

APPLE II 资料(四)

6502 汇编 / 编辑程序

BASIC 程序设计手册

目 录

6502 汇编/编辑程序

引 言

§ 1 系统的描述	(1)
§ 2 在你开始使用之前	(1)
§ 3 系统的启动	(1)

第一章 编辑程序

§ 1 编辑程序的说明	(3)
§ 2 命令的输入	(3)
§ 3 命令方式下的灵活措施	(4)
1)、多个命令的输入	(4)
2)、命令分界符的设置	(4)
3)、反复执行上一条列表指令	(4)
4)、直接磁盘操作系统指令	(5)
5)、命令语法结构提示帮助	(5)
§ 4 参数表的语法结构	(5)
§ 5 编辑命令	(7)
1)、ADD 加	(7)
2)、COPY 复制	(7)
3)、CHANGE 更换	(8)
4)、DELETE 删去	(8)
5)、EDIT 编辑	(9)
6)、FIND 查找	(11)
7)、INSERT 插入	(11)
8)、LIST 列表	(12)
9)、PRINT 打印	(12)
10)、REPLACE 置换	(12)
§ 6 磁带和磁盘的命令	(12)
1)、LOAD 装入	(13)
2)、SAVE 保存	(13)
3)、APPEND 增补	(14)
4)、TLOAD 磁带装入命令	(14)
5)、TSAVE 磁带存入命令	(14)
§ 7 操作命令	(15)
1)、SLOT (插座位置)	(15)
2)、DRIVE (驱动器)	(15)

3)、CATALOG (编目)	(15)
4)、FILE (文件存贮)	(16)
5)、HIMEN = (上界)	(16)
6)、LOMEN = (下界)	(16)
7)、LENGTH (长度)	(16)
8)、MON (监控)	(16)
9)、NEW (刷新)	(17)
10)、PR* (打印)	(17)
11)、TRUNCATE (截短)	(18)
12)、TABS (标记)	(18)
13)、WHERE (找地址)	(18)
14)、END (编辑结束)	(19)

第二章 汇编程序

§ 1 汇编程序的概述	(20)
§ 2 启动之前	(20)
§ 3 汇编命令 (ASM)	(21)
§ 4 汇编方式命令	(22)
1)、ABORT ASSEMBLY (停止汇编)	(22)
2)、SUSPEND OR SINGLE—STEP LISTING (中止或步进地列表)	(22)
3)、LIST PART OF PROGRAM (程序的表格部分)	(23)
§ 5 源程序的格式	(23)
1)、标记	(23)
2)、操作码	(24)
3)、操作数字段	(24)
4)、注释字段	(24)
§ 6 操作数字段的构成	(24)
1)、标号	(25)
2)、常数	(25)
3)、保留字	(26)
4)、算术运算符	(26)
5)、地址表达式	(26)
§ 7 汇编程序的命令	(26)
1)、ORG (起始值)	(27)
2)、OBJ	(27)
3)、EQU	(27)
4)、MSB (最高位)	(27)
5)、DSECT	(28)
6)、DEND	(28)

7)、REL (RELocatable).....	(28)
8)、EXTRN	(28)
9)、EXTRY	(29)
10)、CHN (链接).....	(29)
§ 8 列表命令(共有六种).....	(29)
1)、PAGE	(30)
2)、LST.....	(30)
3)、REP	(30)
4)、CHR	(30)
5)、SKP	(30)
60、SBTL	(30)
§ 9 数据定义命令(六条).....	(31)
1)、ASC.....	(31)
2)、DCI.....	(31)
3)、DFB	(31)
4)、DW	(31)
5)、DDB	(32)
6)、DS	(32)
§ 10 条件汇编命令	(32)
1)、DO	(33)
2)、ELSE	(33)
3)、FIN.....	(33)
§ 11 寻址方式一览表	(33)
§ 12 汇编程序命令一览表	(34)
§ 13 操作码一览表	(34)
§ 14 列符号表	(35)

附 录

附录 A 编辑程序/汇编程序的内存使用情况.....	(37)
附录 B 编辑程序时的磁盘操作系统的错误.....	(39)
附录 C 浮动装入程序.....	(42)
附录 D 汇编程序的“OOPS”磁盘操作系统出错码.....	(44)
附录 E 目标文件之格式.....	(46)
附录 F 符号表之格式.....	(49)
附录 G 编辑 BASIC 程序.....	(51)

第一章	引言	(53)
§ 1	所需附件	(53)
§ 2	联接电视监控器或电视机	(53)
§ 3	插入游戏控制器	(53)
§ 4	接上盒式录音机	(53)
§ 5	Apple 的键盘	(54)
§ 6	键盘的标注法	(55)
§ 7	控制和其它不显示出的字符	(55)
§ 8	进入 BASIC	(56)
§ 9	调节磁带录音机	(57)
§ 10	一个很有用的提示	(58)
§ 11	使计算机停止工作	(58)
§ 12	如果你偶然触动了 RESET 怎么办	(59)
§ 13	带用的装入磁带的步骤	(59)
§ 14	调整电视屏幕的色彩	(59)
§ 15	玩“爆发”游戏	(61)
第二章	开始用 BASIC	(62)
§ 1	开始用 BASIC	(62)
§ 2	32767 有什么特殊之处	(63)
§ 3	关于 RETURN 键的进一步说明	(64)
§ 4	在你按动 RETURN 键之前应做些什么	(64)
§ 5	屏幕上着色	(65)
§ 6	画图出错信息	(66)
§ 7	画线	(67)
§ 8	游戏控制	(68)
§ 9	暂存单元 (pigeonholes) 和更多的计算能力	(68)
§ 10	模拟掷一对骰子的例子	(70)
§ 11	优先级	(71)
§ 12	如何改变优先级	(73)
第三章	基本的程序设计	(74)
§ 1	延迟执行	(74)
§ 2	基本的编辑	(74)
§ 3	基本的循环	(77)
§ 4	一些有用的功能	(78)
§ 5	移动光标能擦除或复制文件中的任何字符	(79)

§ 6 关于学习 BASIC	(80)
§ 7 发生意外.....	(80)
§ 8 真假值.....	(80)
§ 9 IF 语 句.....	(83)
§ 10 更复杂的作图程序.....	(85)
§ 11 一些方便操作的功能.....	(86)
§ 12 打印格式的控制.....	(91)
§ 13 关于竞赛程序的一点说明.....	(93)
§ 14 程序的保存.....	(94)
§ 15 从器壁上弹跳回来的例子.....	(96)
§ 16 发出声音.....	(98)
§ 17 弹球的声音.....	(99)
§ 18 一行上有多个语句的特性.....	(99)
§ 19 在一行上多个语句和 IF 语句	(100)
第四章 字符串、数组和子程序	(101)
§ 1 字符串.....	(101)
§ 2 联接.....	(104)
§ 3 数组.....	(105)
§ 4 有关数组的出错信息.....	(107)
§ 5 数组的初值.....	(108)
§ 6 排除故障.....	(108)
§ 7 关于质数的一点提示.....	(111)
§ 8 子程序.....	(111)
§ 9 跟踪.....	(113)
§ 10 一个更好的画马子程序例.....	(114)
§ 11 结束语.....	(116)
附 录.....	(117)
§ 1 APPLE 计算机的信息: 产生的原因及消除的方法.....	(117)
§ 2 使程序运行得更快些.....	(120)
§ 3 若干其它的函数和能力.....	(121)
§ 4 POKE, PEEK 和 CALL 指令.....	(121)
§ 5 整型 BASIC 中的保留字	(125)
§ 6 整型 BASIC 程序例	(126)

引 言

§ 1 系统的描述

Apple II 汇编/编辑系统是为 6502 汇编语言的大量程序进行汇编和编辑而设计的。该系统包含了三种程序：编辑程序，汇编程序，以及调用编辑程序和汇编程序的命令解释程序。命令解释程序通常是存贮在存贮器中的，而编辑程序只有在屏幕上出现冒号，且此时没有汇编时才存放在存贮器中。当执行 ASM 命令时，汇编程序才被装进存贮器，并用它取代编辑程序同时还清除掉所有的编辑文件。而当完成了汇编工作之后，编辑程序就立刻代替汇编程序被装进存贮器。

编辑程序用来产生按汇编程序所要求的格式的源程序，它也可用来形成 EXEC 文件以及编辑或形成 BASIC 程序。编辑程序在磁盘操作系统的控制之下可以处理磁盘文件或盒式磁带文件。

汇编程序的功能是把源程序译成 6502 机器代码。它从磁盘中读出源程序文件，然后又直接将每个输出模块写回到磁盘中去。用把多个源文件链接在一起的办法，汇编程序能够对非常大的程序进行汇编。汇编程序仅仅只将符号表保存在存贮器中，如果需要的话也可以把浮动目录保存在存贮器中。因此，一个汇编的大小不局限于源文件的大小。

要充分使用所有的系统特性，就需要一台至少有一个磁盘驱动器和带有 P5A 和 P6A 控制器的容量为 32K 的 Apple II 或 Apple II 机器。更大的程序则要求有一个容量为 48K 的系统以及为源文件需有附加的磁盘驱动器。编辑程序可以对 30K 以下容量的源文件进行编辑或者对 1000 到 1500 行源代码的注释行进行编辑。

汇编程序/编辑程序系统被设计或在 16 段磁盘操作系统下进行工作。系统软盘含有一个磁盘操作系统的拷贝。因此，该系统总是在适当的环境下运行。系统的磁盘操作系统文件的结构是和 Applesoft 及整型 BASIC 相兼容的。而与 Apple Pascal 系统是不兼容的，因为 Pascal 系统有它的自己的编辑程序和汇编程序。

§ 2 在你开始使用之前

在你准备使用编辑程序和汇编程序之前，你应该对 APPLESOFT “工具箱”系统软盘复制一个备用拷贝。为了做到这一点，首先必表对原版进行写保护，然后从磁盘操作系统主盘中调出磁盘操作系统，再运行拷贝程序。

这本手册假设了你已经在 6502 汇编语言程序方面具有一些经验，并且熟悉 APPLE 磁盘操作系统。假如你对使用磁盘没有经验，则你必须在使用该系统之前阅读一下磁盘操作系统手册。如果你有一个单驱动系统，那么可以利用在 Apple 实用软盘上的 FID (文件复制) 程序产生一个备用拷贝。

§ 3 系统的启动

为了运行汇编程序/编辑程序，可先装好 Apple “工具箱”软盘，然后打入：

RUN EDASM

如果系统回答了：

LANGUAGE NOT AVAILABLE 则再打入

RUN INTEDASM

注意：“工具箱”软盘必须装在驱动器 1 上。因此，命令翻译程序完全可以装入编辑程序和汇编程序模块。

现在，在屏幕上就会显出汇编程序 ID 标记：

```
APFLE II EDITOR—ASSEMBLER
CURRENTASSEMBLER ID STAMP 15;
@ 9 —NOV—79      # 0 0 0 0 0 0
```

用箭头键→和←利用移动光标的办法在屏幕上对所显示的信息进行修改，然后在你想修改的那个字符的位置上打印上一个新的字符。以同样的方法对 BASIC 程序行进行编辑，在你认为满意时，按一下 RETURN 键。系统的名称和拷贝信息将显示在屏幕上，并且在屏幕的底部显示系统提示符和一个冒号，后面接有光标和一个闪烁的下划线。

注意：如果磁盘操作系统已经被调出且已运行的话，那么你就可以运行汇编程序/编辑程序，而不必再重新调出磁盘操作系统。假如你不想修改 ID 标记，则在装好汇编程序/编辑程序系统软盘后，打入 BRUN EPASM OBJ 并按下 RETURN 键即可。

此时，执行所有的编辑命令可以通过简单地打入命令，然后按下 RETURN 键，在一行的开始处出现一个冒号提示符，总是表示命令方式。

第一章 编辑程序

§ 1 编辑程序说明

汇编程序/编辑程序系统的命令解释程序通过在每一行的开头处由所显示出来的冒号表示它已准备好，可接收命令了。命令可以用简单地打入其命令。然后按下 RETURN 键的方法输入到机器中去。命令解释程序对所读出的命令的头、尾及中间的空格均不予识别。命令的名称中的任何任选字符不能作为命令后面的参数的起头。但由于命令后面的参数大多数都是数字，所以很少用到这种限制。也很注意到在命令的名称中可以省略掉一个或几个任选字符，只要按正确的顺序出现即可。例如：列表命令(LIST)可以用 L、LT、LS 或 LIT 来代替。

编辑程序在被编辑的文本文件之中不保留行号，也不按照你的命令中所用的行号相对应数来查找文件。编辑程序对“行”作如下的定义：“行”就是两次 RETURN 之间的所有语句。编辑程序产生行的数目，此数目是从文件的起头开始一行一行数下来的数字表示的。因此，当删去一行或几行时，所有在它后面的行的数目会自动地减小。同理，当插进新的行以后，则在它后面的行的数目会自动的变大。因而，编辑程序中所使用的行的数目仅仅只是作为编辑程序的指向，而不是属于文件的一部分。

这种相对行号的方法，在使用编辑程序时，有大量隐含的意义。关于这一点将在有关的命令描述中加以讨论。编辑程序仅仅能编辑一个能存入编辑缓冲器中的文件。对于 48K 系统来讲，编辑缓冲大约占 2 K 容量；而对于 32K 的系统来讲，编辑缓冲器约占 13K 容量。一个典型的 6502 带有注解的“源代码行”大约为 40 个字符长，因此，一个 48K 的系统至少可以编辑 700 行的一个程序，或许还能编辑到 1400 行的一个程序，这是因为许多行是非常短的或者没有注解。

编辑程序提供了 16 个可调整屏幕位置的标记，以使文本成为容易阅读的格式，这些都将在“Tabs”命令这一节中叙述。编辑程序仅仅用 40 列的 Apple 视频显示屏显示文本文件，也期望通过 Apple 机的键盘来接收输入。可以使用外终端设备，但是输出程序将从第 40 列开始绕到下一行。Edit 命令(编辑命令)设计得仅仅只用 Apple 显示器和键盘进行操作：它可以在键盘上的键每次按下去时将经编辑的行输出到外终端设备上。

这种编辑程序不打算成为“字处理”编辑程序，而主要是为了程序的编制，用磁盘操作系统手册中“EXEC 命令”所描述的步骤把 Applesoft 和整型 BASIC 程序转成文本文件的方法来对它们进行编辑完全是可以的。在转换以后，该文本文件就可以用编辑程序来进行编辑了，这种类型的编辑在附录 G 的“编辑 BASIC 程序”一节中再加以详述。

§ 2 命令的输入

在编辑程序中有一个 Help 命令，为了看到 Help 显示，先启动系统然后打入一个？再按下 RETURN 键，你就可以在屏幕上看到各种命令的语法结构。HEIP 显示占了两幅屏幕，想看第二幅则必须再按一下 RETURN 键。

当你输入编辑程序的命令时，你可以对命令进行简略化，这样就可以使用最少的字符来确定一个命令。例如：List 命令可以简略成 L、LI、LIS 或 LIST。在显示屏上你可以注意到每种命令所要求的字符以一种正常显示方式显示出来，而可任选的字符则以相反的方式（白底黑字）显示出来。

这种选择字符的方式只能用于字母与逗号，在 ASM 命令中，如果逗号后面的参数都省略掉时，那么该逗号才也可省去，这在 ASM 命令和 COPY 命令中详细加以讨论。以 HEIP 显示时，命令的名称之后，参数的语法结构只是显示出其最一般的形式，对于许多命令比较复杂的参数表都是合法的，并且将在后面的每个命令中给予讨论。

§ 3 命令方式下的灵活措施

当处在命令方式时(即正显示冒号：标记时)，某些灵活措施可以使系统使用起来更为方便，这一节就是要叙述这一问题：

1)、多个命令的输入

这个灵活措施使得你可以在一行上打入多个编辑程序命令，命令解释程序会把第二个和其后的各个命令存贮起来，并且执行了每条命令之后，顺序取出下一条命令。多条命令由命令分界符分离，在你未作修改之前，命令分界符是用冒号：来表示的。如果有一条命令有错的话，则清掉队列中剩下的所有命令。如果 Find, Chang 或 Edit 指令查找串失败的话系统并不认为出错。

如果 ASM 命令进入队列，那末它应该是最末一条被执行的命令。当汇编程序被装入之后，则在它后面的任何命令都将被冲掉。

在使用命令队列时，要很小心，因为如果一个命令改变了文件行的行号，那么命令队列中嵌入使用的行号可能不再够用了。Add、Insert、Copy、Delete 和 Replace 等命令，依照它们各自的功能，都能够改变文件中的行号。队列的内容在任何时候，都不能由输入方式或编辑方式进行存取。因此，有可能在一条进入这些方式之一的命令之后，接有另一个命令，象 List 命令或 List Recall 命令（下面要叙述到）。这样可以使你用多个插入（Insert）或加进（Add）命令来点缀其它命令，这种灵活措施在实际应用中很有效。

2)、命令分界符的设置——

命令分界符的设置能力使得你能改变命令分界符，它的予置字符是冒号，这种功能是要的，因为命令分界符不能用作 Find Chang 或 Edit 命令中的查找串。选用冒号：作为预置字符的道理是因为该字符通常不在 6502 汇编语言源程序中使用。

为了改变命令分界符，需按一下 ESC 键（“换码符”键），然后再按“冒号键”，以及所要的分界符字符。假如你按了“冒号键”之后又按了一下“空格键”，那末系统将会发出嘟嘟的响声并等待你继续打印另一个字符。如果你按下 RETURN 键，那么系统就会立即打印出当前的命令分界符，然后回到命令方式，任何其它字符都可以用来作为分离命令用的分界符。不能用句号。作为命令分界符，因为它是直接磁盘操作系统命令的换码字符。

3) 反复执行上一条列表指令——

这是一个简单的灵活措施，就是当列表指令被执行时再把它保存起来，并且用一个单字符将它连同其所有的参数一起重新调入，为了达到重新列表的目的，先打入 CTRL—R，然

后按下 RETURN 键，或者打入命令分界符以及更多的命令。那么象其他任何别的命令一样，重新列表命令进入队列，记住，系统总是重复执行上一条列表命令而不管上一条列表命令何时已使用过。CTRL—R 必须作为命令的首字符。

4)、直接磁盘操作系统指令——

这可以使你执行大多数的磁盘操作系统指令而不需要离开命令方式，这样，在编辑程序的工作范围内增加了文件管理的能力，同时使编辑程序保持较短，因而使 APPLE 机的主存可以容纳更多的文件。它被设计得只可调用以下的指令：

RENAME LOCK UNLOCK MON 以及 NOMON

当使用 Applesoft 或整型 BASIC 时，要调用直接磁盘操作系统指令时，只需简单地打入一个句号。来作为命令行的首字符，然后再完整地打入磁盘操作系统指令即可。

例如：命令

：• RENAME XX, YY,

可以重新命名当前在磁盘驱动器上的文件。

汇编程序/编辑程序系统会受到初用者乱用而破坏。这个系统是这样被设计的，对于熟练的程序人员来说能提供最大的效能。直接磁盘操作系统指令对其现有的标准 DOS 指令的存取不给予限制，这些命令可以执行 BASIC 程序。这种灵活措施使用时必须小心，否则很容易破坏主存中的任何文本内容。

如果你使用任何一种存数、取数命令，任何一种机器语言文件命令，或者是任何一种 BASIC 命令，如 LOAD AVE, RUN 或 MAXFILES 等命令时，那么也许会出现不可预测到的结果。但是，如果你滥用了该灵活措施，那么多半会破坏汇编程序/编辑程序。如果你通过此灵活措施执行 INIT 命令的话，那么初始化过的软盘上将不含有 HELLO 程序，因为在存内中没有供 INIT 使用的任何东西，而且该盘不能自举，如果你想初始化系统的主程序，那么，你先要搞清是否制好了一个备用盘。

5)、命令语法结构提示帮助——

这是？命令或 Help 命令，当你打入一个？来作为命令时，就会显示出一个专用的命令语法结构表。机器显示出两幅缩写的命令语法结构，为了能看到第二幅，必须再按一下任何一个键。这是为了在你的手边提供一个“参考表”。你也可以在？的后面送一个命令进去，但显示出来的只是这个命令的语法结构。如果你打入一个不能理解的命令，那么，就象在？后没有命令一样，进行完整的显示。

§ 4 参数表的语法结构

在下面的对命令的讨论中，这里所定义的术语有时是用来说明命令的参数表的语法结构。有些命令没有参数而有些要求有参数或任选参数，或两者都要求有。下面的表中描述了每种类型的参数，但是命令可以使用它们的各种组合。在每一种指令说明中给出的次序就是参数输入所需的次序，如果不需要任选参数的话，可以省掉。

串参数要求有一个分界符（在语法结构图中用 (Delim) 表示），该分界符只能用一个不出现在被分界的串中的字符来表示，它不许用逗号(,)、连字符(—)、数字、空白或 RETURN。所推荐使用的分界符是单引号、双引号、句号、问号、感叹号和分号等。如果

你使用小括号或大括号，那么在命令中每个分界符必须要用相同的形状，即：两个左括号“(“才是正确的分界符，一左一右的括号“()”是不正确的分界符。

在下面的说明和例子中，竖线应该读作“或”，如：1 | 33333 | 0 | 6 应该读作 1 或 33333 或 0 或 6。

参数名称

说 明 及 举 例

Rnumber	要求一个十进制数，而数值范围由命令而定。1 33333 0 6 是正确的，不能用：• 3 — 4 C\$ 0 0 0
Onumber	任选的十进制数，其数值范围由命令而定。1 33333 0 6 是正确的。• 3 — 4 C\$ 0 0 0 是不正确的。
Rlinenum	要求范围从 1 到 65519 的十进制行号 1 33333 99 6 是正确的。• 3 0 — 555 是不正确的。
Olinenum	范围从 1 到 65519 任选的十进制行号 1 33333 99 6 正确的。• 3 0 — 555 是是不正确的。
Range	包含有一个或两个 Olinenum 的任选行号范围，如果有两个 Olinenum，那么必须用连字符“—”将它们分开，如果只出现一个 Olinenum，那么它就是范围的开头和结尾。第二个 Olinenum 可以用保留一个连字符“—”来省略，它隐含文件结束的意思。如果第二个 Olinenum 比第一个要小，并且它本身又是小于 100，那么第二个 Olinenum 必须加上第一个 Olinenum，如：100—99 应该看成 100—199。 1 — 2 1000— 500—666 1000—99 800—1 77—77
Rangelist	这是一组独立的范围，用逗号，进行分开，每个范围都独立地处理，表中每一个区域的行号的次序都不受限制。 1 — 2 1, 2 — 100, 3 1000—, 222—33, 4, 55—1, 77—77
Dstring	含有一个分界符，零或字母数字字符，以及相同分界符的分界字符串。 • What i Want to See • ; ; ' a good example,
Chgsting	Chgstring 与 Dstring 是相似的，它含有一个分界符，零或其它字符同上，以及相同的分界符，零或字符，以及再有一个相同的分界符。要进一步了解，请参见 Change 命令。? 旧字符串? 新字符串? % 旧的 % 替换的 %
Rfilename	要求文件名称是 1 到 30 个字母数字字符并不包括逗号，命名一个磁盘操作系统文本文件。 MYFILE MYBASICPROGRAM MYEXEC ETC
Ofilename	一个任选文件名称是 0 到 30 个不包含逗号的字母数字字符，命名一个磁盘操作系统文本文件。 MYFILE MYBASICPROGRAM MYEXEC ETC
OObjfilename	与 ofilename 相同的任选的目标文件名称，不过它指明的是由汇编程序产生的二进制的或浮动的目标代码文件的名称。

MYDRIVER OBJ | THE LONG NAME OF MY OBJECT

Devctlstring 为了初始化一台 Apple 外围设备以用来作为汇编程序的输出设备而设计的字符行。

CTL—I120N | CTL—AF

§ 5 编辑命令

这一节描述了操纵编辑文件内容的各种命令。下一节再描述将编辑文件内容在磁盘和缓冲器间相互传送的一些命令。第三节描述控制编辑程序的全部操作的命令。

在语法结构上需注意，命令名称的字母中的大写字母必须打入，才能执行那个命令。在方括号 [] 中的参数是任选的。

1)、ADD 加

Add[Olinenum]

ADD 命令通常是用来在编辑文件的尾部加一些新行。当你向命令提示打入 A，AD 或 ADD 时，系统就对文件中的行进行计数并指出下一个要加到编辑文件中去的行号。现在已经处于输入方式，那么，你可以把正文打入到编辑文件中，在每行结束时需按一下 RETURN 键。每按一次 RETURN 键之后，编辑程序就会很快地提示要加的下一行的行号。

当你把文本行输入完之后，（这些文本也可以是一个汇编源程序），就可以紧接在屏幕上显示的行号之后打入 CTRL—D 或 CTRL—Q，然后再按一下 RETURN。这样，就结束了输入方式，而由在新行上显示冒号：，表示你重新又可回到编辑程序命令方式。

你也可以带有行的数目使用加法指令，当你要这样做时，应该和上面叙述的一样，进入输入方式，但是新的字符行不是加到文件的最后一行上而是加在一给定数的字符行后面。这与插入命令的工作相似，但是不同的是插入命令把新的字符行是插入在给定数的行的前面。

当你处在输入方式时，那么标准的 Apple II 输入编辑功能全部是有效的。假如你的 Apple 机有一个“自动启动只读存储器”的话，那么为光标移动而附加一个 ESC 特性也是可用的。编辑程序的输入方法扩展了标准的 Apple II 输入子程序。当你把控制字符送到输入字符行，并且再用左箭头“←”键把它们退格回来时，屏幕上的光标就会避免退回。

虽然这种方式允许你和控制字符加到你的文件中去，但是，因为控制字符在汇编语言中是非法的，你必须很小心地别让这些控制字符错送到汇编语言程序中去。

2)、COPY 复制

Copy Rlinenum 1 [—Olinenum 2] TO Rlinenum 3

该命令可以将一个或几个字符行复制到文件中的 linenum 3 (行号 3) 的前面，当使用任选的 linenum 2 时，则可复制字符行的范围是从 linenum 1 到 linenum 2，而如果去掉 linenum 2，那么，仅仅复制 linenum 1。复制命令是唯一的一个要求在参数列中加一个关键字 TO 的命令。正如在上面命令语法中用大写字母所指出的，字母 CO 是必需的，而后面两个字母 PY 是任选的。

注意：编辑程序中没有“Move”命令。（传送命令），字符行可以用“复制”的方法传送到新的位置，然后在原来的位置将它们删掉。要知道如果以 linenum 3 小于 linenum 1 来做 Copy，那么，当复制的行插入到原来的行前面时，原来的行数将发生改变。因此，如果

你在这之后，想删去这些原先的字符行，则你必须在删去它们之前先确定它们的新的字符行数。假如你以 `linenum 3` 大于 `linenum 1` 来拷贝的话，那么就不存在此问题，因为原先的字符行没有作任何移动。

当使用 `linenum 2` 时，那么它必须是大于或等于 `linenum 1`。

3)、CHANGE 更换

`Chang [Rangelist]Chgstring`

更换命令可以使你对在整个编辑文件中或是在由任选的 `Rangelist` 所指定的字符行中的老的字符行(一些或所有的出现之处)用一个新的字符串来更换它。如果你没有 `Rangelist`，那么编辑程序将会在整个文件内搜索所有老的字符串，并且用新的字符串来代替它。

`Chgstring` 的语句结构为：

`<分界符> [字符串] <分界符> [字符串] [<分界符>]`

这里，第一个字符串是老的字符串，而第二个字符串为新字符串。语句结构指出至少要含有两个分界符[<分界符>]。老的字符串不能是空，而新的字符串则可以是空，末尾的分界符可以省掉，因为 `RETURN` 可以代替它。<分界符>不能用命令中的字符。

更换命令常常用下面的字符行来提示：

`ALL OR SOME? (A/S)`

然后你可以选择：是由编辑程序来更换那些规定范围中的所有的字符行呢？还是通过提示来引进更换，或不进行任何更换。假如你用字母 `S` 或是用 `ASCII` 码的任何一个上档键来作出响应的話，那么，在屏幕上显出每个更换之后，编辑程序就停止，然后出现一个闪烁的光标，请求实现所指定的更换。

如果你按下 `ESC` 键或是除 `CTRL—C` 以外的任何一个其它的控制键，那么编辑程序将不进行更换操作，而搜索下一个老的字符串。`CTRL—C`，能勾消所有的更换命令，使你返回到命令方式。如果你按下空格键或者任何一个非控制键(除了 `RESET` 以外)，那么，编辑程序将会用新字符串替换文件中的老字符串并继续进行搜索。而假如你不想进行更换的话，那么，编辑程序继续搜索，虽然文件中的字符行不进行更换，但是编辑程序不把在屏幕上的字符行变回到它的原来形式。

在每一字符行完成其更换以后，更换命令不能从头开始重新对当前的字符行进行扫描，并且由更换命令刚加入的新字符也不扫描。这意味着，如果你在一个字符行中，例如：有10个“*”号，而你作一个 `/ ** /` 的更换，那么在把所有的 `/ ** /` 更换成 `/ */` 之后，该字符行只保留了5个“*”号了。

在特殊的场合，可利用“百搭”字符 `CTRL—A` 来代表更换字符串(`Chgstring`)中的部分旧字符串，它还表示任何一个字符除了“回车”外，与在更换字符串的旧字符串部分找到的 `CTRL—A` 字符相配，这样就提供了一种方法。即：可以把文件中的一组或一类字符串都更换成相同的新字符串。你不能在更换字符串的新字符串部分，使用“百搭”字符 `CTRL—A`，在那个位置上只能插入一个 `CTRL—A` 字符。

4)、DELTE 删去

`Delete [Rangelist]`

删去命令用来删去由 `Rangelist` 所指定的字符行。如果在 `Rangelist` 中有好几个范围，那么它们必须颠倒其次序(即：首先是行数范围最高的，依次下降)，当对好几个范围进行删

去, 如果不这样颠倒的话, 较低的范围的删除将会改变掉所有后面字符行的行号, 然后, 这些字符行被较高的字符行计数范围删掉时就会出错! 要记住, 每一次删去都会减小文件中所有随后的字符行有关的行号。

删除命令不能在没有 Raangelist 的情况下使用, 否则将导致错误, SYNTAX 错, 但是, Rangelist 不可以是 lineum 一的形式, 也就是说, 你必须指定 Rangelist 的起点和结束点。

5)、EDIT 编辑

Edit [Rangelist] [Dstring]

编辑命令是编辑程序中最万能的文本操作命令。它可以使你进入编辑方式, 在这种方式中, 你可以进行一次一个字符的处理。同输入方式一样, 编辑方式是与命令方式区分开的。编辑方式一次作用于一个字符行, 并且是一维的。也就是说, 在所编辑的字符行中, 你只能利用“—>”键作向前的移位或用“←—”键作朝后的移位。

编辑命令使系统转入到编辑方式并确定所编辑的字符行。如果可选择的 Rangelist 和可选择的分界编辑字符串 (Dstring) 均省去的话, 那么, 在文件中的所有字符行在编辑方式中依次出现。如果 Rangelist 和编辑字符串两者都使用的话, 那么只有那些列在 Rangelist 中的并且包含编辑字符串的字符行将在编辑过程中被选用和出现。假如省略编辑字符串, 那么, 在 Rangelist 中的所有的字符行都将被选用。

这样, 你可以知道, 假如只使用编辑字符串, 那么, 文件中包括编辑字符串的所有字符行都将为编辑过程所选用。编辑字符串可以含有“百搭”字符, 以允许配上任何字符 (参见 \$14 中更换命令的说明)。

当你处于编辑方式时, 你可以向前, 朝后地移动光标, 也可以替换, 插入和删掉一些字符。在屏幕上的字符行将很快地显示出所有的变化。这样, 就不会让你担心其结果如何。

编辑方式中, 所有这些方便性都是通过控制字符来完成的, 这些控制字符指挥编辑工作做些什么, 打印一个编辑方式的控制字符就能使它执行其相应的命令, 而任何其它的控制字符将以嘟嘟的响声而拒绝接收。若打入不是控制字符的其它字符, 则这些字符一对一地替换行上的字符。这些字符可能和已经在那里的字符相同, 因为有时重新打印一些字符比起用光标移动的办法更快一些。所编辑的字符行的长度不受原来的字符行的长度的限制, 可以比原来的字符行长一些或短一些。下面介绍如何使用编辑方式的功能。

首先, 建立编辑方式, 用编辑命令来确定所编辑文件行的范围, 该范围中的第一个字符行将被显示出来, 而光标就移到这一字符行的第一个字符位置上。

现在用箭头键把光标移到你想要改变的第一个字符或空格上, 如果光标后面的字符前后跳动的话, 你不需担忧, 这是由于空格用来作为制表字符 (tab) 的缘故。(除非你改变了制表字符), 不要用空格键来移动光标, 这样做就使得用空格来代替字符了。

为了替换闪烁的字符, 只需简单地打入一个新字符。此新字符将替换掉闪烁的字符而不影响任何其它的字符, 而光标将移到下一个字符上, 如果你继续打入, 则更多的字符被替换掉。

为了插入一个字符 (插在闪烁的字符的前面), 打入一个 CTRL—I 那么该字符就被插入, 它将出现在闪烁字符的前面, 而闪烁的字符则右移一格, 被插入的字符就填进右移后产生的空档里。如果你继续打入的话, 则可连续地插入许多字符。要想结束这些插入, 只需按一下箭头键或打一个除了 CTRL—V 以外的任何一个控制符即可。

为了删掉那个闪烁的字符, 可以打入 CTRL—D。那么此闪烁的字符就会消失掉, 而它

右边那个字符则移到光标底下填进所删去的那个位置。要想抹掉更多的字符，可多次打入 CTRL—D，或在 CTRL—D 下使用 REPT(重复)键。

为了恢复屏幕上的字符回到它原先的格式，需打入 CTRL—R，那么屏幕上的字符行由原先的拷贝所代替，这个拷贝是在文件中的未经改变过的字符行。当你已把字符行破坏了而想重新编辑它时，这种恢复性能就显得很有用了。在你还没有按下 RETURN 键的情况下恢复性能才有效，否则，新行已放入文件中了。

为了查找在光标之后某一个字符的出现之处，需打入 CTRL—F 再打入要找的字符，这样该个字符就可以被找到了。当该字符在当前的字符行中被找到的话，则光标就跳到这个字符上，否则，光标将停留在原先的位置上不动。

控制字符可能令人感到讨厌，当你不要用它时，它常常一闪一闪的，而当你最需要它们时，它都消失了。编辑程序有两种功能帮助你控制字符进行控制，这两种功能就是忽隐忽现(Popout)和原封不动(Verbatim)。

所谓“忽隐忽现”就是用闪烁的忽明忽暗的方式来显示文件中的控制字符，通常来讲，屏幕上的控制字符是看不到的，但“忽隐忽现”的功能却可以使你每次一个地检查它们。在编辑工作方式中，如果你在字符行中向前或者退后地移动光标，那么，任何控制字符在你面前忽隐忽现地显示出来，而字符行的其余部分被推到两旁。如果你不能确定这个字符究竟是否是控制字符(因为光标也在闪烁)，那么你只要简单地移走光标，如果该字符也跟着移走，那么它就是一个控制字符。

“原封不动”的性能或 CTRL—V 可以使你能够把控制字符以替换或插入的方式放入文件中去。(通常，编辑工作方式用嘟嘟叫的响声拒收控制字符。)为了用一个控制字符来代替一个字符，可以把光标移到这个字符上。(如果光标已经正好位于那个字符的前面，并且刚刚在它前面插入了一个字符，那么你可以用箭头键来回移动光标，使插入取消掉。)然后打印 CTRL—V 让你所希望的字符进入。如果你想用控制字符替换几个字符，那么在每次替换之前都需打印一下 CTRL—V。

为了在光标前面插入一个控制字符，必须先打入 CTRL—I，然后再打入 CTRL—V，最后再打印该控制字符。如果你想插入一些控制字符，那么在插入每一个控制字符之前，都必须打入 CTRL—V 才行。当完成上面的工作之后，你可以继续停留在插入方式，也可以返回到替换方式去，如上描述的那样。

如果你完成了你所要做的所有的更换工作，或者你决定不做任何别的工作了，那么用下列三种方法的任何一种就可以离开编辑方式。

第一种方法：

在字符行的任何地方打入一个 CTRL—X，它彻底取消编辑方式，即使在 Rangelist 中的所有行还没有处理，也返回命令方式，把当前行保留为原先的形式。即便在屏幕上可能已经作了一些更改，但文件中的行均不改变，否则须使用其它两种跳出方式。

为了把经变化过的字符行放进文件中去并退出编辑方法，则按一下 RETURN 键。这样就显示在屏幕上的字符行放到文件中代替了原先的字符行，除非这样做，否则文件中的字符行是不会有变化的。当使用 RETURN 时，不论光标处于文件的任何位置上，所有的字符行都放进文件中去了。

为了想把字符行中的最前面的那部分放进文件中去，只需把光标放到你想保留的最后一

个字符的位置上, 然后打入 CTRL-T, 这样就会抹掉光标右边的所有的字符, 并将缩短了
的字符行放进文件中去。

一旦编辑已经由 RETURN 或 CTRL-T 而结束, 然后如果另一字符行可以用来进行编
辑, 那么该字符行将会被显示出来, 且光标处于该字符行的第一个字符上。而如果没有选出
的字符行, 则编辑过程自动结束而回返到命令方式上。同时, 带有满意的编辑命令条件。

编辑方式中的控制字符

作 用

RETURN

接收显示在屏幕上的字符行, 并用它替换文件中的旧字
符。

CTRL-X

概取消编辑方式, 不替换文件中现有的字符行。

CTRL-T

以当前光标的位置, 截短字符行并替换文件中的旧字
符行。

CTRL-R

用仍然在文件中的原先的字符行替换尚未被接收编辑的
字符行。

CTRL-D

删掉当前光标位置的字符, 该字符行缩短一个字符。

CTRL-I

进入字符插入方式, 字符被插入到当前光标位置的前面。

“→”键

使光标在字符行中向前推进一格。

“←”键

使光标在字符行中退后一格。

CTRL-V 字符

根据当前字符的方式, 进行插入或替换, 将字符原封不动
地放入到文件中。字符被插入之前不要打“空格”
键。

CTRL-F 字符

从当前光标位置的右边开始查找给出的字符的出现之
处。在该字符被找到之前不要打“空格”键。

6)、FIND 查找

Find [Rangelist] [Dstring]

查找命令对 Rangelist 的字符行中含有 Dstring 的所有的字符进行定位和列出来, 如果
没有指定 Rangelist, 则整个编辑文件均被查阅, 如果没有给出 Dstring, 则在Ranglist中的
所有字符行均被列出, 而如果两者都没有指定, 则文件中的所有字符行都列出来。“百搭”字
搭 (CTRL-A) 可以被用在查找命令的 Dstring 中, 关于详细的说明, 可参阅更换命令的
说明。

不论字符行中出现多少个 Dstring, 查找命令一次只能对一个字符行进行列表。如果没
有找到 Dstring 或者在 Rangelist 中没有行存在, 系统就由“查找”返回到命令方式。

7)、INSERT 插入

Instrt Rlinenum

当执行插入命令时, 系统就进入输入方式, 并且把所有新的字符行插入到给定的行号前
面。输入方式已在 Add 命令中描述过了, |Insert| 表示在文件的第一个字符行的前面 将 插
进一个新建行, 打入 CTRL-D 或 CTRL-Q 就可结束输入方式。插入一些字符行后, 会使其
后的所有的字符行的行号增加, 因此, 对这些字符行执行任何处理之前, 必须检测一下新的
行号。