家用电脑应用软件100例

家用电脑应用软件100例

JIAYONGDIANNAO
YINGYONGRUANJIAN100LI

雷方桂
吴耀斌　编著
向期中
赵显富　审定

中南工业大学出版社

[湘] 新登字 010 号

家用电脑应用软件 100 例

雷芳桂 等编著

责任编辑：肖梓高

*

中南工业大学出版社出版发行

中南工业大学出版社印刷厂印装

新华书店总店北京发行所经销

*

开本：787× 1092 1/16 印张：11.25 字数：267 千字

1994 年 8 月第 1 版 1994 年 8 月第 1 次印刷

印数：0001—6000

*

ISBN 7-81020-674-5/TP · 039

定价：7.80 元

前　言

随着计算机的普及应用和人们生活水平的不断提高，家用电脑正作为一种热门的家用电器进入寻常百姓家庭，然而家用电脑的操作又不同于一般家电的简单使用，其功能随着操作者的熟练使用程度和知识水平的不断提高而增强，其作用和应用范围也不断增大。为克服目前家用电脑停留在玩游戏等简单使用的状况，满足广大家庭和青少年学习电脑的热切要求，挖掘和开发青少年的智力水平和潜力，我们编写了此书。同时，此书也可以作为电脑爱好者的一本较全面的入门参考书和手册，也是非计算机专门人员使用电脑的一本工具书。

本书着重介绍目前能使用在家用电脑上的常见软件的使用方法和操作过程，包括如下几个主要方面：操作系统类软件（EDLIN、DEBUG 等）；工具类软件（PCTOOLS、SOURCE、DM 等）；辅助教学类软件（芸心、创意、龙碟等系列）；文字处理软件（CWS、WPS、SPT 等）；数据库管理软件（SPDMS、OFFICE、FOXBASE 等）；病毒处理软件（CPAV、SCAN、KILL 等）。这些软件在家用电脑上的应用，将使其功能和应用跃上一个新的台阶。

本书面向广大家庭和青少年，力求由浅入深、循序渐进地指导未接触电脑的操作者学习电脑基本知识及基本操作方法。

参加本书编写的还有王建新、吴迪文、牛丽娜等同志。

由于作者的水平和时间有限，书中错误在所难免，恳请广大读者批评指正。

编者

1994 年 4 月

目　录

1　操作系统类软件

1.1　行编辑（EDLIN）

用户可以使用行编辑程序（EDLIN）来建立、修改以及显示源文件或文本文件。源文件是采用源语言格式的未汇编程序。文本文件采用易读的格式。

EDLIN是一个行文本编辑程序，它可以用于:

(1) 建立一个新的源文件并保存之。

(2) 更新现有的文件并且保存更新过的文件以及原始的文件。

(3) 删除、编辑、插入及显示行。

(4) 在一或更多的行检索、删除或替换文本。

由EDLIN建立或编辑的文本文件被分成不同长度的行，每行至多有253个字符。

在编辑过程中，EDLIN动态地产生并显示行号，但在所保存的文件里实际上出现行号。当用户插入行，所有跟在插入文本之后的行号根据插行的数目自动地增加。当用户删除行时，所有跟在删除文本之后的行号根据删除行的数目自动地减少。因而，行号总是连贯地从1至最后的行号。

(一) 如何启动EDLIN程序

启动EDLIN程序时，输入:

EDLIN　filespec

注意: 当用户启动EDLIN程序时，EDLIN擦除了文件的备用副本（.BAK），（如果其存在的话），以保证软磁盘上有足够的空间供更新的文件。然后EDLIN分配带有用户以EDLIN命令和扩展。￥￥￥指定文件名的新文件。带有扩展￥￥￥的文件包含更新的文件，由结束编辑命令最终重命名至指定的文件区。

(1) 如果指定的文件存在于指定的或默认的磁盘驱动动器上，该文件被装入存储器，直到存储器装满75%。如果整个文件被装入，则显示以下的信息和提示符:

End of input file

*_

然后用户可以编辑一个星号（*）。

(2) 如果整文件不能装入存储器，EDLIN装入各行，直到存储器装满75%为止，然后显示*提示符。其后用户可以编辑存储器里的文件部分。编辑文件剩余部分时，用户必须把一些编辑过的行写入软磁盘以便释放存储器，这样用户就能从软磁盘把未编辑过的行存入存储器。

(3) 如果指定的文件不存在于驱动器上，将以指定的名字建立一个新文件。显示下列信息和提示符:

New file

*—

这时通过输入所希望的文本行，用户可以建立一个新的文件。在输入文本之前，用户必须输入一个*代表插入行。

当用户完成编辑后，作用结束编辑命令便可以保存原始的和更新的文件。结束编辑命令在本章称为"EDLIN 命令"一节中详述，原始文件被重命名为扩展.BAK，新文件有用户以 EDLIN 命令指定的文件名和扩展名。

注意：用户不能用 EDLIN 命令编辑一个带有文件名扩展.BAK 的文件，因为系统假定其为备用文件。如果用户认为必须编辑这样的文件，把文件重命名为另一个扩展；然后开始 EDLIN 并指定新名称。

（二）EDLIN 命令参数

line　代表用户必须指定一个行号。用户可以采用三种可能的参数输入。

①输入一个从1至65529的十进制整数。如果用户指定一个比在内存行号中更更大的一个号，该行将被加在现有的最后一行之后。行号必须用一个逗号或空格彼此分开。

②输入一个星号（*）来指定内存中最后一行之后的行。输入一个*号与指定一个比内存中最大行号更大的号有相同的效果。

③输入一个句点（.）来指定当前的行。当前的行指出文件最后一次修改的位置，但不一定是所显示的最后一行。

n　代表用户必须指定行。输入用户想写入软磁盘或从软磁盘装入的行数。用户只能把该参数用于写入行和附加行命令。只有当将被编辑的文件太大而不能存入内存时，这些命令才有意义。

string　代表用户必须输入一个或多个符号来表示将要查找、被替换、被删除的文本或替换其它的文本。用户只能把该参数用于检索文本或替换文本的命令中。

（三）EDLIN 命令

本节叙述 EDLIN 命令并告诉用户怎样使用它们。命令是以字母表顺序列出，每个命令有其用途和格式。

EDLIN 命令的通用资料。

①除编辑行命令外，所有的命令都是单个字母。

②除结束和停止编辑命令之外，命令通常有参数前缀和／或后缀。

③用大写体或小写体，或两者的结合输入命令以及串参数。

④为了易读，用定界符隔开命令和参数；然而，定界符仅在相邻的行号之间用。记住，定界符为空格或逗号。

⑤只有当用户按下输入键之后命令才有效。

⑥结束命令要按下 Ctrl-Break 键。

⑦对于产生大量输出的命令，可按下 Ctrl-Num Lock 暂停显示，这样用户以在显示消失前看清楚。按下任何字符便可重新开始显示。

⑧当使用 EDLIN 命令时，可使用第一章里叙述的控制键和 DOS 编辑键。在一行之

内编辑时它们非常有用，而 EDLIN 命令可用于编辑所有的操作。

⑨EDLIN 的提示符是一个星号（*）。

(1) 附加行命令。

用途　把指定数目的行从软磁盘加到在内存中被编辑的文件上，这些行被加到内存中当前行的后面。

格式　〔n〕A

备注　如果被编辑的文件太大而不能存于内存中，该命令才有意义。当用户启动 EDLIN 命令时，尽可能多的行就被读入内存供编辑用。为了编辑文件中那些不能存于内存中的剩余部分，用户必须把内存中编辑的行写入软磁盘，然后才能使用附加行命令把未编辑的行从软磁盘装入存储器。请参考写入行命令便可获得怎样把编辑的行写入软磁盘的资料。

注意　①如果用户没有指定行的数目，把文本行读入内存直到有效的存储器装满 75%为止。如果有效的存储器已经装满 75%，便不会有任何动作。

②当附加行命令已经把文件的最后一行读入存储器时，将显示出End of input fille（输入文件结束）的信息。

(2) 删除行命令。

用途　删除一个指定范围内的行。

格式　〔line〕〔，line〕D

备注　即便被删除的范围包括内存中最后一行，后随被删除范围的行变为当前行。当前行及所有后随行被重新编号。如果一个或两个参数被省略将提供缺省值。

如果用户省略第一个参数，如:

，line D　则从当前行开始删除至由第二个参数指定的行结束。需要用一个逗号开始来代表被省略的第一个参数。

如果用户省略第二个参数，如:

line D 或 line，D　则只有一个指定的行被删除。

如果用户省略两个参数，如:

D　则只有当前行被删除，后随的那一行变为当前行。

(3) 编辑行命令。

用途　允许用户编辑文本的一个行。用户必须输入将要编辑的行的行号，或输入一个句点(.) 来指出当前行。

格式　〔line〕

备注　如果用户仅仅按下Enter键，则规定当前行之后的一行将被编辑。行号及其文本被显示出并且行号在下面的行上重复。

(4) 结束编辑命令。

用途　结束EDLIN并保存编辑过的文件。

格式　E

备注　把编辑过的文件通过写命令使它保存到用户在启动EDLIN命令指定的驱动器和文件名中。原始文件（即当启动 EDLIN 时指定的文件）可以重命名，给它一个.BAK 文件扩展名。如果没有原始文件（即: 如果用户建立一个新文件而

不是在编辑期间更新一个旧文件的话）将不会产生.BAK 文件。

注意　用户要确认软磁盘有足够的自由空间来保存整个文件。如果用户的软磁盘没有足够的自由空间，只有一部分被保存。内存中未写入软磁盘的部分丢失。

(5) 插入行命令。

用途　紧靠在指定行之前插入文本行。

格式　〔line〕I

备注　如果用户不指定行，或以一个句号（.）的形式指定行，插入将在紧靠当前行之前进行。如果用户指定的行号大于现有的最大行号，或者指定#作为行号，则插入将在内存中最后一行之后进行。EDLIN 显示适当的行号以便用户能够输入更多的行，结束每一个行时按下 Enter 键。在操作插入方式期间，每次按下 Enter 时，连续的行号自动地出现。用户必须按下 Ctrl-Break 键来中断插入方式操作，即使插入的行被加到内存中的最后一行后随插入行的那一行变为当前行。当前行和所有保留的行被重新编号。当用户建立一个新文件时，用户必须输入插入行命令后才能插入文本。

(6) 列表行命令。

用途　显示一个指定范围内的行。当前行保持不变。

格式　〔line〕〔，line〕L

备注　如果一个或两个参数被省略将提供缺省值。

如果用户省略第一个参数，如:

，line L　则从当前行的前11行开始显示并于指定的行结束显示。需要一个逗号开始来代表省略的第一个参数。如果指定的行比当前行之前的 11 行还多，则显示与用户省略两个参数时的显示一样。

如果用户省略第二个参数，如:

line L 或 line，L　从指定的行开始，总共显示出23行。

如果用户省略两个参数，如:

L　总共显示出23行。当行之前11个行，当前行之后11个行。如果当前行之前没有11个行，则当前行之后额外的行被显示出，使总数达到 23 个行。

(7) 退出编辑命令。

用途　停止编辑，不保存用户当前输入的任何更改内容。

格式　Q

备注　EDLIN提醒用户确信不想保存更改部分。如果用户想要停止编辑，输入Y，这样就不保存编辑的更改部分也不产生.BAK 文件。有关.BAK 文件的资料请参考结束编辑命令。如果用户想要继续编辑，输入 N 或任何其它字符。

注意　开始执行时，EDLIN擦除文件（filename.BAK）的任何原先备用副本，以便留出空间来保存新的副本。所以，如果用户对 Abort edit（Y／N)?（是否停止编辑?）信息的回答为 Y（是），则用户原先备用副本将不复存在。

例如: Q

Abort edit（Y／N)?

(8) 替换文本命令。

用途　用第二个信息串替换指定范围行里出现的第一个信息串。如果用户省略了第二个信息串，替换文本命令将在指定范围的行里删除所出现的所有第一个信息串。每当发生更改时将显示出更改的行。最后更改的行变为当前行。

格式　〔line〕〔，line〕〔?〕 Rstring〔＜F6＞string〕

备注　每当一个修改的行显示后，用户可以指定一个任选参数? 来请示提示符(O.K.?)。如果用户想要保留修改，按下 Y 或 Enter 键。如果用户不需要修改，按下任何其它字符。不论那一种情况，都将在一定范围的行里检索再次出现的第一个信息串，包括在同一行里多次出现的该信息串。如果任一个或两个参数都丢失则产生缺省值。如果用户省略了第一个行参数，检索将从行 1 开始。如果用户省略了第二个行参数，检索于内存中最后一行结束。如果用户省略了两个行参数，系统将在内存中所有的行中检索出现的第一个信息串。

注意　第一个信息串从接R后面位置的字符开始，并继续直到用户按下F6或Ctrl-Z为止（或者当第二个信息串被省略时按下输入键)。如果用户省略了第一个信息串，则不会进行检索，这样命令立即结束并且示出 Not found（未找到）的信息。用户按下 F6 或 Ctrl-Z 之后第二个信息串立即开始，并继续直到用户按下 Enter 键为止。

(9) 检索文本命令。

用途　检索指定范围的一些文本以便给一个指定的信息串定位。

格式　〔line〕〔，line〕〔?〕 S　string

备注　包含指定信息串的第一个行被显示出并且结束检索（除非用户使用? 参数)。所找到的与指定信息串相称的第一个行变为当前行。

注意　检索命令总是在文本中检索完全相同的字符。即，如果用户输入大写体，它就检索大写体，如果用户输入小写体则检索小写体。每当显示包含指定信息串的行之后，如果用户要一个提示符（O.K.?)，用户必须指定任选参数?。如果指定的信息串未找到，检索结束并显示出信息 Not found（未找到)。当前行保持不变。如果用户输入 Y 或按下输入键，则与指定信息串相称的行变为当前行并结束检索。输入任何其它字符来继续检索直到另一个信息串找到为止，或直到该范围内所有的行都被为止。一但该范围内所有的行被检索，则显示出 Not found（未找到）的信息。如果用户省略了第一、第二或两个行参数，系统将提供缺省值。如果用户省略了第一个行参数，系统默认为行 1。如果用户省略了第二个行参数，系统默认用户文件里的最后一行。如果用户省略两个行参数，系统将检索内存中所有的行。

信息串从紧跟S之后位置的字符开始，并继续直到用户按下输入键结束检索为止。如果信息串参数被省略，则不进行检索，这样随着 Not found（未找到）的信息，命令立即结束。

(10) 写入行命令。

用途　把内存中正被编辑的行按指定行数写入软磁盘。从行号1开始写入各行。

格式　〔n〕 W

备注　只有当用户正在编辑的文件太大而不能存于内存中时，这个命令才有意义。当

用户开始 EDLIN 命令时，EDLIN 把行读入存储器装满 75%为止。为了编辑用户文件中的剩余部分，用户必须把内存中编辑过的行写入软磁盘，然后才能使用附加行命令从软磁盘里剩下的未编辑过的行装入内存。

注意　如果用户不指定行的数目，文本行将被写入直到有效存储器的25%被使用为止。如果有效存储器已经小于供使用的 25%时，则不会动作。所有的行重新编号，因此第一个保留的行变为行 1。

（四）EDLIN 命令概要

EDLIN 命令如表 1-1 所示。

表 1-1　　EDLIN 命令表

命　令	格　式
附加行	〔n〕A
删除行	〔line〕〔，line〕D
编辑行	〔line〕
结束编辑	E
插入行	〔line〕I
列表行	〔line〕〔，line〕L
退出编辑	Q
替换文本	〔line〕〔，line〕〔?〕R string〔<F6>string〕
检索文本	〔line〕〔，line〕〔?〕S string
写入行	〔n〕W

1.2　连接程序（LINK）

连接程序（LINK）的作用是:

（1）组合独立产生的目标模块。

（2）检索库文件以便取得未加外部引用的定义。

（3）分析外部相互对照引用。

（4）产生一个可打印的列表，该列表指出外部引用和错误信息的判定。

（5）产生一个浮动装入模块。

（一）文件

连接程序有下面的输入、输出及临时文件。

（1）输入文件。输入文件如表 1-2 所示。

表 1-2　　LINK 使用的输入文件

类型	缺省执行	优先执行	由…产生
目标文件	.OBJ	是	编译或汇编程序
库文件	.LIB	是	编译程序
自动响应程序	(无)	N/A	用户

（2）输出文件。输出文件如表 1-3 所示。

表 1-3　　LINK 使用的输出文件

类型	缺省执行	优先执行	使用前
列表文件	.MAP	是	用户
运行文件	.EXE	否	COMMAND.COM 程序

（3）VM.TMP（临时文件）。连接程序使用尽可能多的存储器来保存给正产生的装入

模块定义的数据。如果模块太大，不能在用的内存空间中处理，连接程序可能需要附加的存储空间。如果情况是这样的话，则在 DOS 缺省驱动器上产生称为 VM.TMP 的临时软磁盘文件。

当到软磁盘的溢出开始时，连接程序显示下列的信息:

VM.TMP has been created

Do not change diskette in drive A

一旦该临时文件产生，在连接程序结束之前用户不可移动软磁盘。当连接程序结束时，它就擦除了 VM.TMP 文件。

如果 DOS 缺省驱动器已经有了一个称作 VM.TMP 的文件，该文件将被连接程序删除而分配一个新的文件；原先文件的内容消失。因此，用户应该避免作用 VM.TMP 作为自己的一个文件名。

（二）定义

段、组、类是本节出现的术语。这些术语说明了连接程序的基本功能。弄清楚这些术语定义的概念就能使用户基本上懂得连接程序的工作方式。

（1）段。段是存储器的一片连续的区域，长度达 64K 字节。一个段可以定位于存储器任何地方的一节（16 字节）边界上。四个寄存器中的任一个可以规定一个段。段可以重迭。每一个 16 位地址是一个段始点的偏移。段的内容由一个段寄存器／偏移量寄存器对编址的。

当机器语言产生时便决定了段里不同部分的内容。其规模和位置并不一定由机具语言发生器确定，因为段里的这个部分在连接时可以与其它部分结合而形成一个单段。

一个程序在内存中的最终位置是由 COMMAND.COM 里提供的重定位装入该程序时根据用户是否指定／HIGH 参数决定。／HIGH 参数在本节后面部分讨论。

（2）组。组是在存储器的一个 64K 字节区段中各个段之集合。汇编程序或编译程序把这些段命名为一个组。一个程序可包含一个或多个组。

组用来对内存进行段编址。组内各段的不同部分由一个段基本指针加上一个偏移量来编址。连接程序检查一个组的目标模块是否符合 64K 字节的限制。

（3）类。类是段的集合。把一些段命名为一个类影响了内存中段的次序和相应的布局。类的名称由汇编编译程序指定，指定为同一类名的所有部分一个接一个地被装入存储器。在一个类中各段按连接程序在目标文件中所遇到这些段的顺序排列。只有在输入到连接程序时第一类的一个段优于第二类中所有的段情况下，存储器里一个类优于另一个。类的大小并不限制。类可以分成组以便编址。

（三）命令提示符

当用户开始运行连接程序时，将接收到一个包含 4 个提示符的系列。用户可以从键盘对这些提示符作出响应，可以通过命令行对这些提示符作出响应，或者用户可以使用称为“自动响应文件”的一种特殊软磁盘文件来对提示符作出响应。

连接程序向用户提示目标文件、运行文件、列表文件和库文件的名称。当连接结束时，连接程序返回到 DOS，并且显示出 DOS 提示符。如果连接失败，连接程序将显示

出一个信息。

提示符按它们在屏幕上出现的次序加以说明。提示符后面的方括号里为缺省规则。在表中响应一栏，方括号表示任选输入。Object Modules 是仅有的要求用户响应的提示符，如表 1-4 所示。

表 1-4　　Object Modules 要求用户响应的提示符

提　示　符	反　　应
Object Modules〔.OBJ〕:	filespec〔+trilespec〕…
Run File〔filespec.EXE〕	〔filespec〕〔/p〕
List File〔NUL.MAP〕:	〔filespec〕
Libraries〔.LIB〕:	〔filespec〔+tfilespec〕…

注意　①如果用户输入一个文件而不指定驱动器的话，则采用缺省驱动器。库提示符是一个例外——库的缺省驱动器由编译程序决定。

②按下Ctrl-Break，用户可以在连接程序正常结束之前使之结束。

以下详细的说明包含用户可以对提示符输入的响应的资料。

(1) 目标模块〔.OBJ〕：为将被连拉的目标模块输入一个或多个文件说明。多个文件说明必须用一个加号（+）或括号隔开。如果省略任一文件名的扩展名，连接程序假设文件扩展名为.OBJ。如果一个目标模块有不同的文件扩展名，则必须指定扩展名。目标文件名可有用@符号开始（@被保留下用于自动响应文件）。

连接程序按照它所遇到的段的先后顺序把这些段装到类中。

如果用户指定一个目标模块，但连接程序没有查到该文件，则显示出下列的提示符:

Cannot find file filespec

Change diskette (hit Enter)

用户必须插入包含所请求模块的软磁盘，这就允许有关的一些软磁盘的.OBJ 文件。在单驱动器系统中，只有当 VM.TMP 未被打开时才能安全地进行软磁盘交换。

如果 VM.TMP 文件已经被打开，用户不能把包含 VM.TMP 文件的软磁盘移去。记住，一旦一个 VM.TMP 文件被打开，不能移动存有该文件的软磁盘。

打开一个 VM.TMP 文件之后，如果用户在存有 VM.TMP 的同驱动器上指定一个目标模块，而连接程序找不到它时，连接程序结束并以下信息:

Fatal error

Cannot find file filespec

(2) 运行文件〔filespec.EXE〕。建立用户输入的指定文件来贮存连接程序形成的运行（可执行）文件。尽管用户指定其它扩展名，所有的运行文件只接受文件扩展名.EXE。如果用户指定另一个扩展名，所指定的扩展名将被忽略。

运行文件提示符的缺省文件名是所指定的目标模块中的第一个文件名。

(3) 列表文件〔NUL.MAP〕。连接程序列表文件有时称作连接程序图。除非用户特别需要，否则将不产生列表文件。用户可以用一个文件名或一个设备名来取代缺省值便可请求列表文件。若用户没有包括一个文件扩展名，则使用缺省扩展名.MAP。如果用户不

输入一个文件说明，DOS保留的文件名NUL则规定不产生列表文件。

列表文件包含输入（目标）模块里各个段的入口。各个入口在运行文件里指出了偏移（偏址）量。

注意　如果列表文件指定为软盘的一个文件，则连接程序结束之前不可以移动软盘。

如果用户为列表文件指定的驱动与所指定的目标模块的驱动器相同，并且连接程序找不到它，这时连接程序结束并显示以下信息:

Fatal error

Cannot find file fiiespec

为避免在一个软磁盘里产生列表文件，用户可指定显示器或打印机为列表文件设备。

例如: List File〔NUL MAP〕: CON

如果用户把它直接输出到显示器，按下Ctrl-prtsc亦可打印输出。

(4) 文件库〔.LIB〕。用户可以给自己的文件库指定文件名，或仅仅按下输入键。如果用户按下输入键，连接程序默认为供作编译程序包一部分的文件库。编译程序包还提供库的位置。对于连接从宏汇编程序来的目标，并无自动缺省库检索。

如果用户响应库提示符，用户要指定一个驱动器ID表以及由加号（+）或空隔分开的文件说明。用户可以输入1至8个库文件说明。驱动器ID告诉连接程序何处寻找库提示符上的所有顺序库。自动检索文件名在理论上被置于对库提示符响应的末端。

连接程序按库文件被列出作外部引用分析的次序检索库文件。当连接程序找到规定外部符号的模块时，该模块被处理为另一个目标模块。

如果两个或多个文件库有相同的文件名，则不论其位置，仅检索表上第一个库。当连接程序找不到一个库文件时，它将显示出以下信息:

Cannot find library A: library. LIB

Enter new drive letter:

必须打入所指文件库存在的那个驱动器。

可以使用下列的库提示符响应:

Libraries〔.LIB〕: B:

在驱动器B寻找编译程序.LIB。

Libraries〔.LIB〕: B: USERLIB

在驱动器B寻找USERLIB并在驱动器A寻找编译程序.LIB。

Libraries〔.LIB〕: A: LIB_1+LIB_2+B: LIB_3+A:

在驱动器A寻找LIB1.LIB和LIB2.LIB，在驱动器B寻拭LIB3.LIB，在驱动器A寻找编译程序.LIB。

(5) 连接程序参数。在四个连接程序提示符任一个的末端，用户可以指定一个或多个参数来命令连接程序作不同的工作。只需要/符号和任何参数的第一个字母。

①/DSALLOCATION。/DSALLOCATION（/D）参数使得连接程序把定义在DGROUP里的所有数据装到组的高端。如果指定/HIGH参数（模块装入高端），这就允许任何可用空间能属于已分配的特殊区中，该区是在由用户应用动态分法分配的DGROUP中并且仍然可以由同一数据空间指针寻址。

注意　可用来动态分配的最大存贮量为64K（或实际上用的存贮量）减去DGROUP

分配的部分。

如果不指定／DSALLOCATION 参数，则连接程序把定义在组名为 DGROUP 的组里所有数装入组的低端，且开始时位移量为 0。因而数据空间指针引用的存储只是定义在组里的空间。

除 DGROUP 之外，所有的 GROUP 里任何类型的其它段被装入它们各自的组的低端，就象没有指定／DSALLOCATION 参数一样。

②／HIGH。／HIGH（／H）参数促使装入程序把运行空间映象为尽可能高的内存区中。如果用户指定／HIGH 参数，即告诉连接程序使得装入程序把运行文件装到尽可能高的内存区，但不能复盖 COMMAND.COM 的暂时部分，该部分在装入时占据存储器的最高区域。如果用户不指定／HIGH 参数，连接程序将使得装入程序把运行文件装到内存中尽可能低的内存区中。

／HIGH 参数与 DSALLOCATION 参数一起使用。

③／LINE。对于某些 IBM 个人计算机语言处理器来说，／LINK（／L）参数使得连接程序在 LIST 文件中必须包含输入模块中的源语行号和地址。

④／MAP。／MAP（／M）参数使得连接程序列出在输入模块里指定的所有公共（全局的）符号。连接程序列出每一个符号的值以及它在运行文件中的段偏移位置。符号列在表文件的末端。

⑤／PAUSE。／PAUSE（／P）参数告诉连接程序给用户显示以下信息:

About to generate .EXE file

Change disks (hit Enter)

该信息允许用户插入包含运行文件的软磁盘。

⑥／STACK: size。SIZE 项为 65536 字节之内任何正十进制数值。该数值用来取代宏汇编程序或编程序为正产生的装入模块所提供的堆栈容量。如果用户指定大于 0 而小于 512 的一个值，则采用 512。

如果用户不指定／STACK（／S），则使用宏汇编程序或编译程序提供的原始堆栈容量。如果堆栈的容量太小，则在运行装入模块时将产生不可预料的结果。

至少有一个输入（目标）模块必须包含一个堆栈分配语句。这是由编译程序自动提供的。对于宏汇编程序，源程序必须包括含有 STACK 语句，连接程序则包括含有与 STACK 组合形式的 SEGMENT 命令。如果不提供堆栈分配语句，连接程序则返回一个警告: NO Stack Statement（无堆栈语句）信息。

（四）如何启动链接程序

（1）开始启动前。

• 要确保将要用于链接的文件已存于相应的磁盘上。

• 确保用的磁盘有足够的存贮空间接受的文件和所产生的任何数据。

使用下列三个选择者之一，可以启动链接程序。

（2）选择项 1——控制台应答。

从键盘上，打入“LINK（回车）”。

这样链接程序就被装入到存贮器里并且一连显示四个提示符，这时，对这些提示符必

须打入所要求的应答。

如果打入了一个错误的应答，如：拼写错误的文件说明，就须按 Ctrl–Break 键退出 LINK 状态，然后再重新启动 LINK。如果仅输入了错误的应答还未按下 Enter 键，就可以删除掉错误的字符（仅限于在本行内操作）。

一旦打入完最后一个文件名，链接程序就开运行了。如果链接程序发现了什么错误，它就将错误显示在屏幕上并列出清单式文件。

注意　在任何一个应答打入后，而在按 Enter 键前，都可以用逗号连续打入应答去回答下一个提示符，而不必等待那些提示符出现在可以应答它时所在的位置上。如果用分号；来结束任何操作，以后的应答都假设为是失败的。链接程序不再给出什么提示符就立即开始处理了。

(3) 选择项 2——命令行。

从键盘上，打入"LINK objlist, runfile, mapfile,liblist〔parm〕...;"

objlist 是由空格或正号+分隔开的一列目标模块。

runfile 是想给运行文件的名字。

mapfile 是想给链接程序图的名字。

liblist 是一列被使用的程序库，它们由正号+或空格分隔。

parm 是一个可选择的链接程序参数。每个参数都必须用斜扛／开头。

链接程序装配后立即就开始执行由命令指定的任务。

当使用这命令行时，如果规定一个项用于所有的四个文件或者如果令行以分号；结束，那么就不会显示出选择项 1 中描述的提示符。

如果没有给出一完整的行并且改用分号，链接程序将提示余下的未说明的文件。但决不会提示参数，尽管它可能加至命令行的末端或加到为响应提示符所给出的任何文件说明中去。每次提示都会显示他的错误，这些错误都可靠 Enter 键接收或用显示文件名或设备名取消。然而，如果给出了一个完整的行并且命令行用分号结束，就不会再进一步提示未定义文件的错误。

这种命令行可允许有一定的变化格式。例如:

LENK 模块——这目标块是 MODULE.OBJ。给出一提示符，则表示 MODULE.EXE 中有错误。打入对它的应答后，给出一提示符表示 NUL.MAP 错误。给定应答后，显示一个提示符表示.LIB 扩展名错误。

LINK 模块；——如果加上一个分号，则不会再显示任何提示符。链接 MODULE.OBJ 目标模块后，运行的文件就转成 MODULE.EXE 并且不产生任何清单式文件。

(4) 选择项 3——自动应答。

到后来使用连接程序时，省掉应答这操作，常常是非常便利的。这在需说明一长列目标模块时是特别有用的。

作用该选择项前，必须创建一个自动应答文件。它包含几行文本，每行文本都应答于一个链接程序提示符。这些应答必须与这节前面部分讨论过的链接程序中出现的提示符有相同的顺序。如果需要，可使用正号+连续对下面的行进行相同的应答从而可将对目标模块或程序库提示的长长的应答包括在几行里。为了说明一个自动应答文件，要打入一个以

@符号为首的文件说明代替提示符应答或提示符应答的一部分。由磁盘文件内容给出对提示符的回答。文件说明不能是个被保留的 DOS 文件名。

从键盘上，打入:“LINK @ 文件说明（回车)”。文件扩展名可选择地使用。不存在什么错误的文件扩展名。

使用这选择项，可允许启动 LINK 的命令从键盘上打入或不需要做什么应答随一组文件一起打入。

注意 ①在RESPI头一行末尾的正号使头二行列出的模块被认为是输入目标模块。读入 RESPI 后，链接程序返回到命令行去识别+mymod，这样也就在目标模块清单里包括了 MYMOD.OBJ。

②在按了ENTER键后，上面的每一行结束。

1.3 动态调试程序（DEBUG)

DEBUG 程序是软件设计中常用的程序调试工具，它不仅能跟踪执行程序的运行踪迹，而且能直接与磁盘的指定扇区对话，以便读写磁盘文件某个扇区的具体内容，为分析或修改磁盘文件提供了方便。此外，DEBUG 在加密解密等方面有广泛的应用。

(一) DEBUG 程序启动

DEBUG 是 DOS 系统提供给用户的一个外部命令，因而启动该命令时在当前路径或 PATH 指令路径目录中必须含有 DEBUG.COM 文件。DEBUG 程序启动命令格式为:

DEBUG [d:] [path] [filename[.ext]] [parm1] [parm2]

其中 d: 和path 指定filename.ext所在的驱动器名及路径名; filename.ext 表示被调试程序的文件名及扩展名; parm1 parm2 表示文件说明的任选参量。

例如: DEBUG DISKCOPY.COM A: B:

DEBUG 程序启动成功后，屏幕显示 DEBUG 提示操作符“—”，说明系统已在 DEBUG 程序的管理之下，等待用户键入命令。此时，寄存器和标志位设置如下:

段寄存器（CS、DS、ES和SS） 设在可用内存的底端，即接在DEBUG程序尾部的第一段。

指令指针（IP） 设在设 0100H 处。

堆城指针（SP） 设在段尾或装配程序临时区的底部，取两者中较低的一个。

其余寄存器（AX, BX, CX, BP, SI 和 DI) 都设置为零。

如果起动 DEBUG 程序时带有文件说明，则 BX 和 CX 为以字节计的文件长度。

标志位的初始状态为: NV UP EI PL NE NA PO NC。

(二) DEBUG 命令参量（数）说明

(1) address（地址)。按下列格式输入其中的一部分或两部分。

①段寄存器: 偏移值;

②段地址: 偏移值;

③偏移值（此时使用默认段)。