电子计算机软件

IBM——PC机

宏LISP语言
人工智能程序设计

李卫华　蒋冠珞编著

湖南科学技术出版社

电子计算机软件

IBM——PC 机

宏LISP语言
人工智能程序设计

李卫华　蒋冠珞编著

湖南科学技术出版社

电子计算机软件IBM——PC机
宏LISP语言人工智能程序设计
李卫华　蒋冠珞　编著
责任编辑：周翰宗

*

湖南科学技术出版社出版
（长沙市展览馆路8号）
湖南省新华书店发行　湖南省新华印刷二厂印刷

*

1986年1月第1版　1987年7月第2次印刷
开本：787×1092毫米 1/16　印张：17.75　字数：445,000
印数：6,001——12,100
统一书号：15204·161　定价：4.30元
湘科 87—5

内 容 简 介

IBM-PC 机是国内外广泛使用的微型计算机。人工智能则是计算机科学中最有希望的学科之一，受到全世界普遍重视，前景极为诱人。由于它具有许多潜在的可应用性，对于这一工作的探索与开拓，正方兴未艾，吸引着众多计算工作者的浓厚兴趣。本书介绍的宏LISP 语言是一种能够在 IBM-PC 机上运行、并实现了中英文兼容的 人工智 能设计程序语言。她版本新，用途广，简单易学，适于普及。她可用于编写有关推理、规划、设计、思维和学习等智能程序，是人们进行人工智能研究、开发工作的重要工具。

书中详细介绍了宏LISP语言的全部功能及运行环境，举例说明了用宏 LISP 语言编写、修改、动态调试程序的技巧，具体展示了用宏LISP语言实现PROLOG语言以及进行人工智能程序设计的方法，资料新颖，内容丰富，被许多高等院校遴选为人工智能程序设计课程的教科书。由于列有难易兼顾的程序、例示数百个，很适于广大微型计算机工作者自学，也可供从事人工智能研究、开发工作的科技工作者参考。

目　录

前　言

人工智能是一门有研究前途和实用价值的新兴学科。它通过赋予机器推理、判断以及学习的能力，使电脑能模仿人脑从事推理、规划、设计、思考、学习等智力活动，解决迄今需由人类专家才能处理好的事情。如：博弈、定理证明、自动程序设计、通用问题求解、自然语言理解、自然语言生成、模式识别、物景分析、医疗诊断、化学分析、矿床勘探、工程设计、军事决策、符号数学、艺术创作、智能机器人以及办公室自动化等。人工智能的最终目标就是构造用来求解上述难题的计算机程序。从人工智能这二十多年的发展历程来看，不论是早期的人工智能程序还是新近出现的著名的人工智能系统，多数是用LISP语言书写的。无论是以用LISP语言书写的代码行数来看，还是以它对研制其它语言的影响来看，LISP是目前人工智能语言族中最重要的成员。美国著名的Jear Sammett教授甚至宣称：程序设计语言可分两类，一类是LISP，一类是所有其它程序设计语言。可见，LISP语言对计算机科学的教学、科研与应用有着十分重要的作用。

LISP语言是一种表处理语言。它由举世闻名的计算机科学家、图灵奖获得者、现美国斯坦福大学John McCarthy教授发明于1960年。之后，不少新版LISP相继问世，其中最有代表性的是宏LISP语言。本书将向读者介绍的IBM-PC机宏LISP语言就是在基本LISP语言基础上增加许多易于构造大程序、便于作结构程序设计的新特征、并可在国内拥有几十万台的IBM-PC机及其兼容机上运行的新LISP语言版本。

诚然，从理论上说，可用任一语言书写任一程序。但大量证据表明，所用的语言不同，构造人工智能系统的难易程度也大不相同。用来构造人工智能系统的程序设计语言通常应具备下述特征：

要有各种数据类型，以利于描述由大系统需要的多种信息。在IBM-PC机宏LISP语言中，数据类型有T、NIL、小整数、大整数、短浮点数、长浮点数、2至16进制数、串、表、变量、数组、指针、文件、堆栈、记录、入口对象和可执行程序块等；

要有把系统分解成若干易于理解的小单位的能力，从而既能较为容易地改变系统的某一部分，又不破坏整个系统。为此，IBM-PC机宏LISP语言提供了函数定义、包定义功能；

要有灵活的控制结构，以简化系统的递归与并行分解。在IBM-PC机宏LISP语言中，除配有传统语言所具备的一般迭代控制结构外，还配有其它语言很少具备的结构性良好的循环控制结构和递归控制结构；

要有用户与系统可以彼此交互的能力，从而既便于程序研制，又能最有效地使用已完成了的系统。IBM-PC机宏LISP语言提供了相当强的交互式程序设计环境。数据、程序、特性有错，可不离开当时环境作现场诊断、处理。之后从断点开始继续运行。窗口是在其它语言以及一般LISP语言中见不到的重要特征。用窗口能将显示屏幕分成若干物理显示区，把不同程序数据、图形送至不同窗口，并在不同窗口运行不同程序等等；

要有生成有效代码的能力，从而使系统的执行易于接受。在IBM-PC机宏LISP语言中，除用汇编语言编写全部基本函数外，还提供了在LISP程序中调用汇编程序的机制。此外，借助8087的支撑，可使浮点运算速度增快许多倍；

要有推迟约束时间的能力。在程序执行时，IBM-PC机宏LISP语言能对数据结构的大小、待运算对象的类型等作戏剧性地改变；

要有履行自动推理、存贮作推理基础的断言数据库的机制。借助IBM-PC机宏LISP语言实现的PROLOG语言库具有内部谓词多、自动推理快、存取数据方便等优点。用著称为人工智能之汇编语言的LISP语言作此事特别适于调试、改进日新月异的PROLOG语言；

要有辨别数据、确定控制的模式匹配机制。数据的模式匹配是构造大型知识库的重要部分。控制的模式匹配构成了执行产生式系统的基础。LISP中的匹配函数是作此事的能手，它们已运用于许多人工智能系统中；

要有把过程与说明式数据结构混合起来的能力。IBM-PC机宏LISP语言将程序与数据均表示为表。因此

可把程序作为数据处理，也可将数据当程序来执行。

总之，宏LISP语言配备有很好的编程环境，提供了种类繁多的自动化工具，是一种越来越受到人们高度重视的人工智能程序设计语言。最近，我们解决了宏LISP语言的汉字支持问题。用户既能以英文又能以汉字输入/输出并作程序设计。这特别适于中国计算机用户使用宏LISP语言作人工智能的开发、推广、普及和应用工作。为便于读者尽快掌握和使用该语言，本书除介绍美国IBM-PC机宏LISP语言的全部系统函数以及我们对原系统所作的扩充外，还列有一批我们在教学和上机实践中积累起来的涉及面广、难易兼顾的程序实例。其中几乎用到每一个常用宏LISP函数。想了解何谓人工智能、怎样步入人工智能之门、如何深入研究人工智能、想浏览更多人工智能程序实例的计算机爱好者、决策人、大学生、研究生、科技工作者都可从本书以及参考文献〔1，2〕中获得满意的回答。

宏LISP语言简单易学。若能在一台IBM-PC机旁学宏LISP，几天后便会惊奇地发现：自己不仅学会与掌握了一门计算机语言，而且已步入了富有研究前景的人工智能领域！

在本书编写及程序调试过程中，在解决宏LISP语言的汉字支持问题时，武汉飞帆研究所的方初同志作了大量工作，编者在此深表谢意。编者还要感谢甘永良、卓彦、陈湘义、汤庸、刘建伟、童伟、黄芳栋、程磊、张能胜、黄玲、胡卫宁等同志在程序调试过程中所付出的辛勤劳动。

虽然宏LISP语言是一门普受计算机科学工作者喜爱的人工智能程序设计语言，但由于编者对该语言理解浅薄，用此语言作人工智能程序设计的经验不足，加之编写时间仓促，书中一定存在不少错误，敬请老师、同行和广大读者指正。

李卫华　武汉大学计算机科学系

蒋冠珞　武汉飞帆研究所

一九八五年十月于武汉

第一章　启动与使用

§1·1　启　　动

凡带有DOS1.1或DOS2.0、内存在128KB以上的任一型号的IBM-PC机、长城0520机及其兼容机均可运行IBM-PC机宏LISP语言；内存在64KB至128KB之间的同型机可运行一般LISP语言．具体操作步骤如下：

1．打开显示器开关．

2．将MACRO-LISP盘插入驱动器A．

3．打开主机开关．在输出正确的内存自检信息之后，机器将控制赋予DOS，DOS则要求用户输入新日期和新时间．这时，用户可按月—日—年（如5—10—1985）输入日期，按点:分:秒.百分秒（如10:30:25.30）输入时间；也可简单输入两个回车键默认机器内的当前日期与当前时间．

4．当DOS给出命令提示符A>时，发命令：　LISP↵　启动一般LISP．　发命令：LISP　MACRO↵　或　LISP　CMACRO↵　启动具有强交互式程序设计环境的宏LISP或汉字宏LISP．启动命令发布后，DOS把控制交给LISP，LISP回答提示符*．等光标出现在*之后，置于驱动器A中的MACRO-LISP盘不再需要，故可将之取出、放入用户盘或MACRO-LISP-LIB盘，并运行你的LISP程序．

下面是一般LISP的运行实例，接着是汉字宏LISP的运行实例．为便于注释，笔者增加了行号．例中的黑体字是DOS或LISP的输出，其它是用户输入．

```
1.A>LISP
2.MACRO LISP for IBM PC
3.Version 1.7A May 1985
4.Copyright (C) USa
5.Serial Number 008503

6.* (+ 5 6)
7.11

8.* (LIST 1 "2" (CAR '(3 4)) (CDR 5))
9.(BAD ARGUMENT)

10.1▶ (RETURN 5)
11.(1 "2" 3 5)

12.* (DEF 'FAC '(LAMBDA (N)
```

```
13.       (COND ((= 0 N) 1)
14.             (T (* N (FAC (SUB1 N]
15.(LAMBDA (N) (COND ((= 0 N) 1)  (T (* N (FAC (SUB1 N))))))

16.* (FAC 6)
17.720

18.* (EXIT)

19.A>
```

上述程序运行注释：

1. 在DOS控制下启动一般LISP.
2～5. LISP显示宏LISP语言版本信息.
6. LISP显示提示符，用户敲入待求值表达式.
7. LISP显示运算结果.
8. LISP显示提示符，用户敲入有错表达式.
9. LISP显示对非表5取CDR的出错信息.
10. 错误级别为1，1后面的▶是出错提示符，用户用5作为(CDR 5)的值.
11. LISP显示改后第8行表达式的运行结果.
12～14. LISP显示正常的提示符，用户敲入求n!的自定义函数. 右方括号] 等价于一个或多个右圆括号），即：] 可与若干个（自动匹配. 此处的] 就等价于7个）的作用.
15. LISP显示新定义函数的定义体.
16. LISP显示提示符. 用户敲入运行新函数的表达式.
17. LISP显示新函数的运行结果.
18. LISP显示提示符，用户要求退出LISP、回到DOS.
19. LISP退出. DOS显示命令提示符.

```
1.A>LISP CMACRO
2.汉宇宏LISP语言  V1.7B版  1985年10月
3.武汉飞帆研究所和武汉大学AI室联合开发
4.Serial No. 008507
5.* (DEFUN 和 (LAMBDA (N)
6.       (LET (SUM 0)
7.            (FOR I FROM 1 TO N (INC SUM I))
8.       SUM]
9.OK
10.* (和 10)
11.55
12.* (FLAG '李欢 '好动)
13.好动
14.* (PUT '李欢 3 '年龄)
15.3
```

```
16.* (PUT '李欢 '湖北 '籍贯)
17.湖北
18.* (FLAG '李欢 '男)
19.男
20.* (PROPS '李欢)
21.(男 (籍贯·湖北) (年龄·3) 好动)
22.* (EXIT)
23.A>
```

下面是以上程序运行的注释：

1. 在DOS控制下启动汉字宏LISP.

2～4. LISP显示汉字宏LISP语言版本信息.

5～8. LISP显示提示符，用户敲入求 $\sum_{i=1}^{n}$ 的自定义函数.

9. LISP显示接受新定义函数的信息.

10. LISP显示提示符，用户敲入运行新函数的表达式.

11. LISP显示新函数的运行结果，即 $\sum_{i=1}^{10} i$.

12. LISP显示提示符，用户敲入赋标志给标识符李欢的命令.

13. LISP显示赋给标识符李欢的标志.

14. LISP显示提示符，用户发命令赋特性给标识符李欢.

15. LISP显示赋给标识符李欢的特性值.

16. LISP显示提示符，用户发命令赋特性给标识符李欢.

17. LISP显示特性值.

18. LISP显示提示符，用户发命令赋标志给标识符李欢.

19. LISP回答标志.

20. LISP显示指示符，用户发命令询问标识符李欢的标志和特性.

21. LISP显示标识符李欢的标志和特性.

22. LISP显示提示符，用户发命令退出LISP并回到DOS.

23. 退出LISP，DOS显示命令提示符.

§1·2 键　　盘

当你在显示监视上一见到闪烁的光标时，你应通过键盘输入相应信息给LISP. LISP将根据你所输入的信息作出相应的处理. 现把IBM—PC机键盘分成如图1.1所示的五组：

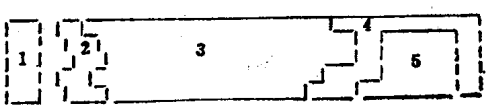

图1.1

(1) 功能键. 功能键F1,…,F10位于键盘最左部.

(2) 数据、控制与方式键。该组键均呈深灰色，位于键盘左部、功能键右部。计有：Esc，Tab（包括|←和→|），Ctrl，Shift和Alt。

(3) 字母、数字与特殊字符键。本组键呈白色，位于键盘中部。计有：26个英文字母，0到9共10个数字，空格以及32个特殊字符键。

(4) 键盘控制键。该组键呈深灰色，位于第5组键周围。具体有：CapsLock，Shift，PrtSc，Return，BackSpace，NumLock，ScrollLock/Break，－，＋，Enter。

(5) 数字键盘与光标控制键。该组键是被第4组键包围的11个白色键。计有：0/Ins，1/End，2/↓，3/PgDn，4/←，5，6/→，7/Home，8/↑，9/PgUp，·/Del。

键盘上五个组内的按键都有连发功能。因而，当持续按下某键时，将产生重复效果。下面是几个由LISP识别的特殊键：

·空格键 第3组中的空格键用来将一空格放在光标当前所在位置。对于LISP来说，空格非常重要。因为在LISP中，参量之间、运算符与运算数之间都是以空格分隔的。

·转换键↑ 第2组和第4组中的Shift键可用来转换第3组中字母键的大小写。若按某字母键后，屏幕上显示该字母的小（大）写形式，则同时按下Shift键和该字母键后，屏幕上便显示该字母的大（小）写形式。此外，若同时按Shift键和第3组中的某非字母键，则屏幕上显示该键上方位置所标明的符号。

·字母锁存键CapsLock LISP区别同一标识符的大写小写，尤其规定基本函数名和库函数名都得用大写。因而，在运行LISP时最好将第4组中的CapsLock按下使字母输入显示成大写形式。基本函数（RAISE 1）也自动将执行RAISE之后输入的小写字母转换成大写字母。再次按下CapsLock或执行（RAISE 0）表示不作字母从小写到大写的转换。

·退格键BackSpace 位于第4组顶行的退格键←或BackSpace用来删去敲入的字符。

·中止键Ctrl—Break 同时按下第2组的Ctrl键和第4组的ScrollLock/Break键使LISP中止正在执行的函数，并将控制交给错误处理程序。

·暂停键Ctrl—NumLock 同时按下第2组的Ctrl键和第4组的NumLock键可暂停打印和任何计算，使计算机处于暂停状态。此暂停状态一直维持到另一非Shift键、非Ctrl键、非Alt键、非NumLock键、非CapsLock键、非Break键、非Ins键按下为止。

·系统复位键Ctrl—Alt—Del 同时按下第2组的Ctrl键和Alt键以及第5组的Del键使LISP立即终止、并重新启动DOS一次。从功能上看，此步相当于断电、接通、启动DOS这三个动作，但此步较那三个动作快得多。

·打印屏幕键Shift—PrtSc 在打印机接通电源后，同时按下第2组（或第4组）的Shift键和第4组的PrtSc键可将屏幕内容打印在打印机上。

·扩充ASCII字符输入键 有些ASCII字符在键盘上找不到，因而要借助第2组的Alt键和第5组中的数字键来输入这些字符。具体地说，即按下Alt键不动、敲入相应欲输字符之ASCII代码的十进值、松放Alt键后，所需字符便显示在屏幕上。基本函数CHRVAL就需用此法输入在键盘上找不到的字符。

·行为键 有些键用来输入数据，有些键用来控制数据的输入，有些键既可输入数据又可控制数据的输入。第3组的键用来输入数据。Shift、CapsLock、NumLock、Alt和Ctrl键本身不输入数据，但影响由其它许多键输入的数据。所输入的数据一个挨一个地存在缓冲区。在按下某一行为键后，LISP才处理存在缓冲区的输入。在按任一行为键之前，退格键BackSpace有效。按下行为键之后，退格键不能再用来消去已输入字符。LISP认为下述键是行

为键：

Enter, Return, F1, …, F10, Shift—F1, …, Shift—F10, Alt—F1, …, Alt—F10, Ctrl—F1, …, Ctrl—F10, Shift—Tab（即|←), Home, ↑, PgUp, ←, →, End, ↓, PgDn, Ins, Del, Alt—A, …, Alt—Z, Alt-0, …, Alt—9, Alt——, Alt—=, Ctrl—Home, Ctrl—PgUp, Ctrl—PgDn, Ctrl—End, Ctrl—←, Ctrl——→.

·续行键～ 在键盘输入缓冲区内最多含120个字符.若在按某一行为键之前该缓冲区装满，则以后每敲入一个字符将导致响一次嘟嘟声．这时，你应用 BackSpace 退到已输入数据中的某一空格处、敲入续行键～、按 Enter/Return键，并在下一行接着敲入你尚未输完的数据．用续行键～可使长数、长名、长命令正确跨过多行．但退格键BackSpace不能用来删去按Enter/Return键之前敲入的数据．

§1·3 编　辑

知道 LISP 的启动方法和键盘的使用后，应了解如何输入、修改、显示程序等．§1.1曾举例说明了在LISP控制下直接输入并运行程序的方法．遗憾的是，人们很难保证程序一次输入就能正确无误，且输入后永不修改．因而，我们需要考虑程序的建立、修改、显示等编辑问题．

有两种方法编辑程序，一是在 LISP 系统之外用 DOS 的行编辑程序来建立、修改、显示LISP程序．编辑后的程序待启动 LISP 后用FLOAD调入并运行．另一种方法是结构编辑，即在 LISP 控制下输入、修改、显示 LISP 程序，编辑好后直接运行．结构编辑程序的方法在§4.13详细介绍，这里简单介绍在DOS控制下用行编辑程序EDLIN编写、修改、显示LISP程序的编辑命令，其中：

1.行编辑程序的提示符同LISP的提示符一样，均是*．

2.行编辑命令以按Enter键后有效，以按Ctrl—Break键结束．

3.命令中的形式参量Line指的是行号．行号可为1至65529之间的一个十进制整数．大于内存行号的line意即最后一行之后．#意即最后一行．·意即当前行，当前行指出文件最后一次修改的位置，当前行的标志是，行号与该行文本第一字符之间有一星号*．－line和＋line分别表示当前行之前第line行和当前行之后第line行．

4.一行上可有多个行编辑命令．命令之间用分号隔开．

5.控制字符可插入文本，也可在检索文本和替换文本命令的串中使用．办法是，先按Ctrl-V，再用大写字体输入所需控制字符．例如，按Ctrl-V后，再按Z便产生控制字符Ctrl-Z．

6.DOS编辑键可用来帮助行编辑．起初，将当前行的副本作为由DOS编辑键处理的样板，这样板将随着DOS编辑键的操作而变化．由DOS编辑键对样板编辑的结果放在样板行下面的显示行．下面就是DOS编辑键的功能：

- Del　删掉样板中的当前字符，显示行不变．
- Ins　在显示行光标位置插入跟在Ins之后的字符．
- Esc　取消显示行，样板不变．
- F1　将样板行中的当前字符复制到显示行亮标所指位置．
- F2　将样板行中第一字符到F2后面指定字符间的所有字符复制到显示行，但指定字符

不复制。

- F3 将样板行中所有剩余字符复制到显示行。
- F4 跳过样板行中第一字符起到跟在F4后面那一指定字符止的所有字符，但指定字符不跳过。
- F5 将显示行作为样板行，并将记号@放在显示行后以示该显示行已成为新的样板行。

下面就是行编辑命令。

EDLIN 〔d:〕〔Path〕file.lsp〔/B〕

d：驱动器编号，若不提供此参量，则用默认驱动器编号。

Path：路径。

file：文件名，规定该文件的扩展名为LSP，这便于LISP调用。

/B：若不提供此参量，EDLIN在装入〔d:〕〔Path〕〔file·lsp〕时遇到Ctrl-Z便认为文件已装毕。若希望在你的文件中包含Ctrl-Z字符，则须带/B参量。

该命令启动行编辑程序。若 file.lsp 是新文件，则本命令显示新文件信息和提示符*：

```
New file
*
```

若file.lsp是驱动器d上在路径Path下的一个已存文件，则将该文件装入内存，并显示：

```
End of input file
*
```

〔line〕I 这是插入行命令。该命令指示行编辑程序在行line之前插入文本。若省掉参量line或line为·，则将文本插在当前行之前。若line大于现有最大行号，或line为#，则将文本插在最后一行之后。插入文本时，每插完一行应按一次Enter，每按一次 Enter，连续行号将自动出现。插完后要用Ctrl—Break来中断插入方式。

〔line1〕〔，line2〕L 显示文本命令L让行编辑程序将第line1至line2中的文本显示在屏幕上。若省掉参量line1，则从当前行之前的11行开始显示，一直显示到 line2 止。若省掉参量line2，则从line1开始显示，共显示23行。若省掉line1和line2，则显示当前行前11行、当前行以及当前行后11行。

〔line1〕〔，line2〕D 删除行命令指示行编辑程序将从line1 到 line2 的文本行删除，紧接line2的行变为当前行，紧接line2 后的后随行均重新编号。若省掉参量 line1，则将从当前行起至line2止的文本行删除。紧接line2 的行变为当前行，line2 后的行均重新编号。若省掉参量line2，则仅删除第line1行。若两个参量均省掉，则删除当前行。

〔line〕 本编辑行命令对编号为line的行进行编辑。若line为·，则对当前行进行编辑。若省掉参量line，即仅按Enter键，则对当前行之后的一行进行编辑。在此编辑行命令发布后，行编辑程序显示行号line及该行文本，并将之作为样板行。同时，将行号 line和*显示在下一行作显示行。这时，用户可用前面讲的DOS编辑键对样板行 line 进行编辑，也可重新敲入新的内容。在样板行line编辑完后，若用户按Enter键，则用显示行line取代样板行line，并

默认line是当前行．若用户按Esc键或按Ctrl—Break键，则样板行line不变．若光标位于显示行line之始点，则按Enter和Ctrl—Break均不改变样板行 line，只是退出本编辑行命令．若光标处于显示行line的中间，则按Enter键会删去样板行line的其余字符，并用显示行line替代样板行line．

〔line1〕，〔line2〕，line3〔，count〕C　该拷贝行命令将行 line1 到行 line2 全部拷贝到行line3之前，此拷贝工作重复count次．拷贝完毕后，用line3作为当前行．若line1或 line2省略，则默认当前行；若count省略，则默认1．此外，字符＋、－均不能放在count参量内．

〔line1〕，〔line2〕，line3　M　移动行命令M将行line1至行line2 全部移到 line3之前．移动后的line1作为当前行．若line2参量为＋line2′，则将从line1 起共 line2′＋1 行的文本移到line3之前．若line1或line2省略，则认为它们指当前行．

〔line1〕〔,line2〕〔？〕R〔string1〕〔‹F6›string2〕　该替换文本命令将line1至line2之间所有出现的string1用 string2 替换．若用户提供参量?，则每当一个修改的行显示后，行编辑程序将用O. K. ？向你提问．若你回答Y，则保留改后内容；若你回答任何非Y键，则刚作的修改作废．另外，string2前的功能键F6将显示为^Z（即Ctrl—Z），它标志着string2的开始．

〔line1〕〔,line2〕〔？〕S〔string〕　本检查文本命令在line1至line2之间查找string 的出现，并在含string的那一行结束检索．若用户提供参量?，则每找到一次string，就显示找到string的那一行，之后用O. K. ？向你提问．若你回答Y，则与 string匹配的行作为当前行，并结束检索，否则继续检索string的另一出现．

〔line〕T〔d:〕file.lsp　传送行命令 T 将驱动器 d 上的另一文件 file.lsp 读入内存，并放在line之前．其中file是你的文件名，lsp是文件扩展名．

Q　这是退出行编辑程序的命令．当此命令发布后，行编辑程序显示：

Abort　edit (Y/N)?

若此时你输入 Y，则退出行编辑程序，所编辑的文件不作任何更改地放在磁盘内．若你回答N 或其它非Y字符，则继续编辑．

E　该结束编辑命令将已编辑内容存在磁盘上进入 EDLIN 时 规 定的 文件中，并返回到DOS．

其它与行编辑有关的命令A、P、W可参阅你的DOS说明书．此处用得不多．

§1·4　出　　错

出错类型有多种．常见的有程序运行有错、中途打断程序运行出错以及空间不够出错．程序运行有错时，LISP就停止对出错表达式的求值、显示出错信息、并根据不同的运行环境

对错误作不同处理。若你运行一般 LISP，所有出错都交给 ERROR 函数处理。若你运行宏 LISP，则出错后由错误监控程序处理。后者优于前者的主要原因是，它利用了提供在IBM—PC机宏LISP语言中的窗口函数、上下文控制函数和功能键支撑函数，从而能交互式地找出出错根本原因、动态修改并继续运行改后程序。

中途打断程序运行导致的出错是用户故意造成的。当某一程序运行许久没有自然终止、用户怀疑程序进入死循环时，可按Ctrl—Break键来中断程序运行。这时，出错信息为(^C)，用户可继续按 ^^ 回到LISP的顶层控制。

有两种空间不够出错信息，即

Work Space Exhausted

与

Stack exceeds 64 K—Reset to top level

前者说，内存空间用光；后者说，用来保存环境的64K 堆栈超过。出现这两种错误的原因有二，一是内存或堆栈的确用光，一是由于程序或数据方面的错误导致内存或堆栈不够.例如，任何时刻运行程序：

(DEFUN A (LAMBDA ( ) A))

将导致堆栈用光出错信息。对这类错误的处理方法是先查程序或数据错误,再考虑程序分段、删去内存中不必要的函数、数据或扩大内存的问题。

§1·5 8087

IBM—PC机宏LISP语言提供了许多作浮点运算和比较的算术函数、三角函数、求根函数、指数和对数函数。这些函数的运行依赖于所用计算机上是否配置8087协处理器。8087和8086/8088一起保证了PC机上的浮点数运算。若你的PC机没有8087，可在市面上购买8087。打开你的主机机壳，将8087插入8088旁边标有8087字样的槽内、保证8087和8088缺口方向一致、将Sw1开关第2位置为OFF状态、装上你的PC机壳。这时，你的 PC 机便已配上8087协处理器了。

之后，在第一次启动LISP并希望启动后作浮点运算时，要先执行 (I8087 T)。该函数告诉LISP用8087作浮点运算。

如果你的PC机没有8087，一定不能执行(I8087 T)，那样会使PC机挂起，且只有断电才能恢复正常。为让那些没有8087的用户也能作浮点运算，LISP提供了用软件实现浮点运算的库函数 FLOAT.LSP。用 (FLOAD 'FLOAT) 将此库函数调入内存后，可象有 8087一样作浮点运算，但仍旧不能求解三角函数、求根函数、指数和对数函数。此外，有8087作浮点运算比无8087用软件实现浮点运算要快几倍，并要节省许多内存空间。

若在你的PC机上有8087但未执行(I8087 T)、或没有8087又没执行(FLOAD 'FLOAT)，则每作一次浮点运算就会导致一次出错。与8087有关的函数均在第三章介绍。

§1·6 库 盘

在MACRO—LISP—LIB库盘里含有20个库文件，用来扩充或帮助扩充基本LISP的功能。当你启动宏 LISP时，内存已装入其中11个文件，另9个文件在你需要时调入。现简单介绍这

20个库文件的功能。

1 **ARITH.LSP** 该文件长5826字节。当用户想用中缀算术表达式时用 (FLOAD 'ARITH)、将包ARITH调入。在LISP回答提示符＊后，便可使用诸如{3＋5＊6}的中缀算术表达式。若为节省内存并已习惯前缀算术表达式，则可用 (PDELETE ARITH) 将该包从内存删去。在本书§4.5详细介绍了宏LISP对中缀算术表达式的处理。

2 **BASE.LSP** 该文件在启动宏 LISP 时装入内存，共有1838字节，其中含有库函数DEFUN、DEFRM、DEFSM、FLOAD、FNAME、MEMBER、PACKAGE以及服务函数MPUT和MREMPROP。

3 **COMMENT.LSP** 该文件长 425 字节。当想在程序中插入注解时，可用 (FLOAD 'COMMENT)将该文件作为包COMMENT调入；不需要时用 (PDELETE COMMENT)将之删去。注解用一对花括号{ }括起，中间可含除右花括号}外的任何字符。COMMENT 包定义了读宏{，该读宏使{注解}读为(；注解)。然后，(；注解)由COMMENT 包中的函数；解释为NIL。也就是说，注解的返回值为NIL，因而可放到程序中能置放NIL的任何地方。

4 **CONTEXT.LSP** 该文件在启动宏LISP时作为包CONTEXT装入内存。共有1955字节，内含支撑ERROR查找上下文堆栈的函数STKPINIT、STKSRCH、UP、DOWN以及NEXTEV、PREVEV、＊NPEV等辅助函数。

5 **DEBUG.LSP** 该文件在启动宏LISP时作为包DEBUG装入内存，共有2831字节，内含追踪、打断、撤销追踪以及撤销打断等库函数。

6 **ED.LSP** 本文件在启动宏LISP时作为包ED装入内存，共有18512字节，内含用来实现结构编辑的库函数ED、EDF、EDP、EDV，以及编辑命令COP、CW、DEL、F、FB、IA、IB、MOV、NEXT、P、PP、PREV、REP、SA、SB、SL、UL、X、^、^^、?、＊以及定位整数命令等。

7 **ERROR.LSP** 本文件在启动宏LISP时作为包ERROR装入内存，共有5683字节，内含帮助分析、查找、修改错误的若干函数，该包要求FKEY包和CONTEXT包支撑。

8 **FKEY.LSP** 本文件在启动宏LISP时作为包FKEY装入内存，共有1949字节，内含帮助定义功能键的函数FKEY、FKPUSH、FKPOP、FKDEF、FKHELP以及几个辅助函数。

9 **FLOAT.LSP** 该文件长11776字节，当用户所用PC机未配8087又需用浮点运算时，用 (FLOAD 'FLOAT) 将该文件作为包 FLOAT 调入内存。这时函数 i＊、i＋、i－、i/、iABS、iADD1、iEQUAL、iGE、iGT、iLE、iLT、iMINUS、iSUB1 作整数运算，去掉i后的相应函数符作浮点运算。

10 **HELP.LSP** 该文件长9612字节，当用户想动态地在机子上查找第三章中所述基本函数的调用格式、参量个数以及参量类型时，可先用(FLOAD 'HELP) 将该文件调入内存。以后，在LISP执行期间，用

(HELP ‹functionname›)

便可查到名为 ‹functionname› 的基本函数的调用格式。若 ‹functionname› 不是一基本函数名，则显示 "No Def" 并返回 NIL。若 ‹functionname› 是一基本函数名，则显示一张表，表的第一元素是基本函数名 ‹functionname›，表的其余元素是参量表。显示这信息后返回NIL。函数名不同，显示的信息也自然不同。例如，(HELP ASSOC) 将显示

(ASSOC ‹item› ‹list›)

意即：基本函数 ASSOC 带两个参量，一个是项，另一个是表。又如，(HELP MAP) 将显

示：

(MAP ‹function-definition-or-name› ‹list› /// ‹list›)

意即：基本函数MAP带若干个参量，第一参量是函数定义或函数名，后接个数不定的表参量。因为由HELP显示的信息在装文件HELP.LSP进内存时要一一存在各基本函数的特性表上，所以装入HELP.LSP文件要花约2分28秒，但装入内存后使用起来既方便又迅速。若你不需某些函数（如:CAR、CDR、CONS、LIST）的帮助信息并想节省由它们占有的内存空间，则可通过执行下一函数来作到：

```
(MAPC '(LAMBDA (L) (REMPROP L 'HELP))
      '(CAR CDR CONS LIST))
```

之后，你再用HELP询问那些已删去HELP特性的基本函数时，LISP将回答“No Help”并返回NIL。

11 **MACRO.GEN** 在你启动一般LISP并运行一段时间之后，若想启动宏LISP，则仅发布下述命令：

```
(EVAL (READ (INPUT "MACRO.GEN" )))
```

上述命令的功能是将文件BASE.LSP、包QUASI、UTIL、XPROG、FKEY、CONTEXT、ERROR、PP、ED、DEBUG、PACKAGE装入内存。其实，用户可仿照MACRO.GEN编写自己的环境生成文件，并在启动一般LISP之后，照调入MACRO.GEN的上述命令启动你自己的运行环境。

12 **MAXMIN.ASM** 该文件有4412字节，内含求任意多个参量之极大值、极小值的汇编函数MAX和MIN。本文件可在DOS控制下用TYPE打印出来。从中可以了解与LISP相连的汇编程序的编写方法。

13 **MAXMIN.OBJ** 该文件有234字节，是用标准IBM PC机汇编编译对汇编程序MAX、MIN编译后的目标代码。当你想快速求出任意多个参量的极大值和极小值时，可采用下述步骤将MAXMIN.OBJ调入内存：

```
(FLOAD 'OBJLOAD)
(OBJDEF (MAX MIN) "MAXMIN.OBJ" NIL)
```

在FLOAD 和 OBJDEF回答一些信息并显示提示符*后，你便可以运行目标函数MAX和MIN了。如：

(MAX 1 2 3 4) 和 (MIN 4 3 2 1)

14 **OBJLOAD.LSP** 该目标文件装配程序占14422字节，当你想调入并在LISP控制下运行你的目标程序时，先要用 (FLOAD 'OBJLOAD) 将该文件作为包OBJLOAD装入内存。该包有两个主要函数OBJDEF和OBJLOAD以及若干个辅助函数，它们均用LISP程序书写。由于该包用到包XPROG和包UTIL中的一些函数，故在装入本包时，XPROG包和UTIL包一定要先装入内存。

15 **PACKAGE.LSP** 该文件在启动宏LISP时作为包PACKAGE装入内存，共有7689字节，内含用来实现包的包支撑函数和命令：PCHK、PDELETE、PDIR、PSAVE、EDPK、DEF、DEFRM、DEFSM、DELETE、EVAL、FLAG、PUT和SETQ等。

16 **PP.LSP** 本文件在启动宏LISP时作为包PP装入内存，共有6678字节，内含整齐输出函数PP和整齐打印函数PPF。

17 **QUASI.LSP** 该文件共1918字节，在启动宏LISP时作为包QUASI装入内存，内含