11.0 版起，AutoCAD 提供了用 C 语言进行开发的环境，即 ADS(AutoCAD Development System)。在这种环境中，用户可以用目前最流行的计算机高级语言——C 语言进行二次开发。由于 C 语言的程序可以进行编译、连接生成可执行文件，因而具有保密性，运算速度也加快了；同时它可以利用标准 C 的数据结构和库函数，使得程序更简洁、能更有效地利用内存。另外，ADS 还可以访问某些 AutoLISP 所不能访问的系统和设备。正是由于这些优点，越来越多的 AutoCAD 用户倾向于用 C 对该软件进行二次开发。

本书结合大量实例详细介绍了 12.0 版 ADS 的各方面知识，主要包括：ADS 与 AutoLISP 的接口；ADS 的各种文件、数据类型、变量；12.0 版 ADS 的全部库函数；结合实例分析 ADS 程序的设计技巧及初学者易犯的错误；在实模式、保护模式下开发 ADS 应用程序的方法等。本书的内容既适合于初学者，也适合于有过 11.0 版开发经验的读者。

在本书的编写过程中，郭淑芬老师和郭艳军提出了许多宝贵意见并提供了多方面的帮助；陈光、张恩和、信保军、郭建设等老师给予了大力支持，另外还得到赵学红、倪艳、白玲等的关心和帮助，在此一并致以深深的谢意！

由于水平有限，时间紧迫，书中不妥和错误之处敬请广大读者批评指正。

编　者

目　录

第一章 绪 论

1.1 ADS 简介

ADS(AutoCAD Development System)是计算机绘图软件 AutoCAD 从 11.0 版起提供的建立于 AutoCAD 内部的 C 语言程序设计环境。用 C 编成的 ADS 应用程序不能脱离 AutoCAD 环境独立使用,但经过特定的编译器和连接器进行编译、连接后,可以作为 AutoLISP的外部函数装入 AutoCAD 中运行。

与嵌入 AutoCAD 内部的 AutoLISP 语言相比,ADS 功能更强。AutoLISP 只是 LISP 语言的一个子集,而 ADS 则不同,它包含了 C 的全部功能,既不是 C 的一个子集,也不是基于 C 而编制的专用语言(例如 Microstation 的 MDL)。ADS 几乎具备 AutoLISP 语言的所有功能,此外,它还有一些优点:比如,ADS 程序可以编译、连接成可执行文件,因而运行速度快,且具有保密性;它可以利用 C 语言的数据结构,这样程序会更简洁,并能更有效地利用内存。同时 ADS 还可以访问操作系统和硬件等 AutoLISP 所不能访问的系统和设备。在那些需要大量计算或与主机环境交互较多的场合,ADS 更能显示出其优势。

本书将介绍 ADS 的基本知识,包括 ADS 与 AutoLISP 的接口、ADS 的各种组成文件、数据类型、头文件中定义的变量和 12.0 版 ADS 库的全部函数,通过分析一些典型程序说明 ADS 程序的设计技巧,最后介绍如何在实模式、保护模式下编译连接 ADS 程序。

本书的内容是针对 AutoCAD 12.0 版的 ADS。但大部分内容也适用于 11.0 版(除 12.0 版的扩展功能外)。随着 AutoCAD 软件本身的发展,ADS 的功能也将不断扩充,本书的内容也将随之修订。

ADS 具有向上兼容性,在 11.0 版开发的 ADS 如果平台不变,那么在 12.0 版下仍可运行。保护模式下 11.0 版 ADS 程序可以直接在 12.0 下运行;如果是实模式,则需重新编译、连接后才能在 12.0 下运行。关于保护模式与实模式,请参阅第七章.

1.2 ADS 的组成

ADS 由装在同一目录下的一个目标库和 4 个头文件组成。

ADS 目标库由一个文件组成,文件名因平台的不同而有所不同。但无论目标库是什么名字,在编译连接 ADS 程序时都要指明,否则将出错。

ADS 有 4 个头文件,其名称和内容如下:

(1)adslib.h 包含 ADS 一般功能的定义;

(2)adscodes.h 包含传给 ADS 库函数或从库函数返回的代码的定义;

(3)ads.h 包含 ADS 库类型定义和函数说明;

(4)ol_errno.h　包含 AutoCAD 系统变量 ERRNO 所用的错误代码(见附录 C)。

头文件 adslib.h 中包含了 adscodes.h 和 ads.h 的 #include 语句,所以每个源程序只用一个关于 ADS 的 #include 语句即可:

```
#include "adslib.h"
```

ADS 程序一般不必包含 ol_errno.h,除非使用了系统变量 ERRNO 的那些出错代码。

1.3　支持 12.0 版 ADS 的平台

1.3.1　640K DOS (AutoCAD 286)环境

该环境需要如下工具:

(1)编译器　Microsoft C(5.0 版以上),或者 Meta Ware High C(用于 16 位系统的)1.6 版本;

(2)构造环境　Rational System DOS/16M 库的支持文件;

(3)连接器　Microsoft 的覆盖连接程序(link);

(4)调试器　Rational System 的 Instant-D;

(5)构造工具　Microsoft 的程序维护工具(Make)。

其中,(2)和(4)是 Rational System 的 DOS/16M 软件开发包的一部分。

1.3.2　扩展 DOS(AutoCAD 386)环境

该环境需要的工具有:

(1)编译器　Meta Ware High C(用于 32 位系统的)1.5 以上的版本或 7.0 以上版本的 WATCOM C 386;

(2)连接器　Phar Lap 386 |LINK;

(3)调试器　Phar Lap 386 |DEBUG;

(4)虚拟内存管理器　Phar Lap 386 |VMM;

(5)汇编工具　Phar Lap 386 |ASM。

其中(2)、(3)、(4)和(5)是 Phar Lap 386 |DOS-Extender 软件包的一部分,(3)和(4)是任选项。

构成此环境的另一种形式是,利用集编译、连接、调试于一身的 Borland C(或 Borland C++),对一般用户来说,这种环境下开发 ADS 程序较为方便,因为所用的支持软件大家都比较熟悉。

1.3.3　UNIX 环境

该环境所需工具:

(1)编译器　C 编译器(CC);

(2)连接器　连接编辑器(ld);

(3)调试器　dbx(标准调试程序);dbxtool(适于 SUN 工作站);dde(适用于 Apollo 工作站);

(4)构造工具　源程序维护及重新生成工具(make)。

以上几种工具除 dbxtool 外，全部由 UNIX 操作系统提供。dbxtool 是调试程序的基于窗口的版本，它作为 Sun OS 标准操作系统的一个标准命令提供给用户。Domain 调试环境中的 dde 是由 Apollo Domain 系统提供的基于窗口的调试程序。

1.4 12.0 版 ADS 的功能扩展

ADS 12.0 比 11.0 版有了许多改进，在功能上更趋于完善，具体如下所述。

(1)用于筛选实体的函数 ads_ssget()，可以支持组代码的组合、关系判断和逻辑条件，还可以支持栅格(fence)和折线(polyline)选择模式以及 PICKFIRST 集。

(2)函数 ads_initget()可以规定用户输入函数(即 ads_get×××())接受用户的任意输入。

(3)函数 ads_entsel()和 ads_nentsel()不但可以返回拾取点，还可以返回关键字(keyword)。

(4)引入函数 ads_nentselp()，它可以使用一种新的标准转换矩阵类型 ads_matrix。还有几个新增加的函数(ads_draggen()、ads_grvecs()、ads_xformss())也可以使用这种类型，函数 ads_nentselp()还允许程序中指定实体拾取点。

(5)引入函数 ads_draggen()，这个函数可以提示用户以交互方式对选择集中的实体进行平移、旋转或比例变换。

(6)引入函数 ads_grvecs()，该函数允许用户通过一次调用在图形屏幕上画多个向量。函数 ads_grread()对输入条件的控制也有所加强。

(7)引入函数 ads_xformss()，它使用户不必调用 AutoCAD 命令或使用 ads_entmod()就可以对选择集中的实体进行平移、旋转或比例变换等操作。

(8)引入函数 ads_angtof()和 ads_distof()，它们分别可以将代表角度、距离的量变为双精度浮点型。

(9)引入函数 ads_textbox()，它可以获取 Text、Attdef、Attrib 等实体的范围。

(10)引入函数 ads_tablet()，用于控制图形输入板的标定。

(11)引入函数 ads_alert()，它可以显示一个警示框，框中的信息由程序中提供。

(12)引入函数 ads_getfiled()，它可以显示标准 AutoCAD 文件对话框来提示用户。

(13)引入函数 ads_xload()、ads_xunload()、ads_loaded()，用它们可以装入或移去其他的外部 ADS 函数。

(14)引入函数 ads_getsym()和 ads_putsym()来获取或设置 AutoLISP 符号的值。

(15)引入函数 ads_rett()，ADS 程序可以用它返回 AutoLISP 符号 t(true)，并且可以把这个符号以 RTT 形式放到结果缓冲器表中。

(16)使在 AutoLISP 表中设定点对(dotted pairs)的方法变得更可靠和直观。

(17)引入函数 ads_vports()，该函数可以获取当前 AutoCAD 配置的视口描述信息表。

(18)增加了一系列字符处理函数，使 ADS 对字符的处理更加方便。

(19)12.0 ADS 不再把标准 C 的库函数 exit()定义为 ads_exit()。如果以前编的 ADS 程序中使用了 exit()，现在要改为 ads_exit()。

另外12.0版AutoCAD的一些标准外部ADS程序还提供了一些实用外部函数，主要有两类：

(1)外部定义的AutoCAD函数

这类函数包括acad_helpdlg、acad_colordlg和acad_strlsort，它们分别用来显示标准AutoCAD帮助框和色彩选择对话框以及按字母顺序列出字符串。

(2)与交互式AutoCAD命令接口的函数

这类函数用C：×××形式调用交互式的AutoCAD命令，这些命令主要有：BHATCH、BPOLY、PSDRAG、PSIN、PSFILL等。详见5.21和5.22节。

1.5 由AutoLISP访问ADS程序

1.5.1 启动AutoCAD后装入ADS程序

利用AutoLISP的(xload)函数可以装入ADS应用程序(注意，必须是经过编译连接的可执行文件)，这与用(load)函数装入AutoLISP程序类似。该函数的调用方式为：

(xload filename [onfailure])

函数查找filename指定的文件，将它调入内存，并立即执行它的初始化部分，而不是像(load)那样执行整个程序。

例如，要装入一个编译过的名为myapp.exp的ADS应用程序，调用(xload)的形式为：

Command:(xload "myapp")

(xload)将自动为指定的ADS应用程序加上扩展名(.exe或.exp，视平台而定)。如果(xload)找到了指定的程序，并成功地装入了，它将返回该程序名(在上例中，将返回"myapp")。如果没能成功地装入，将显示出错信息。若设定了参数onfailure，装载失败时将返回[onfailure]的内容：可以是字符串或原子值，或者是一个函数(返回函数值)。

装入ADS程序时，AutoLISP将检验所用ADS库的版本是否与当前AutoCAD版本一致。通过多次重复调用(xload)，用户可以装入多个ADS程序，最多可达255个。一个大型ADS程序可以分成多个程序分别装入。

装入ADS程序时，如果不给定路径，(xload)将按AutoCAD库路径所定义的目录来查找指定的ADS程序。AutoCAD库路径包括以下目录：

(1)当前目录；

(2)包含当前图形文件的目录；

(3)如果定义了ACAD环境变量，则为该变量指定的目录；

(4)包含AutoCAD程序的目录。

该路径与AutoCAD查找菜单和其他支持文件以及(load)查找AutoLISP文件的路径一样。根据当前的环境，两个或多个库路径的目录可以相同。如果输入全程路径名，例如"e：/ynj/myapp"，则(xload)就不再查找其他目录。

不同平台路径名的格式也不相同。我们知道MS-DOS中使用反斜线("\")来分隔目录名，而AutoCAD允许输入正斜线("/")来代替反斜线。如果要用反斜线，则需要连续输入两次，如"e：\\ynj\\myapp"。

要想查看当前有哪些 ADS 程序已装入 AutoCAD，可以在 Command：提示符后输入(ads)函数，该函数返回一个字符串表，其中的每个字符串即是一个已经装入的 ADS 程序名。

1.5.2 在 AutoCAD 初始化时装入 ADS 程序

除了在 AutoCAD 的命令行通过(xload)装入 ADS 程序外，还可以在 AutoCAD 初始化过程中装入。在启动 AutoCAD 时，它首先在 AutoCAD 目录下查找名为 acad.ads 的文件，如果找到，则将其中列出的每个 ADS 文件装入。装入成功后，只要进入图形编辑状态，AutoCAD 就会给出相应的提示。

用户每次关闭当前图形文件打开另一个文件时，AutoCAD，包括 AutoLISP 都要重新初始化，但 ADS 程序不必重新装入。

acad.ads 文件是个文本文件，由一个或多个文件名组成：每个文件名要占一行，扩展名可有可无。如果有扩展名，那么这个扩展名就不再变化；如果没有，AutoCAD 将为它加上一个系统隐含的扩展名。当文件名中指定了路径时，AutoCAD 就在指定的路径中查找，否则就像前边所讲的按当前库路径查找——不过在初始化时，没有当前图形目录。

系统在启动时装入 ADS 程序的另一种方法是定义一个直接调用(xload)的 AutoLISP 自动执行函数，如例 1-1。

〔例 1-1〕

```
(defun S::STARTUP()
   (if    (not (ads))
          (xload "application")
          .
          .
          .
   )
)
```

1.5.3 移去 ADS 应用程序

利用(xunload)函数可以移去已装入 AutoCAD 的 ADS 程序，形式如下：

(xunload filename)

其中 filename 是要移去的 ADS 程序名。比如，要移去前边已经装入的“myapp.exp”文件，可以这样：

```
Command:(xunload "myapp")
```

在两种情况下 AutoLISP 自动移去 ADS 程序：一是程序本身通过 ads_abort()函数报告出现严重错误；二是在退出 AutoCAD 时。当用户结束当前图形编辑状态而退不出 AutoCAD 时，ADS 程序不会自动移去，仍处于加载状态。

在内存受到限制的平台上开发大型的 ADS 程序，可以编成多个文件，按需要随时对文件进行加载和卸载。

1.6 ADS 函数的一般特点

ADS 库函数是在头文件 ads.h 中进行说明的。库函数和大多数数据类型的定义是以 ads_开头的，还有一些以 adsi_开头，是作为内部使用的。ADS 程序不能调用 AutoLISP 程序，但它可以调用其他 ADS 程序中所定义的外部函数，详见2.1.5节。

1.6.1 ADS 函数与 AutoLISP 函数的比较

ADS 环境下的一些库函数是 AutoLISP 所没有的，但也有一大部分 ADS 函数的功能与 AutoLISP 函数功能相同，并且函数名也比较接近，只不过 ADS 库函数大多带有前缀 ads_。这样，AutoLISP 程序可以比较容易地改成 C 程序。关于 AutoLISP 和 ADS 函数的对照，参见附录 A。

1. AutoLISP 函数与 ADS 函数的自变量表

许多 AutoLISP 函数自变量的数目允许是任意的，但要求 ADS 库函数都有可变长度的自变量是不可能也是不必要的。所以 ADS 库有一个规定：ADS 库函数取与之相应的 AutoLISP 函数的所有自变量。对 AutoLISP 中的任选项，如果 ADS 函数没有选用，则传给一个特定的值(通常为 NULL 指针，或者一个整型值0或－1)。这样，习惯于使用 AutoLISP 的读者也能很快编出 ADS 程序了。

但有几个 ADS 库函数不符合这个规定：

(1)ads_printf()函数

这个函数与标准 C 的 printf()函数相似，是作为一个不定型函数来执行的，即该函数的自变量表是可变长度的。

(2)ads_command()和 ads_cmd()函数

AutoLISP 的(command)函数不仅允许自变量的个数可变，还可以接受专门为 AutoCAD 定义的类型(比如点和选择集)。为了适应这种情况，ADS 的 ads_command()中使用了可变长度的自变量表，用自变量指定传给 AutoCAD 的值的类型；ads_cmd()的自变量要求与前者相似，只不过是以链表来传递的。因此这两个函数的自变量表与 AutoLISP 的(command)函数不完全一样。

(3)ads_entget()函数

AutoLISP 中，(entget)函数提供了一个可选的自变量来获取扩展实体数据，但在 ADS 中，ads_entget()函数没有这个自变量，不过提供了另外一个函数 ads_entgetx()可以专门用于获取扩展数据。

2. 关于内存的使用

ADS 程序对内存的要求与 AutoLISP 程序不同。一方面，C 程序中的数据结构使它比 AutoLISP 更简洁；另一方面，运行 ADS 程序需要比较大的辅助系统——目标库。目标库大小随平台不同而不同，对 AutoCAD 386来说大约56～87 KB(具体数值与编译器也有关)。

每一个 ADS 程序都要包括与 AutoLISP 的接口，还要与 ADS 库相连接，所以将有联系的外部函数写到一个程序中可以减少内存的使用和装入的时间。装入大量的小程序将

会影响 ADS 程序的运行。

有些 ADS 函数自动分配内存。大多数情况下，如果 ADS 程序本身已分配了内存，那么它必须自己释放，否则会降低系统的性能甚至使 AutoCAD 锁死(参见2.3和6.2节)。

1.6.2 函数返回值与函数结果的比较

ADS 库函数的返回值有的为空(void)，有的直接返回函数结果，但大多数返回值类型为整型的状态码，用来指示函数调用是否成功。这些状态码定义在 adscodes.h 文件中。如果被调用的函数有一个实际的结果值，它将以间接方式传递的自变量返回给调用函数。

要注意区别库函数的结果自变量与返回值的区别。函数返回的是一个整型的状态码，但它的结果值却是放在自变量中传回到调用函数中的。

现在给出几个典型的 ADS 库函数的原型说明：

```
int ads_entnext(ads_name ent,ads_name resault);
int ads_osnap(ads_point pt,char * mode,ads_point result);
int ads_getint(char * prompt,int * result);
```

在 ADS 程序中可以通过下面的语句调用上面给出的函数：

```
stat=ads_entnext(ent,entres);
stat=ads_osnap(pt,mode,ptres);
stat=ads_getint (promt,&intres);
```

这几个函数都被调用以后，变量 stat 的值就可以表明函数调用是否成功：如果成功，stat 值为 RTNORM；如果失败则为 RTERROR 或 RTCAN。函数最后一个自变量必须是以间接方式传递的结果自变量。如果调用成功，ads_entnext()将在其 entres 中返回一个实体名；ads_osnap()将在 ptres 中返回一个点；ads_getint()将在 intres 中返回一个整型结果。由于 ads_name 和 ads_point 是数组类型，所以自变量 entres 和 ptres 不能直接以指针形式出现。

第二章　ADS 库函数使用基础

本章将要介绍 ADS 程序的结构、ADS 库定义的数据类型及 ADS 环境下表的处理。通过对这几个方面的介绍，读者就会看出 ADS 编程方法与其他 C 程序的异同。

2.1　ADS 程序的结构

ADS 程序一定要支持由 ADS 所定义的与 AutoLISP 的接口程序。接口程序要求 ADS 程序用一定的值，以一定的顺序调用一定的 ADS 库函数。本节向读者介绍 AutoLISP 接口程序所用的库函数和这些库函数所用的经过定义的值及 main()函数的结构。

AutoLISP 访问 ADS 程序的顺序如下：

(1)AutoLISP 在初始化时或调用(xload)、ads_xload()时装入 ADS 程序；

(2)ADS 程序利用函数 ads_init()实现与 AutoLISP 通信的初始化；

(3)ADS 程序以结果码 RSRSLT 调用 ads_link()函数，通知 AutoLISP 已准备好执行 AutoLISP 请求；

(4)ads_link()返回 AutoLISP 的请求码 RQXLOAD；

(5)ADS 程序通过 ads_defun()定义它的外部函数；

(6)ADS 程序再一次用 RSRSLT 调用 ads_link()，如果定义函数出错，则用 RSERR 调用 ads_link()；

(7)ads_link()返回 RQSUBR(或其他请求码)时，ADS 就可以对外部函数进行相应处理；

(8)处理完外部函数后，再重复步骤(6)。

可见，ADS 程序通过调用 ads_link()后，就随时准备执行来自 AutoLISP 的请求，比如执行外部函数或进行其他操作。而在 ADS 程序执行 AutoLISP 请求时，AutoCAD 和 AutoLISP 都处于非激活状态，等待 ADS 库函数的请求，此时任何用户输入都不会得到响应。

由于 ads_link()函数要被 ADS 程序反复调用，所以可以把它放在一个无限循环的开始部分，循环中利用 switch 语句来判断并处理各种不同的 AutoLISP 请求。循环体一般放在程序的 main()函数中，如例2-1所示。

〔例2-1〕

```
/* MAIN—主程序 */
void
main(argc,argv)
  int argc;
```

```
int *argv[ ];
{
  int stat;
  short scode=RSRSLT;              /*缺省结果码*/
  ads_init(argc,argv);             /*初始化接口*/
  for(;;)                          /*循环条件*/
  if ((stat=ads_link(scode))<0)
    {
        printf("TEMPLATE:bad status from ads_link()=%d\n",stat);
                                   /*不能用ads_printf()显示这条信息,
                                     因为连接没有成功*/
        fflush(stdout);
        exit(1);                   /*仅当异常中断时使用exit()*/
    }
    scode=RSRSLT;                  /*缺省返回值*/
                                   /*下面的switch语句检查AutoLISP的请求码*/
    switch(stat)  {
        case RQXLOAD:
          scode=loadfuncs()==GOOD ? RSRSLT:RSERR;
          break;
        case RQSUBR:               /*此处一般是选择这个程序定义的某个外部函数*/
          break;
        case RQXUNLD:              /*对这些AutoLISP请求,*/
        case RQSAVE:
        case RQEND:                /*可以返回RSRSLT。如果*/
        case RQQUIT:               /*不需直接处理,则不必*/
        default:                   /*写出这4个语句*/
        break;
    }
  }
}
/*LOADFUNCS—定义外部函数*/
static int loadfuncs()
{
    return GOOD;                   /*一般对每个外部函数调用一次ads_defun()*/
}
```

例2-1中的GOOD(或BAD)在本书的程序中常作为返回值出现(尤其是用做出错处理代码),它不是由ADS库定义的,用户可以自行定义。

例2-1这个程序是AutoCAD所提供的示例程序,读者自己编写ADS程序时可以以此为样本,从而保证与AutoLISP接口的正确性。

由于ADS程序执行过程中AutoCAD和AutoLISP是非激活态,所以ADS程序要尽量避免和用户没有交互作用的长时间的计算。如果必须要有长时间的计算,那么至少也应