386|486 DOS EXTENDER 软件开发系列丛书

386|ASM汇编语言程序设计

罗文海　杨再光　编
郭　峰　郭平平
朱丽华　审校

2

内 容 简 介

386|ASM是用于Intel 80386微处理器的先进的汇编语言。386|ASM能在8086、8088、80186、80188、80286、80386和80486等微处理器上汇编汇编语言源程序。本书对386|ASM汇编语言的语法格式和使用方法作了详细讨论。每一章都附有完整的程序设计实例，帮助读者掌握难点和程序设计技巧。本书以讲授保护模式程序设计方法为特色，是从事80386汇编语言程序设计的必备工具书。

编校者的话

尽所周知，386｜486微机已成为（或正逐步成为）我国各行各业计算机应用中的主流机型，尤其是在工程设计、研究领域更是如此。然而，遗憾的是，尽管这些机型（386｜486）得到了广泛使用，但是真正能发挥这些机型硬件特性的应用软件和软件开发工具却少得可怜。原因是由于受到DOS本身的限制，目前大多数软件都只能在实模式环境运行，而不能在保护模式环境运行。80386较之80286而增加的两个主要功能，即32位运算及数据类型和分页技术（除了8086家族中都有的分段技术之外）均没有得到有效利用。我们深信，386｜486 DOS EXTENDER软件开发系列丛书的出版，能为改变这一状况作出贡献。

本套系列丛书实际上主要由两部分组成：一部分是PHAR LAP软件公司的386｜DOS EXTENDER SDK（即本系列丛书的书目之一至之六）；另一部分是Metaware公司的High C编译器（即本系列丛书的书目之七至之十）。由于用MetaWare公司的High C编译器开发高级专业应用软件需要有PHAR LAP软件公司的386｜DOS EXTENDER SDK环境的支持，所以我们将这两部分合并成一体出版，希望能方便用户的使用。

本套系列丛书适用于DOS环境下开发各种专业性软件包的程序员使用。那些希望施展自己才华、编制出高效实用程序的程序员将会发现，本丛书中所介绍的许多先进特性及工具是他们以前不曾见过的。本丛书不仅提供了386｜DOS EXTENDER SDK这样的开发环境的完整资料，也提供了整体优化编译程序High C和其它一些高级工具（有些工具，如386｜DEBUG是目前分析和调试保护模式程序的必备工具）的完整资料。这套丛书及软件的诸多应用领域之一，就是用于Autodesk公司的AutoCAD绘图软件包的开发。AutoCAD绘图软件包的用户和软件开发者，至少有如下理由要用到本丛书：

·改善以前用AutoLISP语言实现的用户附加程序模块的性能（如增强程序的保密性、提高程序的运行效率等）。

·获得一种既能应用于AutoCAD环境的应用软件开发、又能同时访问系统的软硬件资源（如：访问DOS系统调用，访问绘图仪、鼠标器、打印机和图形卡等等）的手段。

·实现技术上难度更大，以前用AutoLISP语言很难实现或者根本就不可能实现的功能。

本套丛书包括如下书目：

《386™DX微处理器程序员参考手册》（之一）

《386｜ASM汇编语言程序设计》（之二）

《386｜DOS EXTENDER程序员参考手册》（之三）

《386｜VMM程序员参考手册》（之四）

《386｜LINK程序员参考手册》（之五）

《EC编辑器和386 | DEBUG手册》（之六）

《High C程序设计指南》（之七）

《High C库参考手册》（之八）

《High C程序设计工具集》（之九）

《High C程序调试器手册》（之十）

本丛书编校过程中，得到北京希望电脑公司秦人华老师以及海洋出版社的编辑同志的大力支持和帮助，特此致谢。

由于本丛书编写时间仓促，书中某些内容缺乏时间验证，不妥之处在所难免，恳请读者不吝赐教。

编 校 者

一九九一年十月于北京马家堡

目　录

第一章　386 | ASM导言

本书是为有经验的高级系统软件开发者而编写的。使用本书的读者，我们假设已经熟悉了有关的其它Phar Lap的产品，并且对Intel芯片的体系结构有深入的了解。

386 | ASM是用于Intel 80386微处理器的汇编语言。它是Phar Lap软件公司的80386软件开发工具包系列软件中的一个产品。386 | ASM用于把一个或多个汇编语言源程序汇编成一个目标模块。386 | ASM能在8086、8088、80186、80188、80286、80386和80486等微处理器上汇编汇编语言源程序。各种386 | ASM版本，均能运行于IBM PC、PC/AT、VAX/VMS以及在这些系统上提供的不同版本的UNIX操作系统的环境下。

当386 | ASM用于汇编生成8086、8088、80186、80188或80286代码时，它所生成的代码符合Intel 8086目标模块格式（OMF-86）。当386 | ASM用于汇编生80386代码时，所生成的代码对OMF-86格式进行了简单的扩充，我们简称为OMF-386。附录K对OMF-386目标文件格式进行了讨论。

在本书中使用了一定的约定，以便于表达一定的信息类型。下面介绍这些约定。

Courier	这种类型的铅字印出的字样，指出命令行、开关语法和实例。这样可使得它与实际的命令行有区别。
{ }	括在花括号内的项是可选择的项。省略后并不会影响语句的有效性。
,…	该记号表示逗号前的项可以重复任意次。两个项之间要用逗号分隔。
|	竖线是个分隔符。只能在用竖线分隔的各项之中选择一个项。也就是说，用竖线分隔的各项之间，具有逻辑上的"或"关系。
斜体字	以斜体字表示的项可以用对某一特定语句而言是适当的一个符号来代替。斜体字通常是一描述性的名字，用它来识别一类可用的符号。

例如，建立一个10字节的临时实型常数有如下格式：

```
DT    digit.{digits}{E{+ | -}digits},…
```

该语句说明一个实型数。一个实型数可以由如下几部分组成：一个必需的整数部分、一个必需的十进制小数点、后面接一个可选的小数部分、再后面接一个可选的指数部分（如有指数部分，指数部分必需用大写字母开头，后面可接+或-符号（可选），再接一个必需的幂指数值）。一条DT语句，可用于说明任意多个实型常数，各常数之间用逗号分隔。因此，下列语句都是有效的：

```
DT      1.0
DT      1.0, 1.1, 1.2
DT      1.0E2
DT      1.0E+2
DT      1.0E-2
DT      1.E2
```

```
DT      0.
DT      0.E0
DT      314.159E-2, 0.31459E1
```

在本书的所有例子中，都用大写字母表示汇编伪指令和表达式操作符。小写字母用于所有其它的汇编语言保留字和所有用户自定义符号。给出这个约定仅仅是为了使得程序具有更好的可读性。实际上，大小写字母能用于所有汇编语言保留字和用户自定义符号。

386|ASM可以处理能生成与Microsoft MASM 汇编语言源程序相兼容的那些软件产品所生成的汇编语言源程序，它能形成Intel/Microsoft 标准目标模块格式（OMF-86）的目标文件。386|ASM已用如下产品进行过测试，而没有发现任何问题，它们是：

1. 在8086模式下

Phar Lap 386|LINK	1.0版或更高版
Microsoft LINK	3.0版或更高版
Microsoft LIB	3.0版或更高版

2. 在80286模式下

Phar Lap 386|LINK	1.0版或更高版

3. 在80386模式下

Phar Lap 386|LINK	1.0版或更高版

第二章　386 | ASM的使用方法

§2.1　命令行语法

用于运行汇编程序的命令行，是由汇编程序的任务映像名（在IBM PC系列机上，汇编程序的任务映像名是386asm）和跟随其后的一个由文件名、开关构成的一个列表所组成的。开关用于略去汇编程序的缺省操作。对于缺省情况，386|ASM汇编一个或多个输入文件（汇编语言源程序），并且只产生一个输出目标文件和一个输出列表文件。当在汇编过程中出现错误的话，就会有一条出错信息显示在屏幕上，此外，还会将出错信息写入到列表文件中。使用一个开关，还可以将要写到屏幕上的出错信息改向到错误列表文件中。

表2—1　关于文件名的扩展名的约定

文件类型	在IBM PC/MS-DOS下所取的扩展名	在VAX/VMS和UNIX下所取的扩展名
输入源文件	.ASM	.A86
输出目标文件	.OBJ	.086
输出列表文件	.LST	.LIS
输出错误列表文件	.ERR	.ERR

在命令行中，文件名可以写完整（即完整给出文件名和文件名的扩展名），也可以仅给出文件名而省略文件名的扩展名。在不给出文件名的扩展名的情况下，386|ASM会提供一个缺省的文件名的扩展名。在不给出文件名的扩展名时，汇编程序会按表2—1所作的约定设定文件名的扩展名。

除此之外，也可以随文件名一起指定完整的或是部分的路径名。如果在文件名之前没有指定设备名和路径名，则汇编程序假定文件存贮在现行设备和现行路径名内。

开关符以负号“—”开头，后跟一个开关名。负号和开关名之间不允许有空格。开关的参数紧跟在该开关名之后，中间用一个空格作分隔符。相邻的输入文件名可以用空格或者逗号分隔。相邻的开关，或者，一个开关与输入文件名相邻，其间必需用空格分隔。输入文件名和开关名在命令行中可以以任意次序排列。输入文件名不能以负号“—”开头，这样就使得386|ASM能区分出文件名和开关名。下面是一些正确使用命令行的例子，更多的例子请见附录L。

例1：汇编输入源文件名为TEST1.ASM和TEST2.ASM，产生的目标文件名为 TEST1.OBJ，列表文件名为TEST1.LST，错误列表文件名为TEST1.ERR。

```
386asm test1, test2 -errorlist test1
386asm test1, test2 -el test1
386asm -el test1 test1 test2
```

例2：汇编输入现行目录的源文件TEST1.ASM和现行目录的子目录SUBD 中的源文件 test2.A，所生成的列表文件和目标文件存放在现行缺省磁盘驱动器的现行目录的上一层目录LISTD和OBJD中。分为如下三种情况：

1. 在IBM PC/MS-DOS环境

```
386asm test1, subd\test2.a -list\listd\test1 -0 \objd\test1
```

2. 在VAX/VMS环境

```
xa386 test1, [.subd]test2.a -l [listd]test1 -0 [objd]test1
```

3. 在UNIX环境

```
xa386 test, sub/test2.a -l /listd/test -0 /objd/test1
```

§2.2 命令行开关

命令行开关用于改变386|ASM的缺省操作。在缺省情况下，386|ASM将：

- 产生一目标文件，文件名与第一个输入文件名相同，文件的扩展名按表2—1中的约定确定。
- 产生一列表文件，文件名与第一个输入文件名相同，文件的扩展名按表2—1中的约定确定。
- 假定目标CPU是80386。
- 汇编后产传的目标代码将不会有8087或80287浮点处理器指令。
- 在汇编过程中，对大小写字符是不敏感的（意即，一个符号的大小写其意义是相同的）。

命令行开关用负号“—”开头，其后紧跟一个开关名。每一个开关名有两种格式，即长格式和短格式。开关参数必须紧跟开关名，其间用一个空格作为分隔符。如果在命令行中给

出的开关之间发生冲突，则最右的那个开关起作用。

§2.2.1 目标文件开关

1. -OBJECT开关

-OBJECT 开关用于指定一个目标文件名，这可使得目标文件的文件名不取由汇编程序缺省指定的文件名。另外，该开关还可使得目标文件有效到不同于现行缺省目录的文件目录中。

-OBJECT开关的语法是：

-OBJECT filename

-OBJECT开关的短格式是：

-O filename

例1：在IBM PC/MS-DOS环境的实例

```
386asm test -o test.o
386asm test -object t.obj
386asm test -o \objdir\test
386asm test -o \objdir\test.o
```

例2：在VAX/VMS环境的实例

```
xa386 test -o [objdir]test
```

例3：在UNIX环境的实例

```
xa386 test -o /objdir/test
```

2. -NOOBJECT开关

-NOOBJECT开关用于指示386|ASM不产生目标文件。

-NOOBJECT开关的语法是：

-NOOBJECT

-NOOBJECT开关的短格式是：

-NOO

例2：-NOOBJECT开关的实例

```
386asm test -noobject
386asm test -noo
```

3.-NODELETE开关

-NODELETE开关用于指示386|ASM不删除 .OBJ文件，既使在汇编过程中发现了严重的错误。对于缺省情况，在汇编过程中发现存在严重错误时，则会删除 .OBJ文件。

-NODELETE开关的语法是：

-NODELETE

-NODELETE开关的短格式是：

-NOD

例5：-NODELETE开关的实例

```
386asm test -nodelete
386asm test -nod
```

§2.2.2 列表文件开关

1. -LIST开关

-LIST 开关用于指定一个列表文件名，这可使得列表文件名不取由汇编程序缺省指定的文件名。另外，该开关还可使得列表文件存放到不同于现行缺省目录的文件目录中。

-LIST开关的语法是：

-LIST filename

-LIST开关的短格式是：

-L filename

例1：在IBM PC/MS-DOS环境的实例

```
386asm test -list test.l
386asm test -l \listdir\dest
```

例2：在VAX/VMS环境的实例

```
xa386 test -l DRA1:[listdir]test
```

例3：在UNIX环境的实例

```
xa386 test -l /listdir/test.l
```

2. -NOLIST开关

-NOLIST开关指示386/ASM不产生列表文件。

-NOLIST开关的语法是：

-NOLIST

-NOLIST开关的短格式是：

-NOL

例4：-NOLIST开关的实例

```
386asm test -nolist
386asm test -nol
```

3. -NOSYM开关

-NOSYM开关指示386|ASM，在源文件列表的后面不产生符号表的提要。

-NOSYM开关的语法是。

-NOSYM

-NOSYM开关的短格式是：

-NOS

例5：-NOSYM开关的实例

```
386asm test -nosym
386asm test -nos
```

§2.2.3 错误列表文件开关

-ERRORLIST开关用于指定一个存放出错信息的文件。对于缺省情况，错误信息总是显示在屏幕上。如果指定了一个存放出错信息的错误列表文件，那么错误信息总是存入到该

文件中，即使只有一个错误被发现也会生成该错误列表文件。

-ERRORLIST开关的语法是：

-ERRORLIST filename

-ERRORLIST开关的短格式是：

-EL filename

例1：-ERRORLIST开关在IBM PC/MS-DOS环境的实例

```
386asm test -errorlist test
386asm test -el test.e
```

例2：-ERRORLIST开关在VAX/VMS环境的实例

```
xa386 test -el t.err
```

例3：-ERRORLIST开关在UNIX环境的实例

```
xa386 test -el /listdir/test
```

§2.2.4 包含搜索目录开关

-INCLUDE开关用于指定一个或多个目录，将在这些目录中搜索由 INCLUDE 伪指令中所包含的文件（请参阅§3.5的内容）。对于缺省情况（即没有指定包含搜索目录开关），则只在现行缺省目录中搜索。如果在命令行中指定了一个或多个包含搜索目录，386|ASM 在搜索现行缺省目前之前，按命令中指定的顺序从左到右搜索这些目录。

可以指定多个搜索目录。其方法是既可以在命令行上写上多个一INCLUDE开关，也可以在命令行中只写上一个-INCLUDE开关，但在其后给出多个参数，各参数间用逗号分开，参数之间不能有空格。

汇编程序试图根据-INCLUDE开关给定的目录，来定位源文件中 INCLUDE 语句所附加的文件名所给定的文件。因此，包含目录应该用与宿主系统中指定文件路径那样的相同方法来指定，以现行目录为基准的路径也可使用。

-INCLUDE开关的语法是：

-INCLUDE dirname{,dirname…}

-INCLUDE开关的短格式是：

-I dirname{,dirname…}

例1：在IBM PC/MS-DOS环境的实例

```
386asm test -include \includes\
386asm test -i \includes\, isubdir\
386asm test -i —\includes\
```

例2：在VAX/VMS环境的实例

```
xa386 test -i (includes) -i (.isubdir)
xa386 test -i (—.includes)
```

例3：在UNIX环境的实例

```
xa386 test -i /includes/, isubdir/
```

有关INCLUDE语句的详细介绍，请参阅§3.5。

§2.2.5 指令集开关

指令集开关用于实现指定的微处理器（8086；8088；80186；80188；80286或者80386）