IBM PC
宏汇编和维修手册

MACRO Assembler and Service Manual

厦门电子计算机厂
中国科学院计算所八室

目　　录

第一部份　　宏 汇 编

第一章　引言……（1）
第二章　宏汇编……（2）
第三章　索引程序……（7）
第四章　汇编语言格式……（10）
第五章　伪操作……（20）
第六章　指令助记符……（50）
附录　错误信息……（118）

第二部份　　维修手册……

李 盛 碧译
何 玉 珍校

*　　*　　*　　*

“宏汇编”部份由厦门计算机厂李巧珍、秦卫民、林珍、高询四同志译校，最后请厦门大学计算机科学系软件教研室洪岷生讲师审校，並作了修改，特此致谢！

编 者　1984．3于鼓浪屿

第一章　引　言

目录：

机器指令……（1）
汇编指令……（1）
宏指令……（2）
伪操作……（2）
EDLIN 程序……（2）

IBM个人计算机的宏汇编程序将你用汇编语言编写的源码翻译成一个目标模块，这个目标模块是由计算机可以识别的机器语言组成的。

你可以使用小汇编（Small Assembler），ASM（64K），或者使用宏汇编（Macro Assembler），MASM（96K）。如果你有96K内存，那么你可以使用两个汇编程序中的任一个，但是我们建议你使用宏汇编程序，MASM，它支持这本手册中所描述的全部功能。小汇编程序不支持宏汇编以及与之相关的功能：REPT，IRP，IRPC，也不支持STRUC和RECORD伪操作。

宏汇编程序为你提供如下功能：

- 检查和编排（documenting）源程序。
- 产生宏汇编指令。
- 将目标程序从原来指定的地址重新定位到另一个地址。
- 测定源程序的错误。
- 产生源程序语句的列表，并且对每个要汇编的源程序产生一个目标程序。

你所编写的源程序在汇编过程中不被执行，而仅被翻译成机器语言。

宏汇编程序是在磁盘操作系统支持下运行的，《IBM个人计算机磁盘操作系统(DOS)》手册中对该操作系统作了详细的描述。

你所使用的汇编语言是一种符号语言，这个符号语言中的每个操作码是由容易记忆的符号字符组成的，这些符号字符称为助记符。汇编语言在格式和内容上类似于机器语言，它由表示指令和注解的语句组成。指令语句是语言的工作部分，并且可以划分成下列几组：

- 机器指令
- 汇编指令
- 宏指令
- 伪操作

机器指令

机器指令是硬件指令在汇编指令集中的符号表示。

汇编指令

汇编指令是在汇编源程序模块期间，请求汇编程序执行特定的操作。

宏指令

宏指令要求汇编程序处理预先定义的代码序列。从这个序列中汇编程序产生源指令，然后这些源指令就如同一开始就是源模块中的输入的一部分那样被加以处理。

伪操作

伪操作，也称为pseudo—ops，它有与汇编指令助记符相似的助记符。它们告诉汇编程序如何处理数据、条件转移、宏操作和列表。伪操作通常不产生机器语言代码（见第5章对在宏汇编程序中使用伪操作的详细说明）。

EDLIN 程序

EDLIN程序可用于生成、更改和显示将要输入给汇编程序的源文件。

EDLIN程序是一种行编辑程序：

- 删除、编辑、插入和显示行。
- 搜索、删除、替换和显示正文。
- 产生新的文件并保存它们。
- 更改旧文件，并且保存更改前和更改后的两个文件。

关于EDLIN程序的详细描述请参阅《IBM个人计算机磁盘操作系统》手册。

第二章　宏 汇 编

目录：

命令格式……（3）
缺省的文件扩展名……（3）
怎样开始汇编程序……（3）
怎样开始汇编程序—选择方式1……（3）
怎样开始汇编程序—选择方式2……（5）
怎样开始汇编程序—选择方式3……（5）
参考（/parms）……（6）
设备标志……（7）

现在假定你已经用EDLIN程序（见《IBM个人计算机磁盘操作系统》手册）编写了一个汇编源程序，那么你就可以准备对这些程序进行汇编、连接、运行和调试排错。

如果汇编程序测试出源程序中的任何错误，那么这些错误就会显示在列表文件中。你应该用编辑程序修正错误，并且重新汇编你的源程序，以产生无错的机器语言。

连接程序（《IBM个人计算机磁盘操作系统》手册亦对此进行了详细的描述）将汇编程序所得到的机器语言（包括通过其它汇编运行所得到的机器语言），转换成可运行的程序。

所得到的程序最后可以准备进行装入、运行和调试排错。

命令格式

请你记住对本手册使用的命令、语句、函数和变量所做的以下规定：

1．用大写字母表示的字是关键字，并且必须按照说明进行输入。输入的字符可以是大写字符或小写字符。

2．任何用小写斜体字符表示的项是要求你提供的。

3．在方括号（〔〕）内的项是选项。

4．省略号（……）表示一个项可以重复任意多次。

5．除方括号外，所有的标点符号如逗号、括号、尖括号（＜　＞）、斜杠和分号都必须写在指定的地方。

缺省的文件扩展名

缺省的文件扩展名由一个小数点和三个字符组成（.×××）。它们是下列几个扩展名：

扩展名　　定义

.ASM　　表示源文件是汇编程序的文件扩展名。

.CRF　　表示是由CREF实用程序使用的索引文件（Cross reference file）的文件扩展名。

.LST　　表示是可打印的汇编程序列表文件的文件扩展名。

.OBJ　　表示可以重新定位的目标文件的文件扩展名。

.REF　　表示是一个可以打印的列表文件。这个列表文件包含了由CREF实用程序产生的索引表。

怎样开始使用汇编程序

利用下面所描述的三个选项之一，你可以开始使用宏汇编程序（MASM）或小汇编程序（ASM）。对三个选项的选择，取决于你系统本身的配置。

·如果你的系统只有单个软盘驱动器，那么选择方式1可以使你在使用汇编程序命令之前更换盘片。

·如果你的系统不只一个软盘驱动器，那么选择方式2可以让你立即使用汇编程序命令。

·选择方式3可以使你很方便地使用由磁盘操作系统提供的批处理命令。这个批处理命令为汇编程序命令的组合提供了很大的方便，以致使汇编程序能够自动进行工作。

怎样开始汇编程序—选择方式1

从你的键盘输入：ASM或MASM

注意：汇编程序的使用说明部分将会告诉你用符号〔M〕ASM选择哪一个汇编程序。

这时汇编程序就从软盘装入内存，过片刻，汇编程序将显示下列提示：

Source filename〔.ASM〕：

在你回答之前，可以将包含汇编程序的盘片移掉，而换上含有数据文件的盘片。

注意：1．在方括号内显示的名字是缺省的文件扩展名。这个文件扩展名是当你对自己的文件不给出文件扩展名时，由汇编程序作为扩展名使用的。

2．虽然汇编程序在你不给出文件扩展名时会提供一个缺省的文件扩展名，但这些扩展名都可以由明确指定的带有新扩展名的文件名所取代。

3．缺省的软盘驱动器是DOS缺省的驱动器，它可以由在文件标识符中加上驱动器ID来取代。

Source filename是你贮存程序的文件名字。例如，假定你用“myfile”来回答指示符，那么屏幕显示如下：

Source filename [.ASM] : myfile

在这里不必输入文件扩展名.ASM，因为汇编程序会自动寻找.ASM文件名。

在你输入源文件名后，你将看见下列提示：

Object filename [MYFILE.OBJ] :

Object filename是你要目标程序（机器可读）具有的文件名。如果你希望目标文件以MYFILE.OBJ名字贮存，则你只要按下Enter键即可；你也可以给文件另定一个名字，对于这个名字，汇编程序将自动加上文件扩展名.OBJ。（文件扩展名.OBJ可以由指定的另外一个扩展名来取代）。

例如，假定你已经按下Enter键：Object filename [MYFILE .OBJ] :

那么下一个出现的提示是：Source listing [NUL. LST] :

Source listing是你希望给包含汇编后源程序列表的文件所起的名字。如果你不要这个列表，那么只要按下Enter键，这样就给出一个缺省的文件名NUL. LST，这个文件名告诉汇编程序不必产生一个源程序列表文件。

例如：假定你已经输入：**Source Listing [NUL. LST] : myfile**

那么汇编程序将加上扩展名并且产生一个列表文件MYFILE. LST。

现在出现最后的提示：

Cross reference [NUL. CRF] :

CRF文件包含了程序中的变量的引用及其定义的紧凑表示。索引程序处理CRF文件，使其生产一个按字母顺序排列的符号表（详见第三章中描述的“索引程序”）

例如：假定你已经作出如下回答：

Cross reference [NUL. CRF] : myfile. crf

这里的文件扩展名 .CRF就没有必要输入了 因为汇编程序将提供 .CRF做为缺省文件扩展名，而产生文件MYFILE .CRF。

如果你使用上述各例子中的文件名，汇编程序到此为止在屏幕上所显示的全部信息是：

Source filename [.ASM] : myfile

Object filename [MYFILE .OBJ] :

Source listing [NUL. LST] : myfile

Csoss reference [NUL. CRF] : myfile. crf

当你输入完最后一个文件名，汇编程序就开始对你的整个程序进行第一遍扫描。如果程

序中包含任一句法错误，则汇编程序就在屏幕上显示出与列表文件中同样的错误信息。

注意：做出任一回答之后，在按下Enter键之前，你可以用一个逗号来继续回答，这样就可以使你不用等待下一个提示出现而直接做出回答。如果你在任一地方用分号（；）结束，那么剩余的回答都认为是缺省的，汇编程序立即开始处理而不再出现提示。

怎样开始汇编程序—选择方式2

你从键盘输入：〔M〕ASM source, object, list, cross—ref／parms;

则宏汇编程序从盘上装入内存，并且立即执行在命令字段中指示的任务。在命令字段中包含了你所给出的文件名：源文件，目标文件，源列表和索引表，与上述顺序依次对应。

当你使用这种命令行时，如果四个文件全都指定或者在命令行的末尾使用分号，那么在选择方式1中的提示信息就不出现。

注意：你可以任行选择一个文件扩展名加到列表文件（listing file）上。

如果输入不完整（少于四项）并且命令行的末尾没有使用分号，那么汇编程序将对剩余的未指定项目做出提示。汇编程序对／parms不做出提示，但是／parms可以加到命令行的末端或者加到对提示做出回答时所给出的文件标识符号上。每一个提示都显示出它的缺省，可按下Enter键来接受之，也可明确地指定一个文件名或设备名来取代之。但是，如果输入不完整而命令行的末尾又使用了分号，那么汇编程序将未指定的项目用缺省值代入，并且不再出现提示。

例子：

1）〔M〕ASM module

源文件是module.ASM。随后出现了一个提示显示出缺省文件module.OBJ；当你对此做出回答之后，又出现一个提示，显示出缺省文件NUL.LST；再对此做出回答之后，又出现新的提示，显示出缺省文件NUL.CRF。

2）〔M〕ASM module;

若在末尾加上分号，则下面的提示信息就不再出现，汇编程序即对源程序module.ASM进行汇编，所产生的目标文件将以module.OBJ文件名进行存贮，汇编结果不产生列表文件和索引文件。

3）〔M〕ASM module, , ;

这个例子与上例相似，但它要产生一个列表文件module.LST。

4）〔M〕ASM module, , , ;

这个例子与上例相似，但它的汇编结果是多产生一个索引文件module.CRF。

5）〔M〕ASM module, , ,

用了与上例相同的例子，但是没有分号。汇编程序对module.ASM进行汇编，所产生的目标文件以module.OBJ贮存，列表文件是module.LST，但是要给出一个带有缺省值的module.CRF提示信息。

6）〔M〕ASM module, NUL, ;

不产生目标文件，列表文件是module.LST 也不产生索引文件并且不再做出提示。

怎样开始汇编程序—选择方式3

参阅《IBM个人计算机磁盘操作系统》手册中对批处理命令做出的详细描述，批处理命令可以很容易地用来自动开始汇编。

使用批处理命令来显示列表和索引表的例子如下：

注意：在下列例子中，假定驱动器B：有源程序和用作列表的空间，而驱动器A：有DOS，汇编程序和CREF。

用EDLIN程序建立ASSEM．BAT文件。

```
A:MODE  LPT1:132
B:
A:ASM  %1, NUL, , ;
TYPE  %1.LST
ERASE  %1.LST
A:CREF  %1;
TYPE  %1.REF
ERASE  %1.REF
ERASE  %1.CRF
```

要使用这个批处理选项，通过按下Ctrl—Prtsc键将打印机置成回响方式（假如你有一台打印机），以此来获得屏幕上所显示信息的硬拷贝。

再键入：ASSEM module

宏汇编程序结束

当宏汇编程序运行结束，控制就转到磁盘操作系统。

在宏汇编程序运行结束之前，你可以用同时按下Ctrl和Break键来终止宏汇编程序的运行而提前退出宏汇编程序。

对宏汇编程序（MASM）的每一个汇编错误都将导致一个错误信息和产生这个错误的语句显示在屏幕上。而对于小汇编程序（ASM），只仅仅显示一个错误代码。（如果列表字段（list field）没有指定CON：，则屏幕上就仅仅显示错误）。如果你有一台打印机，那么这些列表文件也可以通过按下Ctrl—Prtsc键打印出来。

参数（／params）

参数是一个字母，这个字母可以加到命令串上（选择方式2），也可以加到任一提示信息上（选择方式1）。参数的前面必须加上斜线（／）。这个字母指定一个选择的任务，而这个任务必须在汇编期间被执行。你可以任意次序地使用多个参数，但是每一个参数必须加有斜线（／）。可以使用的参数及其意义如下：

·／D—这个参数使得汇编程序在两次扫描过程中（第一遍扫描和第二遍扫描）产生一个列表。这个列表为你分析第一遍扫描和第二遍扫描之间的状态错误提供信息。

·／O—产生的机器语言及偏移量均以八进制数形式表示。

·／X—取消未成立条件的列表。这个参数通常与伪操作：.SFCOND，.LFCOND和

.TFCOND一起使用。参看第5章中对这些伪操作的详细描述。

例子：ASM MODULE, , /O;

在其列表中，机器语言是以八进数形式表示的。

设 备 标 识 符

如果要取代隐含的DOS的缺省软盘驱动器，你可以为你的输出设备指定设备标志。设备名和指定的标识符（IDs）是：

- 软盘驱动—A：，B：
- 显示屏幕—CON（缓冲输出），或USER（无缓冲输出）。
- RS－232（串行口）—COM1（缓冲输出），或LINE（无缓冲输出）

注意：屏幕显示可以通过按下Ctrl和Num LOCK键来暂停，再按下任一字符就恢复显示。当屏幕要显示大量的信息时，使用这种方法停止显示滚动以便阅读。

第三章索引程序CREF

目录：

产生一个索引文件
怎样开始CREF
怎样开始CREF—选择方式1
怎样开始CREF—选择方式2
怎样开始CREF—选择方式3
CREF结束
CRF格式

索引程序（Cross Reference Facility）（CREF）通常用来帮助你调试程序。这个程序打印一个列表，该列表具有模块内所有符号名的引用索引，为了使用CREF，你首先必须用ASM或MASM命令来规定产生module.CRF文件。

CRF文件包含了你的程序中的变量引用及其定义的紧凑表示。

CREF程序处理CRF文件，产生一个在文件内按字母顺序排列的符号表，并且列出每个符号被引用的行号或被定义的行号（用＃号表示）。

产 生 一 个 索 引 文 件

要产生一个索引文件，你必须在宏汇编的命令引中对第四个提示信息“Cross reference”作出回答，例如，假定你用“myfile”来回答第四个提示则屏幕显示如下：

```
Cross reference [NUL.CRF] : myfile
```

对这个提示作出回答之后，MYFILE.CRF文件就被生成。

怎 样 开 始 CREF

你可以用下列三个选项之一来使用索引程序。对这三个选项的选择取决于系统本身的配置：

- 如果你的系统是单个软盘驱动器，则选择方式1允许你在继续使用CREF命令之前更换盘片。
- 如果你的系统有两个软盘驱动器，则选择方式2允许你立即使用CREF
- 选择方式3允许你使用由磁盘操作系统提供的批处理命令。批处理命令为CREF命令提供一个组合能力这样可以使索引程序自动开始。

怎样开始CREF选择方式1

从键盘输入：CREF

则索引程序就从盘上装入内存，在一个短时间后，这个实用程序将显示下列提示：

Cross reference〔.CRF〕：

在对此做出回答之前，可以将包含CREF的盘片拿掉，换上包含数据文件的盘片。

注意：

1．显示在方括号内的名字（.CRF）是缺省的文件扩展名，当你对自己的文件不给出文件扩展名时，CREF实用程序就使用这个缺省扩展名。这个缺省文件扩展名可以由你明确指定一个扩展名来取代。

2．虽然实用程序在你不指定一个扩展名时总是自动提供一个缺省的文件扩展名，但是这个扩展名（缺省的）可以由明确指定一个新的文件名（带有新的扩展名）来取代。

3．缺省的盘片驱动器是DOS缺省驱动器，它可以用在文件标识符前加上驱动器标识符ID来取代：

Cross reference 是一个文件名，它指示汇编程序将有关信息存放在这个文件中。利用在回答汇编程序最后一个提示时提供的文件名，将这些信息送到索引程序中处理。

输入索引文件名后，你将看见下列提示：Listing〔Cross reference.CRF〕：

Listing是你想要的可打印的索引列表文件的名字。如果你希望这个列表文件与源文件使用相同的名字贮存（但扩展名是.REF），则只要按下〔ENTER〕键即可。你也可以指定另一文件名，实用程序将自动给这个文件名加上缺省扩展名 .REF（当你没给出扩展名时），你还可以用明确指定的扩展名来取代之。CON或COM 1标识符分别表示将列表文件直接送往显示器或RS－232端口。

怎样开始CREF—选择方式2

从键盘输入：CREF Cross reference, list;

如果你对此式要求输入不完整，并且没有使用分号（；），那么CREF程序将对未指定的文件做出提示。这个提示显示出它的缺省文件名，你可以直接按下〔Enter〕键表示接受该缺省文件名；也可以用明确指定一个文件名（允许包含驱动器标识符ID）或设备名（CCN或COM1）来取代缺省文件名。如果你给出的仍然是不完整的输入但是在末尾使用了分号，则汇编程序将对未规定的文件指定缺省的文件名而不再出现提示。

例子：

1) CREF module

输入的索引文件是module.CRF。这时实用程序出现一个提示，显示出缺省的列表文件将是module.REF。’ 当你输入回答之后，实用程序就开始进行操作。

2) CREF module;

如果加进了分号，实用程序就不进一步给出提示。输入的索引文件是 module.CRF。该列表文件是module.REF。

3) CREF module, CON

因为给出了完整的清单，所以有无分号都无所谓了。输入的索引文件是module.CRF；列表文件被直接送到显示器。你可以按下Ctrl—Prtsc键来获得屏幕内容的硬拷贝。

4) CREF module, B :

这个例子与上例相同，但是它的列表文件是缺省文件名module.REF，并且要被送到B：软盘驱动器上，而不是DOS缺省的驱动器上。

怎样开始CREF—选择方式3

参见《IBM个人计算机磁盘操作系统》中对批处理命令用于自动开始索引程序的详细描述。做为一个例子，请参见前面一章中的选择3例子。

CREF结束

当CREF程序运行完毕，控制就回到磁盘操作系统（DOS）。

你可以同时按下Ctrl和BREAK键，使程序中途退出CREF。

CRF格式

除了对汇编程序产生的文件以外，你还可以对其它文件使用CREF程序。module.CRF文件格式是：

第一字节：进纸到下一页前所打印的行数。

第二字节：每行打印的字符数。

第三字节：用来表示记录类型07中页长／宽的标识符。

注意：如果在文件开始没有使用这三个字节，则CREF就认为页的长度为58而页的宽度为80。

记录（records）：文件的其余部分是一系列记录。每个记录的开始字节或结束字节定义如下记录类型。

以一个记录类型字节开始的记录：

记录类型字节	名字	随后字节
01	引用符号	被引用的符号名（1－80个字符）

0 2	定 义 符 号	被定义的符号名（1－80个字符）
0 4	行 结 束	无
0 5	文 件 结 束	IAH

以一个记录类型字节结束的记录：

记录类型字节	名 字	前面字节
0 6	定 义 标 题	标题正文（1－80个字符）
0 7	页 长 ／ 宽	一个字节的长度，随后是一个字节的宽度。

源程序中的每一行module.CRF文件中至少必须产生一个登记项。例如，一个没有引用的COMMENT行，就产生单个的十六进制数04记录（行结束）。这就是CREF实用程序如何与引号保持一致的原因。

第四章 汇编语言格式

目录：

符号/数据的表示
数据项
常数（立即值）
第一遍扫描和第二遍扫描
标号
变量
寄存器代码约定
标志寄存器
操作数
属性操作

源程序中汇编语言语句的组成：

- 名字项（通常是选择的）
- 操作项（要求）
- 操作数项（某些指令要求）
- 注介项（选择）

项的格式和定义：

〔名字〕操作 操作数 〔；注介〕

- 名字项是一个符号。
- 操作项是一个记忆符操作码，它表示一个机器指令或汇编指令，或一个伪操作。
- 操作数项是由一个或多个表达式组成的操作数，它提供专门的操作码要求的信息来完成要求的功能。
- 注介项可以用来描述你的程序。；号为注介场的开始，当第一个字符是；号时，整行是一个注介项。注介项也是伪操作（看第五章的注介）。注意：这些项必须用至少一个空格或

制表跳格符（TAB）分开，并且必须按所述次序书写。语句不必在行的第一个字符位置开始，但不允许连续超过132个字符。

符号/数据的表示

源程序语句用下面的字符来表示名字。

· 字母A～Z（输入时，小写字母变换成大写字母）。

· 0～9数字

· 专用字符：？ ·@ — $

除了数字，所有的字符可以编在源语句的第一个位置。当在名字中用·符时，它必须是第一个字符。·可以用作结构内的结构名字和项名字之间的分隔符（看第五章STRUC伪操作），虽然可以用许多字符来描述名字，但只有前面的31个字符能被汇编程序所识别。

数据项：

数据项有三种：常量，标号或变量。每一种都可以用在源程序语句中允许表达式使用的地方。它们也能作为操作数使用。

常量（立即值）

常量是除了自身的值以外没有其它属性的数值。常量按以下形式给出：

二进制

0和1的序列，后跟着字母B（例00101100B）。

十进制

数字0～9的序列，后跟着的字母D可有可无。（例178；178D）除非缺省值被RADIX伪操作修改（看第五章“RADIX”）。

十六进制

0～9数字和A～F的序列，后跟着字母H(例OFFFH)。第一个字符必须是数字0～9

八进制

数字0～7的序列，后跟着字母Q或字母O（例1777Q；1777O）。

字符

对括在单引号和双引号内的字符串，使用ASC11码值。多于二个字符仅仅对定义字节DEFINE BYTE有效（如“C”和‘C’）

十进制科学记数法

浮点十进制数字（例27.25E—2）

注意：小汇编程序（ASM）不支持。

十六进制实数

十六进制实数是数字（0～9和A～F）跟着字母R。

第一个字符必须是0～9数字之一，除非第一个数字是0。数字的总值数应当是8、16或20，那么实数的总位数则为9.17或21（如OFFFFFFFFR）。

注意：小汇编程序（ASM）不支持。

第一遍扫描和第二遍扫描

你的IBM个人计算机宏汇编程序是一个二遍扫描的汇编程序。它二次读你的源程序。

第一遍扫描是用来为源程序的每一行确定相对偏移量的。第二遍扫描则产生列表、目标和CREF文件。尽管这是两遍不同的扫描，但汇编每次都执行同样的代码，其间只有稍许不同。

第一遍扫描和第二遍扫描的区别

第一遍扫描：

· 仅确定的错误类型被显示（错误涉及必须在第一遍扫描时定义的项，例如不好的.RADIX值）。

· 不产生目标码

· 如果发现未定义的项，则对该符号进行假设，以便在第一遍扫描中产生正确的字节数。

· 不产生列表（一个/D参数将在两遍扫描都产生列表）。

第二遍扫描：

汇编程序使用在第一遍扫描中定义的值以产生输出。当在第二遍扫描时，发现有符号定义，第二遍扫描的值再次与第一遍扫描的值（它在符号表中）比较，如果它们不等，则产生一个相位错误。因为第一遍扫描必须保持相对偏移量的正确跟踪，有一些引用信息必须在第一遍扫描时就知道，如果不是这样，相对偏移量不会是正确的。

下面的引用信息必须在第一遍扫描时知道：

1F/1FE表达式：

如果表达式在第一遍扫描不知道，那么汇编不知道条件汇编哪一块，因此，将在第二遍扫描时产生相位错误。

表达式DUP（…）：

这明确地改变相对偏移量，所以表达式应知道。

（…）不必知道，因为它不影响所产生的字节数。

· RADIX 表达式：

因为这改变基数。常量应是不同的，它使汇编程序对1F或DUP赋值不适当。

1F1和1F2也是伪操作，它们让你知道是在第一遍扫描或在第二遍扫描里进行汇编。它们的主要用途如下：

· 如有条件，汇编要求决定程序的版本时，%OUT可用在1F1/1F2下来显示被汇编的程序的版本。

· 某些程序有大量的宏指令，但它只被定义一次（可在一个独立的文件中并用INCLUDE伪操作将之汇编进当前的源文件）。当它们不改变时，为了节省宏指令，在第二遍扫描时再定义的时间。1F1可以放入后边跟有一个ENDI的定义之前。

相位错误

相位错误表明汇编程序在第二遍扫描时发现一个LABEL，VARIABLE或PROC与第一遍扫描不同的值。原因是：

· 1F1/1F2的使用不适当（在每一遍扫描时有不同的字节数）。

·第一遍扫描时就提前引用STRUC（因为在第一遍扫描时，还未定义故不产生字节，第二遍扫描才产生STRUC）
·在一条指令中的提前引用违背了汇编程序所做的假设。
·第一遍扫描测到一个错误。

错误可以在具有错误标号的前一个标号和该标号之间。检查代码，错误可能容易找到。否则，你可用/D参数和对两个列表中的相对偏移量进行比较，这应能找到错误的地方。除了下面两点，第一遍扫描的列表看起来就象第二遍的扫描的列表：
·由于提前引用，可能产生错误。
·第一遍扫描可能产生不同的操作数编码。
注意：第一遍扫描产生的错误通常被显示出来，且不计入最后的错误总计。

提前引用

编码的开始部分，存放变量，在这些变量被以后的代码使用之前，汇编程序已经能够识别这些操作数的属性（类型，程序段，偏移量）。这就允许汇编程序通过产生较短的指令产生较高效率的代码。引用数据总是用两个字节的偏移量，但控制的传送可以在1，2或4个字节之前变化，这取决于使用的上下文及规定。寄存器不可以提前引用。如果提前引用过后被定义为寄存器，将产生错误。

标号：

标号有三种属性与之使用相联系：程序段，偏移量和类型。

程序段属性是定义标号的程序段起始段号，为了对标号寻址，这个值必须在一个段寄存器中。通过将段寄存器（CS、DS、ES和SS）的信息同标号的引用信息相结合，汇编程序确保标号的可寻址性。标号总是被假设在CS寄存器中（看本章“返回值操作”的“SEG”）。

偏移量属性是16位无符号数。它表示从标号被定义的段的起始之处到该标号被使用之处，这之间的字节数（看本章“返回值操作”的“OFFSET”）。

类型属性指明，标号是否在本段内被引用（称作NEAR）或者可在其它段被引用（称作FAR），NEAR指出指针长度为2字节。FAR指出指针长度为4字节。标号被定义时附加：号，则汇编程序确信它是NEAR属性。标号也能用LABEL伪操作来定义。（看第五章）。

标号可在跳转和调用中作为操作数使用。

它们只能定义在可执行的代码上。

变量

变量也有与之使用有关的三个属性：

程序段，偏移量和类型。

程序段属性是变量被定义的程序段的起始地址段号。为了变量被寻址，这个值必须在段寄存器中。通过把段寄存器（CS，DS，ES和SS）的信息同变量引用信息（通过使用ASSUME伪操作或明确指定段寄存器名称）的结合，汇编程序确保变量的可寻址性。程序段的值可被装入CS，DS，ES或者SS寄存器。偏移量属性是16位值。它表示从变量被定义的段起始处

到变量被使用处这之间的字节数。在当前程序段内给出变量的偏移量等于当前地址计数器的值。当前地址计数器可以用$号来设置。变量的类型属性定义成为变量保留的字节数。DB，DW，DD，DQ，DT，RECORD和STRUC伪操作和变量类型相联系(看第五章)。变量可以以简单的，变址的或者结构形式作为操作数。见本章“变量操作数”。

寄存器代码约定

下面的寄存器被汇编程序隐含了介并作为保留符号。当一个寄存器被一条指令引用时，它必须引用成下列这些符号中的一个：

- 通用8位寄存器——AH，AL，BH，BL，CH，CL，DH，DL。
- 通用16位寄存器——AX，BX，CX，DX，SP，BP，SI，DI。
- 基址和变址寄存器——BX，BP，SI，DI。
- 段16位寄存器——CS，DS，SS，ES。

段寄存器被磁盘操作系统初始化从而包含段的基址值。因为每个程序段开始于某一段号，四个16位的程序段寄存器用作保存程序段开始处的段号。任一时刻都有4个当前程序段，每个段号都作为程序段基址值而加以引用。并且各自被装入如下的程序段寄存器中：

- ES寄存器——能定义一个辅助数据程序段。
- CS寄存器——总是定义一个当前代码程序段。
- SS寄存器——总是定义当前堆栈程序段。
- DS寄存器——通常定义当前数据程序段。

标志寄存器

六个一位标志寄存器被大多数算术运算置成1或0反映运算结果。三个一位的标志是处理器控制标志，每个标志在16位标志寄存器内由一位表示，意义是：

- CF，进位标志。如果运算结果有一个从高位来的进位（加法时）或向高位有一借位（减法时），标志置1，否则，这一位被置0。
- PF，奇偶标志。当偶校验时置1，奇校验时置0。（处理器控制标志）
- ZF，辅助进位标志。当加法运算时从结果的低四位产生一个进位或减法运算时向结果的低四位产生一个借位，这个标志置成“1”；否则，置成“0”。
- EF，运算结果为0，置这个零标志为1；否则，置成0。
- SF，符号标志。当结果的高阶位置1时，它置成1，否则，这个标志位被置成0。
- TF，自陷标志。置系统于单步方式，以便调试排错。内部中断发生于每条指令之后，以便你能一条条指令地检查你的程序。（处理器控制标志）。
- IF，中断允许标志。允许确认外部（可屏蔽的）中断请求。如IF位清0，则关中断（处理器控制标志。
- DF，当方向标志置1（被STD指令）时，SI和DI被减；如果置0（被CLD指令）时，SI和DI被增。这个位用于MOVS，MOVSB，MOVSW，CMPS，CMPSB和CMPSW指令。
- OF，当带符号的运算产生溢出，溢出标志置1；否则，标志置成0。

操作数

汇编指令让你以不同的方法对操作数进行寻址。大多数双操作数指令允许存贮器或者寄存器作为第一操作数（目的），而存贮器、寄存器或在指令中的常数作第二操作数（源）。

立即操作数

立即操作数是指本身内的表示的值。如这条指令中的0：　　MOV　　AX，0

存贮器操作数

存贮器操作是指令内一个地址的某种形式。存贮器操作数或者是一个标号或者是一个变量。存贮器操作数可按如下来寻址：

·直接16位偏移量寻址。

　　MOV　　AX，TABLE

·间接通过一个基址寄存器（BX或BP）可选择带一个8位或16位的位移量。

　　MOV　　AX，〔BX〕＋8

·间接通过一个变址寄存器〔SI或DI〕可选择带一个8位或16位位的移量。

　　MOV　　AX，4〔SI〕

·间接通过一个基址寄存器〔BX或BP〕和一个变址寄存器〔SI或DI〕的总和　可选择带一个8位或16位位移量。

　　MOV　　AX，16〔BX＋SI〕

JMP和CALL操作数

JMP和CALL的操作数可以是一个标号，变量，寄存器或一个地址表达式。对于直接JMP或CALL，操作数是与目标（JMP或CALL的目的）等同的标号。对于间接的JMP或CALL，操作数是一个字长（word）或者双字长（DOUBLEWORD）的指针分别用来指出定义为NEAR或FAR的目标（一个地址表达式）。当一个JMP是在程序段内，且距离不大于＋127至－128字节，假如目标标号用：号定义或者已定义与NEAR，你就能使用SHORD操作。例子看第六章中描述的JMP和CALL指令。

变量操作数

变量操作数可以是简单的，变址的或结构的，具体如下：

简单的：简单的变量是一个被DEFINE伪操作（看第五章）所定义的不改变的标识符。

变址的：变址的变量是一个简单变量后缀一个方括号，方括号中有如下之一：

1．一个常数或立即值。例〔8〕

2．一个基址寄存器（BX或BP）或变址寄存器〔SI或DI〕例〔BX〕或者〔SI〕

3．一个基址寄存器或变址寄存器加或减一个立即值，其次序任意。例〔BP＋8〕

4．基址寄存器加变址寄存器，再加或减一个常量或一个立即值（按任何次序）。例〔BX＋SI－1〕

5．变址数组的下标表达式。例ADDF00〔100＊TYPF　F00〕，9

6．双变址变量。它由一个简单变量后跟括在方括号中的一个基址项和一个变址项组成。

　·给定的基址和变址，后随变量的次序可以是任意的。