

方 铁 编 著

AutoCAD C语言高级编程

- 覆盖 AutoCAD R11和 R12
- 从编程基础到高级技巧的全面介绍
- 最新的 AutoCAD 对话框编程方法和 Windows 环境下 AutoCAD 的开发
- 提供丰富的开发实例



清华大学出版社

AutoCAD C 语言高级编程

方 铁 编 著

清华大学出版社

(京)新登字 158 号

内 容 简 介

JS115/24

本书以AutoCAD R12(DOS版和Windows版)为基础,对AutoCAD C语言编程系统做了系统的介绍。书中剖析了ADS编程的难点和高级技巧,并提供了ADS高级编程、ADS对话框编程、Windows环境ADS编程和ADS库函数等方面的详细资料和丰富实例。全书由浅至深,脉络清晰,实例和理论知识结合贯穿全书,易于学习和掌握,并具有较高的参考价值。

图书在版编目(CIP)数据

Auto CAD C 语言高级编程/方铁编著.-北京:清华大学出版社,1995.1
ISBN 7-302-01765-4

I. A… II. 方… III. 计算机辅助设计-C语言-程序设计
IV. TP391.72

中国版本图书馆 CIP 数据核字(95)第 01525 号

AutoCAD C 语言高级编程

方铁 编著

*

清华大学出版社出版

(清华大学校内 邮编100084)

新华书店总店北京科技发行所发行

清华大学印刷厂印刷

*

1995年1月第一版 1995年1月第一次印刷

开本: 787 × 1092毫米 1/16 印张: 23 $\frac{7}{8}$ 字数: 533千字

印数: 1 - 7000册 定价: 21.00元

ISBN 7-302-01765-6

1100150

前 言

AutoCAD是目前世界上最成功的微机CAD软件平台，这不仅是因为它支持的平台多，外设多，价格低，易学易用，更主要的原因是它具有开放的体系结构。AutoCAD允许用户和开发者在几乎所有方面对其进行扩充和修改，能最大限度地满足用户的特殊要求。AutoCAD最强有力的扩充手段就是支持高级语言编程。

众所周知，AutoCAD从2.18版起引入了AutoLISP语言，该语言与AutoCAD环境融为一体，功能强大，是二次开发的有效手段。到目前为止，已经开发了众多基于AutoLISP语言的应用软件。然而，AutoLISP语言是解释执行的，运行效率低，不适合于开发计算密集型的CAD软件，并且软件加密也是个难题。从11.0版起，AutoCAD提供了它的C语言开发系统ADS。ADS直接利用用户熟悉的各种流行的C编译器，将应用程序编译成可执行文件在AutoCAD环境下运行，从而既利用了AutoCAD环境的强大功能，又利用了C语言的结构化编程、运行效率高的优势。ADS使以前用AutoLISP语言不能做或做起来很困难的工作变得轻而易举，比如：大规模的计算，动态图形显示，直接对宿主操作系统的访问，直接对外部设备进行操纵，等等。由于C语言非常适合于编制较大规模的程序，因此我们可以利用C语言和ADS制作一个完整的专业CAD系统，即：

Professional CAD system = C+AutoCAD

这意味着，AutoCAD C语言开发系统在机械设计、工程分析、建筑结构、土木工程、化学工程、电器工程等各种需要大量交互式计算与绘图的应用领域中都将有广阔的应用前景。

AutoCAD C语言开发系统具有许多先进特性和前所未有的强大功能，但它同时也对程序开发人员提出了更高的要求：开发者不但要有专业知识，还要熟悉AutoCAD环境和C语言编程。本书的编写始终遵循深入浅出、循序渐进的原则。第一章和第二章主要讲解ADS的基本知识和基本编程技巧；第三章详细讲述ADS系统中的各个难点和高级编程技巧；第四章覆盖了对话框编程所需的全部知识；第五章通过对几个完整实例的讲解使读者能够综合运用前面几章的知识；第六章讲述AutoCAD for Windows的开发方法，使读者进入一个AutoCAD开发的新境界；第七章给出了全部ADS库函数的详解，以备读者随时查找。全书的宗旨就是帮助各个专业领域的工程科技人员和科研人员迅速掌握AutoCAD C语言开发系统，并很好地利用这一强大工具完成高水平的CAD应用系统开发。

书中内容原则上适用于AutoCAD R11以上版本，但整体编写以AutoCAD R12为基础。

编者

1994年12月

目 录

第一章	AutoCAD C 语言开发系统——ADS	1
1.1	AutoCAD C 语言开发系统简介	1
1.2	ADS应用程序开发环境	2
1.2.1	ADS头文件和目标库	2
1.2.2	AutoCAD R11支持的编译器和连接器	2
1.2.3	AutoCAD R12支持的编译器和连接器	3
1.2.4	AutoCAD R12 for Windows支持的编译环境	3
1.3	创建ADS应用程序	3
1.3.1	在保护模式开发环境下创建ADS应用程序	3
1.3.2	在实模式开发环境下创建ADS应用程序	6
1.4	ADS应用程序的加载与执行	8
1.4.1	在图形编辑器中加载ADS应用程序	8
1.4.2	在AutoCAD初始化时加载ADS应用程序	8
1.4.3	调用ADS外部函数	9
1.4.4	检查当前已加载的ADS应用程序	9
1.4.5	ADS应用程序的卸载	9
第二章	ADS 编程基础	10
2.1	编写第一个ADS程序	10
2.1.1	HelloADS.c源程序	10
2.1.2	HelloADS.c源程序说明	12
2.1.3	ADS函数调用	12
2.1.4	大写字母标识符	12
2.1.5	HelloADS.c源程序分析	13
2.1.6	运行HelloADS	17
2.2	ADS程序的工作流程	17
2.3	ADS中定义的变量、类型和常量	18
2.3.1	实数类型	18
2.3.2	点类型	18
2.3.3	常用常量	19
2.3.4	ADS库中定义的结果类型码	19
2.3.5	ADS库函数的结果码	19
2.4	程序设计实例	20
2.4.1	设计目标	20
2.4.2	程序parts1.c	21
2.4.3	程序parts1.c分析	24
2.4.4	运行parts1	26
2.4.5	对parts1.c进一步的改进	27
2.4.6	程序parts2.c	27
2.4.7	程序parts2.c分析	30
2.5	本章小结	31
第三章	ADS 高级编程	32
3.1	基本概念	32
3.1.1	实体	32

3.1.2	实体名	33
3.1.3	实体标号(实体句柄)	33
3.1.4	选择集	33
3.2	DXF文件和DXF组码	34
3.2.1	DXF文件的总体结构	34
3.2.2	DXF组码与组值	35
3.2.3	标题节	36
3.2.4	表节	37
3.2.5	块节	37
3.2.6	实体节	37
3.3	结果缓冲器的构成	38
3.4	构造选择集	39
3.5	符号表的存取	50
3.6	有名块与无名块的生成	54
3.7	实体数据的管理	62
3.8	扩充实体数据的管理	80
第四章	高级用户接口与对话框编程	87
4.1	AutoCAD R12的对话框编程概述	87
4.1.1	对话框的运行方式	87
4.1.2	平台间的可移植性	87
4.1.3	为什么要用对话框作为应用程序的用户接口	88
4.1.4	如何控制对话框	88
4.1.5	设置对话框颜色	89
4.2	一个对话框编程实例	89
4.2.1	DLGTEST源程序	89
4.2.2	DLGTEST的功能与作用	103
4.2.3	DLGTEST程序说明	103
4.3	对话框的组成Tile和Tile原型	105
4.3.1	预定义的可激活Tile原型	106
4.3.2	Tile组(Tile Cluster)	108
4.3.3	装饰性和表示信息的Tile	109
4.3.4	Tile的属性	110
4.4	DCL语言	119
4.4.1	一个对话框DCL文件的例子	119
4.4.2	DCL文件的结构	121
4.4.3	DCL语法	122
4.4.4	DCL技巧	125
4.5	对话框C语言编程接口	129
4.5.1	对话框操作函数的调用顺序	129
4.5.2	对话框回调函数	130
4.5.3	ADS定义与声明	134
4.5.4	处理Tile	136
4.6	对话框函数参考	143
4.6.1	对话框函数分类摘要	

5.1.1	程序设计思想	152
5.1.2	FTRIM.c的源程序	152
5.1.3	FTRIM.c源程序说明	172
5.2	图形文件的管理	174
5.2.1	设计目标	174
5.2.2	用户接口设计	175
5.2.3	graphman.c源程序	177
5.2.4	graphman.c源程序说明	212
5.3	模拟天体的运行	213
5.3.1	快速浏览	213
5.3.2	gravity.c源程序	214
5.3.3	背景知识	214
5.3.4	程序功能介绍	215
5.3.5	程序设计技巧	216
5.4	编程常见错误分析	219
5.4.1	数据结构中指针的错用	219
5.4.2	选择集释放不成功	220
5.4.3	选择集索引说明为长型不成功	221
5.4.4	函数调用时缓冲器释放失败	221
第六章	AutoCAD/Windows 版的开发与应用	222
6.1	AutoCAD R12/Windows版的新特性	222
6.2	Windows下的ADS应用程序开发环境	223
6.3	移植Helloads.c到Windows环境	223
6.4	脱离ADS	225
6.5	利用Windows资源的ADS程序	230
6.5.1	ADSW程序的结构	230
6.5.2	对话框资源的使用(具有Windows特征的ADSW程序之一)	245
6.5.3	窗口资源的使用(具有Windows特征的ADSW程序之二)	254
6.6	集成开发环境下ADSW程序的开发与调试	260
6.6.1	集成开发环境下的程序开发	260
6.6.2	集成开发环境下的程序调试	261
第七章	库函数详解	262
7.1	库函数分类摘要	262
7.2	ADS库函数详解	269
附录 A	系统变量列表	324
附录 B	DXF 组码	336
	1.以数字顺序表示的组码	336
	2.所有实体都可共享的群组码	338
	3.实体组码(按实体类型名称)	338
	4.块和表的组码	343
附录 C	AutoCAD R12 命令一览表	348
附录 D	AutoLISP与ADS函数对照览表	370
参考文献		375

第一章 AutoCAD C 语言开发系统——ADS

1.1 AutoCAD C 语言开发系统简介

AutoCAD C语言开发系统——ADS(AutoCAD Development System)是AutoCAD R11最显著的功能之一，在AutoCAD R12中这一功能得到了进一步的增强。ADS使外部可执行文件与AutoCAD核心紧密地联系在一起，为在AutoCAD上进行二次开发提供了一个更为强有力的编程手段。

低版本AutoCAD的编程语言一直是AutoLISP语言。AutoLISP语言完全和AutoCAD环境融为一体。用户可以随时使用AutoLISP语言。AutoLISP语言是被解释执行的，任何一个语句键入后就能马上执行，所以它对于交互式的程序开发是非常便利的。事实上，在AutoCAD的命令行提示状态下，我们仍然处于AutoLISP语言的“读入—求值—打印”循环中。

但AutoLISP语言又有严重的不足。AutoLISP语言的缺点之一是由于继承了LISP语言的编程规则而导致的繁多的括号，所以经常有人把“LISP”解释为“Lots of Insipid Stupid Parenthesis”，而不是它的本意“List Processing Language”。

在应用系统的开发过程中，许多常用的功能和手段是AutoLISP语言无法提供的，它们包括：

- 二进制文件的读写
- 在内存中相互通信
- 实时的数据库存取
- 直接的屏幕I/O操作
- 高级的用户接口
- 调用操作系统的功能
- 直接存取硬件设备
- 高强度的数据处理
- 程序的快速执行
- 软件与数据的加密保护

可以说，AutoLISP语言的根本限制是由于它完全包含在AutoCAD之中。

ADS的引入完全克服了上述的AutoLISP语言的缺点。ADS把外部可执行程序 and AutoLISP语言连接起来，将它作为AutoLISP语言的外部函数来解释。由于ADS应用程序是外部可执行的应用程序，所以事实上它几乎可以完成任何事情。

用C语言编程，二进制的文件读写功能是C语言运行函数库的基本功能之一；在内存中相互通信只不过是把一个指针指向合适的内存位置，然后写入一些数据(当然，内存保护是需要考虑的)；数据库存取操作可以通过调用现有的数据库操作函数(如ASI接口)来实现；屏幕I/O操作是现代C语言的标准特征之一，很容易实现；高级用户接口也

很容易实现，可以利用各种现有的软件包，只要做适当修改，使其和AutoCAD的屏幕管理不冲突即可；至于运行速度，任何编译型的语言都要比解释执行的语言快许多。

ADS是一组可以用来用C语言编写AutoCAD应用程序的头文件和目标库。ADS应用程序是在AutoCAD环境中运行的可执行文件，它和AutoCAD建立通讯联接，向AutoCAD发出命令，并获得命令执行的结果。同时，ADS应用程序拥有C语言运行函数库的全部功能。

1.2 ADS应用程序开发环境

开发ADS应用所需的环境包括ADS提供的头文件和目标库，能支持ADS和AutoCAD平台的操作系统和编译开发工具。

1.2.1 ADS头文件和目标库

每一个ADS应用程序都应该包含三个头文件：

- .ads.h
- .adscodes.h
- .adslib.h

为方便起见，在adslib.h中包含了ads.h和adscodes.h，所以在在一个ADS应用程序中只要包含adslib.h就够了。

每个头文件都包含了不同的说明信息：

ads.h

1. ADS特有数据类型的定义
2. ADS库函数的函数原型说明
3. ads_initget()的位码的定义
4. 其它定义

adscodes.h

1. 对请求码、结果码、结果类型码的定义
2. 其它定义

在AutoCAD软件包中，为不同的开发环境提供了不同的目标库，但没有提供目标库的源码。ADS应用程序必须连接合适的目标库才能运行。在1.3节中将指出各种不同的开发环境分别对应哪些目标库。

1.2.2 AutoCAD R11支持的编译器和连接器

AutoCAD R11只支持保护模式开发环境，包括：

- .MetaWare High C 386 (Version 1.6x, 1.71, 1.73, not 2.3)
- .Watocom C 386 7.0

.PharLap 386Link (Wlink也行)

1.2.3 AutoCAD R12支持的编译器和连接器

AutoCAD R12既支持保护模式开发环境, 也支持实模式开发环境.

保护模式开发环境包括:

.Watcom C 386 9.0 (不再需要PharLap Linker)

.MetaWare High C 386 1.x或3.x

PharLap 386Link 4.1

MetaWare debugger MDB 1.3

.Zortech 386 C++ 3.x

PharLap 386Link 4.1

实模式开发环境包括:

.Borland C++ 2.0以上版本

.Microsoft C 6.0以上版本

1.2.4 AutoCAD R12 for Windows支持的编译环境

.Borland C++2.0以上版本

.Microsoft C++7.0

.Microsoft Quick C for Windows

.Watcom C 9.0

.Visual Basic 2.0

.MetaWare High C/C++3.0

1.3 创建ADS应用程序

创建ADS应用程序可在许多平台和环境下进行, 但目前最常用的还是DOS平台. 因为国内AutoCAD绝大多数都运行在386, 486微机的DOS操作系统上, DOS下又有众多流行的C编译器可供选择, 所以本节内容主要是针对DOS平台的, 但是ADS的编程思想和方法对所有平台都是一致的.

如同标准的C函数库一样, ADS由库函数和头文件所定义. 目标库和头文件都安装在AutoCAD的ADS子目录下. 库文件根据所用编译器有不同的名字, 但头文件都是一样的.

1.3.1 在保护模式开发环境下创建ADS应用程序

保护模式开发环境是AutoCAD最主要的程序开发环境. 在微处理器的32位保护模

式下运行ADS应用程序, 可以直接利用扩展内存, 获得较快的运行速度, 并且, 在一次AutoCAD运行期间, 允许同时装入多个保护模式的ADS应用程序, 这些都是实模式ADS应用程序所不能比拟的. 目前最流行的保护模式开发环境是MetaWare High C和Watcom C, 其中High C还必需用扩展DOS支持.

下面我们将用详细的例子来说明如何编译并连接一个保护模式ADS应用程序(如果你想了解更多的编译细节, 可以参考相应编译器的参考手册). 需要注意的是应该连接哪一个库:

ads.lib: MetaWare High C的保护模式ADS库
wcads90.lib: Watcom C的保护模式ADS库

*MetaWare High C 386 1.x或3.x

下面给出在MetaWare High C环境下创建ADS应用程序的批处理文件, 所得可执行文件带有符号调试信息, 便于用MetaWare High C的集成调试器MDB调试.

mkads.bat

```
if x%1x == xx goto usage
if exist %1.err del %1.err
if not exist %1.c goto nosource
if not exist %2\highc\bin\hc386.exe goto nocomp
%2\highc\bin\hc386 %1.c -c -g -I%2\highc\inc -Hobject=%1.obj
                                                    >%1.err

if errorlevel 1 goto error
%2\phar386\bin\386link -cvsym ads.lib %1.obj -lib
                                                    %2\highc\small\hcc -exe %1

echo off
echo
echo
echo %1.EXP created.
goto exit
:error
echo off
echo
echo **Error**
echo Error occurred during the compilation!
echo Review %1.ERR for the details.
goto exit
:nosource
echo off
echo
echo **Error**
echo No %1.C source file found!
goto exit
:nocomp
```

· 4 ·

1101179

```

echo off
echo
echo **Error**
echo No compiler found!
goto exit
:usage
echo off
echo
echo Usage: mkads CCC [DDD]
echo     where CCC is your ADS C source file.
echo           DDD is an optional. Drive which contains compiler
                        and linker.
echo           Defaults to current drive.
echo
echo Example: 1. mkads jis
echo           2. mkads jis f:      [f: drive contains
                                   compiler and linker]

:exit
echo

debugads.bat

@echo off
mdb acad

```

***Watcom C 386 9.0(不再需要PharLap Linker)**

下面给出在Watcom C环境下创建ADS应用程序的批处理文件，所得可执行文件带有符号调试信息，便于用Watcom C的集成调试器wvideo调试。

```

mkads.bat

@echo off
set WATCOM=E:\WATCOM\
set PATH=C:\DOS;E:\WATCOM\BIN;E:\WATCOM\BINB;C:\TOOLS;F:\ACAD12
set INCLUDE=E:\WATCOM\H
set ACAD=F:\ACAD12\SUPPORT;F:\ACAD12\FONTS;F:\ACAD12\ADS
set ACADCFG=F:\ACAD12\CFG
set ACADDRV=F:\ACAD12\DRV
set DOSX=-priv
@echo on
wcc386p -d2 %1 -fpi287 -3s -s -oait
wlinkp debug all system ads file %1 library wcads90

debugads.bat

@echo off
wvideo /TRap=ADS /SWAP

```

1.3.2 在实模式开发环境下创建ADS应用程序

实模式开发环境是AutoCAD R12最新推出的支持Borland C和Microsoft C等常用编译器的开发环境。它使用和保护模式相同的头文件，但对于不同的编译器则有不同的库。实模式开发环境使得开发者可以使用自己熟悉的编译器开发实模式ADS应用程序。实模式ADS应用程序总是自动释放掉被标准C库函数malloc()、calloc()和realloc()分配的空间；在其它情况下，应用程序必须负责释放掉它所分配的空间。在文件处理上，应用程序应负责关闭它所打开的文件，否则打开的文件句柄会继续占据文件句柄表的空间，使得其它应用程序不能再打开文件。需要注意的是，实模式ADS应用程序必须用大模式编译。

下面我们将用详细的例子来说明如何编译并连接一个实模式ADS应用程序(如果你想了解更多的编译细节，可以参考相应编译器的参考手册)。需要注意的是应该连接哪一个库：

mcsads6.lib: Microsoft C的实模式ADS库
tbcads3.lib: Borland C的实模式ADS库

*Microsoft C

你可以用Microsoft C 5.1或更高的版本编译ADS应用程序，编译器命令是cl，它同时也能够连接应用程序。举例说明，如果你的应用程序名是hello.c，则下面的命令行可以编译它并连接产生最后的可执行文件hello.exe：

```
cl /AL /EM /DADS /FPi87 /Zi /Fehello hello.c /link llibc7  
mcsads6.lib /NOE /NOI /CO
```

各编译开关意义如下：

- /AL 确定适用大模式编译，这对于实模式ADS应用程序是必须的。
- /EM 通知编译器在编译源程序时使用扩展内存
- /DADS 定义符号“ADS”，ADS的某些头文件将需要它
- /FPi87 产生协处理器专用浮点指令
- /Zi 通知编译器在目标文件中包含行号和完整的符号信息
- /Fehello 将可执行文件取名为hello.exe
- /link 通知编译器在完成编译后，自动连接并提供开关：
 - llibc7 包括实模式运行函数库
 - mcsads6.lib 包括实模式ADS库
- /NOE 禁止连接器搜索内部符号表
- /NOI 保持大小写敏感性
- /CO 通知编译器在可执行文件中包含调试信息

当应用程序调试完毕后，应该用以下开关对程序进行优化：

`/Oait` 不考虑别名使用, 产生内部函数, 执行循环优化和速度优化

`/Gs` 取消堆栈检查

对于复杂的应用程序, 应该使用Microsoft C的make工具进行编译和连接.

*Borland C

你可以用Borland C 2.0或更高版本编译实模式的ADS应用程序, 编译命令是**bcc**, 连接命令是**tlink**. 假定我们还是编译hello.c程序, 可用以下命令行:

```
bcc -I\ads;\borlandc\include -AT -l -ml -vi -f287 -N- -G  
-DTURBOC -DPROTOTYPES -DRMADS -c hello.c
```

```
tlink /x /d /c /Tde /v c01.obj  
hello.obj,hello.exe,,tbcads3.lib FP87 mathl cl
```

各编译开关意义如下:

`-Iads;\borlandc\include`

指明编译寻找包含文件的路径是\ads和\borlandc\include

`-AT` 通知编译器识别Borland C的特有保留字, 如“cdecl”、“near”、“far”和“asm”等

`-l` 通知编译器产生80186和80286指令

`-ml` 指明用大模式编译

`-vi` 通知编译器在目标文件中产生符号调试信息

`-f287` 通知编译器产生80287浮点指令

`-N-` 禁止编译器产生堆栈检查指令

`-G` 通知编译器执行速度优化

`-DTURBOC -DPROTOTYPES -DRMADS`

定义符号“TURBOC”、“PROTOTYPES”和“RMADS”

`-c` 通知编译器只编译不连接

hello.c 指明源程序文件名

连接器的开关意义如下:

`/x` 通知连接器不产生映象文件

`/d` 通知连接器在符号重复时产生警告信息

`/c` 通知连接器以大小写敏感的方式处理符号

`/Tdc` 通知连接器产生一个DOS下可执行文件格式

`/v` 通知连接器产生包含符号调试信息的可执行程序

c01.obj 指明应该使用的起动代码模块

hello.obj 指明要连接的目标文件名

hello.exe 将可执行文件取名为hello.exe

thcads3.lib FP87 mathl cl

通知连接器将这些库与应用程序连接

1.4 ADS应用程序的加载与执行

AutoCAD R11只支持保护模式的ADS程序，保护模式应用程序的扩展名是“exp”；在AutoCAD R12中又增加了对实模式应用程序的支持，实模式应用程序的扩展名是“exe”。

1.4.1 在图形编辑器中加载ADS应用程序

在图形编辑器中加载一个已编译好的ADS应用程序和加载一个AutoLISP应用程序十分类似：加载AutoLISP应用程序用(Load)函数，而加载ADS应用程序用(xload)函数；同样，还需要一个文件名，即(xload"文件名")。(xload)函数将查找文件名指定的文件，把它加载到内存中，并立即执行其余的初始化操作。假如现在我们要在图形编辑器中加载ADS应用程序hello.exp，应该按以下的形式进行：

```
Command:(xload "hello")
```

由于在AutoCAD R11中只存在以“exp”为扩展名的ADS应用程序，所以(xload)函数会自动为程序加上正确的扩展名。如果在AutoCAD R12版中，AutoCAD会首先寻找hello.exp，如果找到就将其加载；如果未找到，AutoCAD才会寻找hello.exe，并在找到后将其加载。所以，在AutoCAD R12中，加载应用程序最好带上扩展名。如果找到了应用程序，并成功地装入了，那么(xload)函数将把在函数调用时使用的名字作为字符串返回，在上述例子中将返回字符串“hello”；如果不能成功加载，AutoCAD将显示出错信息。

AutoCAD是按照以下次序寻找ADS应用程序的：

- (1)当前目录
- (2)包含当前图形文件的目录
- (3)由环境变量ACAD指定的目录
- (4)包含AutoCAD程序文件的目录

当然，如果你提供了一个全路径(xload)函数，例如(xload"d:\\working\\hello")，AutoCAD就不会再去查寻其它目录了。

1.4.2 在AutoCAD初始化时加载ADS应用程序

在AutoCAD初始化时，它会检查当前是否存在一个名为acad.ads的文件，如果存在，它会自动加载acad.ads中列出的每个ADS应用程序。acad.ads是一个简单的文本文件，它

必须包含一个或多个ADS应用程序名，这些程序名可以带或不带扩展名，每个程序名占文本的一行。

1.4.3 调用ADS外部函数

每个ADS应用程序都可能定义了一组对AutoLISP来说是外部函数的函数。一旦一个ADS应用程序已用(xload)函数加载，你就可以调用该程序所定义的那些外部函数，方法和调用内部函数或用户定义的AutoLISP函数一样，即输入一个包含外部函数名的AutoLISP表达式。AutoLISP的变量可以作为外部函数的自变量传递。同别的函数一样，外部函数将返回一个结果。

外部函数可以提示用户从键盘上或用指定设备拾取点的方法输入数据。外部函数可被AutoLISP函数调用，但ADS应用程序却不能调用AutoLISP函数。使用象AutoLISP那样的加上“C:”的方法，外部函数也可以定义一个新的AutoCAD命令。此时，在“Command:”提示符下，直接键入函数名(无需带括号)，就可调用该函数。应该注意的是，在重名的情况下，后定义的外部函数将替换掉原来的函数。如果两个ADS应用程序定义了同名的外部函数，那么先加载进来的就会丢失。

1.4.4 检查当前已加载的ADS应用程序

在AutoCAD的“Command:”提示符下调用AutoLISP函数(ads)，将得到当前已加载的全部ADS应用程序的信息。

1.4.5 ADS应用程序的卸载

如果内存中加载了过多的ADS应用程序，将会影响系统的运行速度，此时应该从内存中卸载一些暂时不用的ADS应用程序。卸载ADS应用程序用(xunload)函数。例如，要卸载前面装入的hello.exp程序，应按以下形式进行：

```
Command: (xunload"hello")
```

在(且仅在)下面两种情况下，AutoLISP会自动地卸载一个ADS应用程序：

- (1)该应用程序本身报告一个致命错误。
- (2)在退出AutoCAD时。

第二章 ADS 编程基础

2.1 编写第一个ADS程序

在经典著作《C语言程序设计》(Prentice Hall, 1988年第二版)中, Brian Kernighan和 Denis Ritchie开始使用一个现在已经很有名的“Hello, world!”程序来讨论C:

```
#include<stdio.h>
main()
{
    printf("Hello,world!\n");
}
```

在本章一开始,我们将给出用ADS编写的类似的程序.我们称这个程序为HelloADS,它在AutoCAD的“Command:”提示符下,打印出“Hello, ADS!”.

2.1.1 HelloADS.c源程序

```
/*-----HelloADS-----*/

#include <stdio.h>
#include <string.h>
#include "adslib.h"

#define ELEMENTS(array) (sizeof(array)/sizeof((array)[0]))

static int loadfunc();
static int dofunc();
int hello();

struct {
    char *fname;
    int (*func)();
} functab[]={
    {"hello",hello},
    /*{.....,.....},
    {.....,.....},
    other external functions */
};

void main(int argc,char *argv[])
{
    int stat;
    short scode=RSRSLT;
```