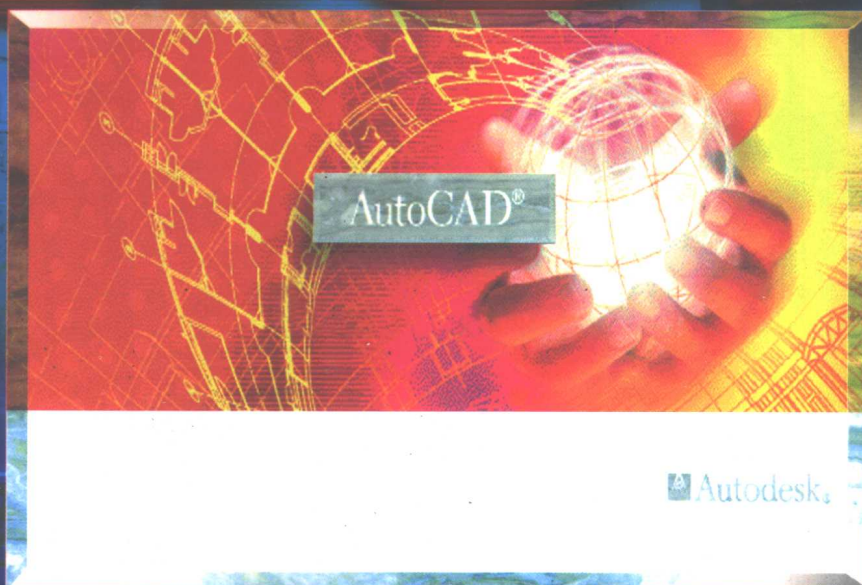


计算机技术入门提高精通系列丛书

AutoLISP

实用教程

梁雪春 崔洪斌 编著
吴义忠 曹 康
康博创作室 审校



人民邮电出版社
PEOPLE'S POSTS &
TELECOMMUNICATIONS
PUBLISHING HOUSE

计算机技术入门提高精通系列丛书

AutoLISP 实用教程

梁雪春 崔洪斌 编著
吴义忠 曹 康

康博创作室 审校

人民邮电出版社

J5500/14

计算机技术入门提高精通系列丛书

AutoLISP 实用教程

◆ 编 著 梁雪春 崔洪斌 吴义忠 曹 康

审 校 康博创作室

责任编辑 顾翀

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号

北京密云春雷印刷厂印刷

新华书店总店北京发行所经销

◆ 开本:787×1092 1/16

印张:20.25

字数:504千字

1998年9月第1版

印数:1—8 000册

1998年9月北京第1次印刷

ISBN 7-115-07272-8/TP·773

定价:28.00元

本书是介绍 AutoCAD R13、R14(for DOS, Win3. x, Win95)的二次开发工具 AutoLISP 的特点和使用方法的实用性教程。全书内容精练,大量的典型实例,包含了编者多年的开发经验和成果。

本书共分十三章,不但介绍了 AutoLISP 的基础知识和语言特点,还对各类函数进行了剖析,并阐述了广大用户关心的参数化绘图、对话框设计及三维实体造型程序设计的方法,最后总结性地给出了一个综合开发的应用实例。

本书不但适合广大 AutoLISP 初学者阅读,对于 CAD 专业开发人员也有很大帮助。

前 言

AutoCAD 是当今世界上较为流行、使用很广的计算机辅助设计和图形处理软件,由此产生的二次开发软件也是层出不穷。AutoLISP 是开发 AutoCAD 重要的工具之一,随着 AutoCAD 版本的更新,AutoLISP 的功能也得到了增强。本书是按照目前市场上较新版本 AutoCAD (R13、R14)的 AutoLISP 编写的。

本书第一章概括地介绍 AutoLISP 语言特点;第二章至第九章详细地分类介绍 AutoLISP 函数;第十章重点介绍参数化绘图——鼠标拖动和尺寸驱动;第十一章阐述如何利用 AutoLISP 进行对话框设计;第十二章介绍立体图的 AutoLISP 程序设计;第十三章给出一个综合实例,说明二维零件图的自动装配。

本书的最大特点是实用性强,在每章后面都有应用实例。这些例子贴切而且实用,都是读者关心的问题,帮助读者在学习 AutoLISP 语言的同时掌握一些开发技巧。其中的一些例子对 CAD 专业开发者也有很高的参考价值。

最后,感谢谢岭南同志对本书的编写给予的大力支持,还要感谢东南大学电光源研究中心和南京电大的老师们对本书编写的帮助。另外,本书从某种意义上讲是康博创作室集体合作的结晶,除封面上署名的作者以外,参加本书内容收集、编写、整理、校对、录入、排版等工作的还有冯志强、张强、刘芝泉、游国权、常征、周秀会、陈君、李伟、李娟、王睿、蔡汇锦、曹锋、刘亚、陈红、李全玉、王宝柱、于朝晖、赵文等同志。

由于作者水平有限,书中难免存在不足之处,敬请读者多提宝贵意见。

编者

1998 年 4 月

●第一章 AutoLISP 语言概述	1
1.1 AutoLISP 语言简介	1
1.1.1 开发 AutoCAD 的重要工具	1
1.1.2 AutoLISP 的特点	2
1.2 AutoLISP 数据类型	2
1.2.1 原子	2
1.2.2 表和点对	3
1.3 AutoLISP 的程序结构	5
1.4 AutoLISP 的运行环境	6
1.5 AutoLISP 的内存分配	6
1.6 AutoLISP 程序的执行过程	7
1.7 AutoLISP R13 和 R14 的特点	8
1.7.1 AutoLISP R13 新特点	8
1.7.2 AutoLISP R14 新特点	9
●第二章 数值函数	11
2.1 计算函数	11
2.2 布尔运算函数	15
2.3 三角函数	16
2.4 综合举例	17
●第三章 表处理函数	19
3.1 提取表中数据的函数	19
3.2 构造和修改表的函数	20
3.3 提取并修改表中数据的函数	21
3.4 表的循环处理函数	22
3.5 其他表处理函数	24
3.6 综合举例	24
●第四章 符号和函数处理函数	27
4.1 符号处理函数	27
4.1.1 赋值函数	27
4.1.2 禁止求值函数与再求值函数	28
4.1.3 其他符号处理函数	29

4.2 函数处理函数.....	31
●第五章 AutoCAD 函数.....	35
5.1 GET 类输入函数.....	35
5.2 图形处理函数.....	38
5.3 command 函数.....	40
5.4 坐标系转换及单位变换函数.....	43
5.5 查询和设置系统状态的函数.....	45
5.6 显示控制函数.....	47
5.7 综合举例.....	50
●第六章 字符串处理函数.....	53
6.1 字符与 ASCII 码互换函数.....	53
6.2 数串互换函数.....	53
6.3 字符串处理函数.....	56
6.4 尺寸标注方式.....	58
●第七章 相等和条件判断函数.....	61
7.1 相等验证函数.....	61
7.1.1 关系运算函数.....	61
7.1.2 逻辑运算函数.....	61
7.2 条件函数.....	62
7.3 循环函数.....	64
7.4 综合举例.....	66
●第八章 文件处理函数.....	73
8.1 搜索、获得文件的函数.....	73
8.2 打开、关闭文件的函数.....	74
8.3 用于文件的输入输出函数.....	75
8.4 综合举例.....	77
●第九章 实体和设备访问函数.....	79
9.1 基本概念.....	79
9.2 选择集操作函数.....	80
9.3 实体名操作函数.....	82
9.4 实体数据函数.....	85
9.5 对符号表的访问.....	93
9.6 对图形屏幕和输入设备的访问.....	95
9.7 综合举例.....	96
9.7.1 实体名和选择集在开发机械 CAD 中的实际应用.....	96
9.7.2 生成局部放大视图的简便方法.....	97
9.7.3 求圆或圆弧中心线.....	101
●第十章 参数化绘图.....	105
10.1 鼠标拖动.....	105
10.1.1 鼠标拖动轴类零件.....	105

10.1.2 推广的鼠标拖动程序(Y1.LSP)	111
10.2 尺寸驱动.....	125
10.2.1 尺寸驱动原理.....	125
10.2.2 复杂零件图的尺寸驱动.....	129
●第十一章 AutoLISP 对话框设计技术	155
11.1 对话框设计过程.....	156
11.2 用 DCL 定义对话框	157
11.2.1 控件及其属性.....	157
11.2.2 对话控制语言(DCL)语法简介.....	165
11.2.3 定义对话框.....	166
11.3 对话框 PDB 函数	180
11.3.1 对话框打开和关闭函数.....	180
11.3.2 控件和属性处理函数.....	182
11.3.3 列表框和弹出式列表框处理函数.....	186
11.3.4 图像控件处理函数.....	189
11.3.5 特定应用数据处理函数.....	191
11.3.6 对话框函数小结.....	191
11.4 对话框的 AutoLISP 驱动	192
11.4.1 驱动程序的基本流程.....	192
11.4.2 对话框驱动程序设计技术.....	193
11.4.3 对话框控件驱动技术.....	198
11.5 综合举例.....	199
●第十二章 AutoLISP 三维立体绘图程序设计	209
12.1 AutoCAD 三维作图命令简介	209
12.1.1 基本体素作图命令.....	209
12.1.2 二维实体扩展命令.....	211
12.1.3 辅助操作命令.....	212
12.1.4 并、交、差命令.....	214
12.1.5 其他相关命令.....	215
12.1.6 利用命令交互式作图举例.....	215
12.2 AutoLISP 在立体绘图中的应用	221
12.2.1 AutoCAD R13 下 AutoLISP 三维程序开发的特点	221
12.2.2 立体图参数化的意义.....	222
12.2.3 三维绘图参数化.....	222
12.3 加工思想在立体绘图中的应用.....	231
12.3.1 简述与思路.....	231
12.3.2 零件加工过程.....	232
12.3.3 立体图绘制过程.....	232
12.3.4 程序算法分析.....	233
12.3.5 结论和说明.....	234

12.4 实例——复杂立体图的程序编制.....	234
12.4.1 一般算法的绘轮廓曲面程序段.....	234
12.4.2 快速算法程序段.....	235
●第十三章 应用举例——二维零件图的自动装配.....	239
13.1 零件图的预处理.....	239
13.2 求图形的内、外轮廓	239
13.2.1 算法简介.....	239
13.2.2 程序设计.....	240
13.2.3 实现要点.....	241
13.2.4 求内轮廓.....	280
13.3 零件图的裁剪与消隐.....	281
13.3.1 算法简介.....	281
13.3.2 程序设计.....	281
13.3.3 实现要点.....	282
13.4 装配图生成.....	301
13.5 程序运行.....	303
●附表 A COMMAND 函数调用常用 AutoCAD 命令格式	305
●附录 B 标准出错信息	310
B.1 用户程序错误	310
B.2 内部错误	313

AutoLISP 语言概述

1.1 AutoLISP 语言简介

1.1.1 开发 AutoCAD 的重要工具

LISP(List Processing Language)是一种计算机的表处理语言,是在人工智能学科领域广泛应用的一种程序设计语言。AutoLISP 语言是嵌套于 AutoCAD 内部,将 LISP 语言和 AutoCAD 有机结合的产物。使用 AutoLISP 可直接调用几乎全部 AutoCAD 命令。AutoLISP 语言既具备一般高级语言的基本结构和功能,又具有一般高级语言所没有的强大的图形处理功能,是当今世界上 CAD 软件中被广泛采用的语言之一。

美国 AutoDesk 公司在 AutoCAD 内部嵌入 AutoLISP 的目的是使用户充分利用 AutoCAD 进行二次开发:实现直接增加和修改 AutoCAD 命令,随意扩大图形编辑功能,建立图形库和数据库并对当前图形进行直接访问和修改,开发 CAD 软件包等。

AutoLISP 语言最典型的应用之一是实现参数化绘图程序设计,包括尺寸驱动程序,鼠标拖动程序等。尺寸驱动是指通过改变实体标注的尺寸值来实现图形的自动修改;鼠标拖动即利用 AutoLISP 语言提供的(GRREAD[<track>])函数,让用户直接读取 AutoCAD 的输入设备(如鼠标),任选项追踪光标移动存在且为真时,通过鼠标移动光标,调整所需的参数值而达到自动改变屏幕图形大小和形状(见本书 10.1 节)。

到目前为止,大多数参数化程序都是针对二维平面图编制的。实际上,立体图同样可以实现参数化绘图,在 AutoCAD 13 中编制实体的立体图参数化程序比其平面三视图程序更简单,而且立体图生成后,可以很方便地生成三视图、剖面图和轴侧图等。

另一个 AutoLISP 的典型应用就是驱动利用 AutoCAD 提供的 PDB 模块构成 DCL(Dialog Control Language)文件创建自己的对话框。

自从 AutoCAD 嵌入 AutoLISP 以后,使仅仅作为交互式图形编辑软件的 AutoCAD 变成能真正进行计算机辅助设计、绘图的 CAD 软

件。由于 LISP 灵活多变,又易于学习和使用,因而使 AutoCAD 成为功能很强的工具性软件。

1.1.2 AutoLISP 的特点

AutoLISP 具有如下特点:

(1) AutoLISP 语言是在普通 LISP 语言基础上,扩充了许多适用于 CAD 应用的特殊功能而形成的,一种仅能以解释方式运行于 AutoCAD 内部的解释性程序设计语言。

(2) AutoLISP 语言中的一切成分都是以函数的形式给出的,它没有语句概念或其他语法结构。执行 AutoLISP 程序就是执行一些函数,再调用其他函数。

(3) AutoLISP 把数据和程序统一表达为表结构,即 S—表达式,故可把程序当作数据来处理,也可把数据当作程序来执行。

(4) AutoLISP 语言中的程序运行过程就是对函数求值的过程,是在对函数求值的过程中实现函数的功能。

(5) AutoLISP 语言的主要控制结构是采用递归方式。递归方式的使用,使得程序设计简单易懂。

1.2 AutoLISP 数据类型

AutoLISP 语言主要用到如下数据类型:

- 符号 (SYM)
- 字符串 (STR)
- 表(及用户定义的函数) (LIST)
- 文件描述符 (FILE)
- AutoLISP 的内部函数 (SUBR)
- AutoCAD 的选择集 (PICKSET)
- AutoCAD 的实体名 (ENAME)
- 函数分页表 (PAGETB)

本节只介绍前五种数据类型,其他类型将在相应的章节中介绍。

在这五种数据类型中,前四种称为原子(ATOM),原子中包括数字原子(整型数和实型数)、符号原子和串原子。

所以 AutoLISP 最基本的数据类型是原子和表,它们又总称为符号表达式(Symbolic-Expression),也称为 S—表达式。

1.2.1 原子

1. 整数

整数是由 0、1、2……9、+、- 等字符组成,整数的大小与所使用的计算机系统有关。现在常用机型,如 PC486、PC586,一般可在 -2147483648 到 +2147483647 之间,如整数大小超出此范围,计算机将提示出错信息。

2. 实型数

实型数用双精度浮点数表示,并且至少有 14 位的精度,即整数后跟小数。如果实数的绝对

• 2 •

值小于 1, 小数点前必须加 0, 不能直接以小数点开头, 否则被误认为点对而出错。实型数的范围比整型数大得多, 如对于 16 位微机, 实型数范围约为 $-1.797693 \times 10^{308}$ 到 1.79793×10^{308} 。它不易超界, 故用户可以尽量采用实型数。

实型数也可采用科学记数法表示, 如 0.12×10^9 可表示为 0.12E9。

3. 符号

(1) 符号(symbol)包括除“)”、“(”、“.”、“,”、“|”、“;”之外的任何打印字符。

(2) 符号原子的长度没有限制, 但尽量不要超过 6 个, 否则要占用额外的内存, 降低运行速度。

(3) 在 AutoLISP 中符号的大小写是等效的, 如以下的符号原子都是合法的。

A A12 PC x-38-6 *A

(4) AutoLISP 中的任何符号都是有值的, 即符号都要赋以一定的数值, 或者说符号总是约束在一定值上。一般用赋值函数 setq 进行赋值。

例如:

(setq x 25.0)

意思是将 25.0 赋给 x, 这时 x 的当前约束值即为 25.0。一个符号在使用前如没有赋以任何值, 则该符号的值为 nil(空), 它不占用内存空间。

注意:

为区别起见, 我们常用术语“符号”来指存储静态数据的一个符号名, 例如内建式函数和用户定义函数名是一个符号。

用术语“变量”来指存储程序数据的符号名, 如上述 (setq x 25.0) 中的变量名为 x, 它的值为 25.0。AutoLISP 程序中每一个变量都要消耗少量内存, 故当变量值不再有用时, 重复使用变量名或将变量值设置成 nil 是良好的程序设计习惯。符号名或变量名不能包含空格字符或分隔符, 并总是以字母开头。其值保持不变的变量称为常量, AutoLISP 仅用一个常量 Pi。Pi 的值约为 3.14159。

4. 字符串

字符串是由包含在一对双引号内的一组字符组成的, 如:

“ABC” “135” “Ab C” “ ”

字符串可以包括任何可打印的字符。字符串中字母的大小写及空格都是有意义的。字符串的最大长度为 100 个字符, 位于 100 之后的字符无效。若字符串中没有任何字符, 则为空串“ ”。

1.2.2 表和点对

1. 表

在 AutoLISP 语言中, 表有如下特点:

(1) 表是指放在一对相匹配的左、右括号中的一个或多个元素的有序集合。

(2) 表中的每一个元素可以是任何类型的 S-表达式, 即可以是数字、符号、字符串, 也可以是表。

(3) 元素与元素之间要用空格隔开, 而元素与括弧之间可不用空格, 因为括弧本身就是有

效的分隔号。

例如：

```
(15 (a b) c d)
```

在此例中,表内有 4 个元素,即 15、(a b)、c 和 d,其中第二个元素又是一个表。

(4) 表是可以任意嵌套的,上例表中即嵌套了一个表(a b)。表可以嵌套很多层,从外层向里依次称为 0 层(也称顶层)、1 层、2 层……我们所指的表中的元素是指表的顶层元素。

(5) 表中的元素是有顺序的,为便于对表中元素进行存取,每个元素都有一个序号。从左向右,第一个元素的序号为 0,第二个元素的序号为 1,第 i 个元素序号为 i-1。

(6) 表的大小为表的长度,即表中顶层元素的个数。没有任何元素的表称为空表。空表用 () 或 nil 表示。在 AutoLISP 语言中,nil 是一个特殊的符号原子,它既是原子又是表。

(7) 表有两种基本类型:标准表和引用表。

- 标准表:标准表是 AutoLISP 程序的基本结构形式,AutoLISP 程序就是由标准表组成的。标准表是用于函数的调用,其中第一个元素必须是系统内部函数或用户定义的函数,其他的元素为该函数的参数,如上面提到的赋值函数的调用,即采用标准表的形式。

```
(setq x 25.0)
```

表中第一个元素 setq 为系统内部定义的赋值函数,x 和 25.0 均为 setq 的参数。

- 引用表:这种表第一个元素不是函数,即不作为函数调用,常作为数据处理。

引用表的一个重要应用是表示图中的点的坐标。当表示点的坐标时,表中的元素是用实型数构成的。

表示二维点的坐标是用两个实型数构成的表,如(20.0 30.5),其中第一个元素表示点的 x 轴坐标,第二个元素表示点的 y 轴坐标。三维点的坐标表示,是用三个实型数构成的表,如(20.0 82.5 1.0),其中三个元素依次表示点的 x 轴坐标、y 轴坐标和 z 轴坐标。

2. 点对(dotted pair)

点对也是一种表,该表中有两个元素,两元素中间为一圆点“·”,且圆点与元素之间必须用空格分开。

例如：

```
(A · B)
```

就是一个点对,A、B 与圆点均用空格分开,其中第一个元素 A 为该点对的左元素,第二个元素 B 为点对的右元素。点对亦可任意嵌套。当使用点对时,切记要注意它的书写格式。

例如：

```
(X · (B · (Y · Z)))
```

为合法点对,而

```
(X · (B · Y) · Z)
```

即为非法的。

点对常用于构造连接表。

1.3 AutoLISP 的程序结构

AutoLISP 语言没有“语句”这一术语, AutoLISP 程序一般是由一个或一系列按顺序排列的标准表所组成。例如:

```
(setq x 25.0)
```

是上面提到的标准表,又可以看作是一个 AutoLISP 程序。

又如:文件名为 pq.lsp 的 AutoLISP 文件是由以下程序组成的:

```
(setq x 25.0)
(setq y 12.2)
(+ (* x y) x) ;表示 xy+x
```

以上是由三个标准表组成的程序,每个标准表的第一个元素(如 setq, +, *)均为系统提供的函数,称为系统的内部函数。setq 为赋值函数, + 为加函数, * 为乘积函数。标准表中的其他元素为相应函数的参数。这个程序是将 25.0 赋给变量 x, 将 12.2 赋给变量 y, 求变量 x 和 y 的值的乘积, 再求此乘积与 x 的总和。

AutoLISP 程序的书写格式有如下特点:

(1) 由于 AutoLISP 语言的一切成份都是函数, 而所有函数又以表结构形式存在, 所以 AutoLISP 程序的所有括号都需要左右匹配。

(2) AutoLISP 程序阅读函数时, 按从左到右的规则进行。

(3) 函数必须放在表中第一个元素的位置, 如赋值函数 setq、算术运算函数 +、* 等应为表中第一个元素, 即放在操作数之前, 而不是放在它们的中间, 这与算术运算的书写格式不同, 初学者可能会感到不习惯。表中的函数与参数, 各参数之间均至少需要一个空格来分开。

(4) 两个表之间和表内的多余空格和回车是不需要的, 故一个表可占多行, 一行也可写多个表, 如:

pq.lsp 程序可写成如下形式:

```
(setq x 25.0) (setq y 12.2) (+ (* x y) x) ↵
```

(5) AutoLISP 程序中可以使用分号; 作注释。注释的作用是对程序作解释。AutoLISP 求值器总是忽略每一行中分号以后的部分。注释可放在程序中的任何地方。

(6) AutoLISP 程序一般是以扩展名为“.LSP”的 ASCII 码文本文件的形式表达。

AutoLISP 程序就是对一个个 AutoLISP 函数的调用。函数是 AutoLISP 语言处理数据的工具, 学习掌握 AutoLISP 语言, 核心就是要掌握 AutoLISP 函数。AutoLISP 函数分为系统内部函数和用户定义的外部函数。AutoLISP 提供了大量的系统内部函数, 以满足编程的需要。

AutoLISP 对函数的调用是通过标准表来实现的。如前所述, AutoLISP 程序的基本结构就是由一系列标准表有序构成的。AutoLISP 程序的运行, 就是对标准表依次进行求值。标准表或者说函数调用的一般格式如下:

```
(函数名 [<参数 1>] [<参数 2>] ..... [<参数 n>])
```

标准表中的第一个元素必须是函数名, 以后的各元素为该函数的参数, 参数的类型及数目

取决于函数。为了便于读者更好地了解和掌握函数调用的格式,本书对所用的符号特做如下的约定。

- <.....> 尖括号中的符号表示函数所要求的参数的类型,它是必须存在的。
- [.....] 方括弧的内容是任选项,它可以存在,也可省略,但该项只能出现一次。
- 省略号,表示省略号前面同样的参数,其数目不限。

学习 AutoLISP 的系统内部函数时,必须掌握以下的基本内容:

- ① 函数调用格式:即函数名、函数要求的参数个数和类型。
- ② 函数的功能:即该函数的功用和作用,它对其参数如何处理。
- ③ 函数的求值情况:即哪些参数要求值,哪些不被求值。
- ④ 函数求值结果的返回值类型:这点很重要,因为大多数函数的返回值都要被其他函数接受,而每个函数所需要的参数都有特定的类型。因此只有搞清被调用函数的返回值的类型,才不会因用错函数的参数而出错。在本书以后各章中将分别介绍 AutoLISP 系统内部函数。

1.4 AutoLISP 的运行环境

AutoLISP 对运行环境有如下要求:

1. 对处理器要求

- (1) 至少应为 386 型计算机,计算机应配有协处理器,即 CPU386+387。
- (2) 最好 CPU 为 486 或 586。

2. 对存储器要求

- (1) 内存: AutoLISP R2.6~R10,有 1M 以上内存即可。
AutoLISP R11 以上版本,有 4M 以上内存即可,最好 8M 内存。
- (2) 硬盘: AutoLISP R2.6~R10,有 10M 以上空闲硬盘空间即可。
AutoLISP R11 以上版本,至少要 35M 空闲硬盘空间。

3. 对输入、输出设备的要求

- (1) 应配有鼠标器或数字化仪,以用于输入。
- (2) 应配有打印机或绘图仪,以用于图形的输出。
- (3) 彩色显示器,分辨率在 640×400 以上。
- (4) 应配有高密软驱,以用于软件的安装。

1.5 AutoLISP 的内存分配

AutoLISP 的变量(整型、实型、字符串等)、用户定义的函数以及标准函数,只是在 AutoLISP 的编辑过程中,存储在计算机的内存中。当运行 AutoLISP 程序时,它需要两个很大的内存区域。

① heap(堆区域),它存储所有的函数和变量。因此程序使用的函数和变量越多,或函数越复杂,则“heap”空间占得也就越多。

② stack(栈区域),它存储函数的变量和局部结果。因此“嵌套”的函数越深,或函数执行的

递归的次数越多,那么所用的栈空间也越多。

AutoLISP 隐含的堆和栈空间大小为:

Heap = 25000 字节

Stack = 20000 字节

在 AutoCAD 下运行 AutoLISP 不能扩展它的堆或栈空间。如果用户定义的函数和变量太多,以致用光了所有的堆空间,AutoLISP 将显示下列信息:

Insufficient node space

并且中止当前函数的执行。如果在执行 AutoCAD 时,没有足够的内存装入 AutoLISP,则显示下列信息:

Insufficient memory-AutoLISP disabled.

直到有足够的内存后 AutoCAD 重新启动时,AutoLISP 才能恢复其功能。

可以在 DOS 状态下用“SET”命令来调整 AutoLISP 所需要的堆栈区域占有的内存量。例如:

```
C> SET Lispheap = 25000(字节)
```

```
C> SET Lispstack = 15000(字节)
```

告诉 AutoLISP 为堆保留 25000 字节的内存,为栈保留 15000 字节的内存。

注意:

堆区域和栈区域这两个区域的总和不能超过 45000 字节。

1.6 AutoLISP 程序的执行过程

对于很短的 AutoLISP 程序,只由一至两个表所组成,如简单的数值函数的运算,或用 defun 函数定义的简单用户函数(见第四章),可直接在 AutoCAD 环境中的 Command:提示符下输入即可,返回结果立即显示在文本屏幕上。

对于一般的 AutoLISP 应用程序,需采用文本编辑器进行编辑。编辑器可以随意选取(如 DOS 下采用 Edit, Windows 下可用 NotePad),当然最好使用那些能够检查相匹配的括号的编辑器(如 Visual C++, Borland C++ 等),因为在 AutoLISP 程序中一对匹配括号是组成表的单元,有时很容易搞乱,影响程序的执行。

在编辑器下编辑好的 LSP 程序(如下面的 tl. LSP),当回到 AutoCAD 环境下,用 Load 函数装载后便可以执行了。执行时,如果程序中没有 defun 函数,系统便边装入边运行;若有 defun 定义的命令或函数,装载后只需再在 Command:提示符下键入 defun 函数定义的命令名或函数名即可运行相应的命令或函数。

例如:在 NotePad 下编辑下列 LISP 程序:

```
(Print "\n Please Use command:TESTLISP\n")
(defun C:TESTLISP()
  (alert "AutoLISP Test Function!")
)
```

)

将上述程序保存为 C:\ACADR13\tl.lsp。

启动 AutoCAD, 在 Command: 提示符下键入

```
(load "C:\\ACADR13\\tl.lsp") ↵
```

或者

```
(load "C:/ACADR13/tl") ↵
```

按回车后, 即执行第一句并返回最后一个 defun 函数定义的函数名:

```
Please Use Command: TESTLISP  
C:TESTLISP
```

装载完后再运行 defun 定义的函数, 在 Command 提示符下键入: (C:TESTLISP) 或 TESTLISP ↵ 即执行该程序段, 屏幕上就出现警告对话框, 如图 1.1 所示。

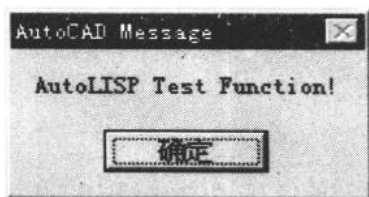


图 1.1

注意:

① 在 AutoCAD 工作目录下有一个 ACAD.LSP 文件, 它是当 AutoCAD 启动、新建文件 (New)、打开文件 (Open) 时自动装载的 AutoLISP 程序。用户可以修改它, 实现一定的目的。例如, 用户想要在 AutoCAD 启动时自动装入自己定义的函数或程序, 则可以在 ACAD.LSP 程序中加入 (defun xxx()) 程序段或 (load "xxx") 函数。

② 在装入 ACAD.LSP 文件时若出现一条 AutoLISP 错误, 剩余的文件就会被忽略而不装入, 并提示出错。如果一个 Load 函数的调用是成功的, 它就返回被加载的文件中最后的那个表达式的值。

1.7 AutoLISP R13 和 R14 的特点

新版本的 AutoCAD 基本上保持 AutoLISP 接口不变。一般情况下都兼容以前版本, 即在低版本 AutoCAD 环境下能够运行的 AutoLISP 程序, 同样能够在高版本环境下正确运行。

1.7.1 AutoLISP R13 新特点

AutoLISP R13 比过去的 R12 功能有所增强, 新增了 53 个函数。由于 AutoLISP R13 与以前版本在三维实体类型上采用了不同的技术, 使得 AutoLISP 在 R13 下不再能使用原来 AME 模块定义的三维图处理函数。因此在 R13 下开发三维实体程序与在以前版本下开发大不相同, 而且比以前更方便、更快速。