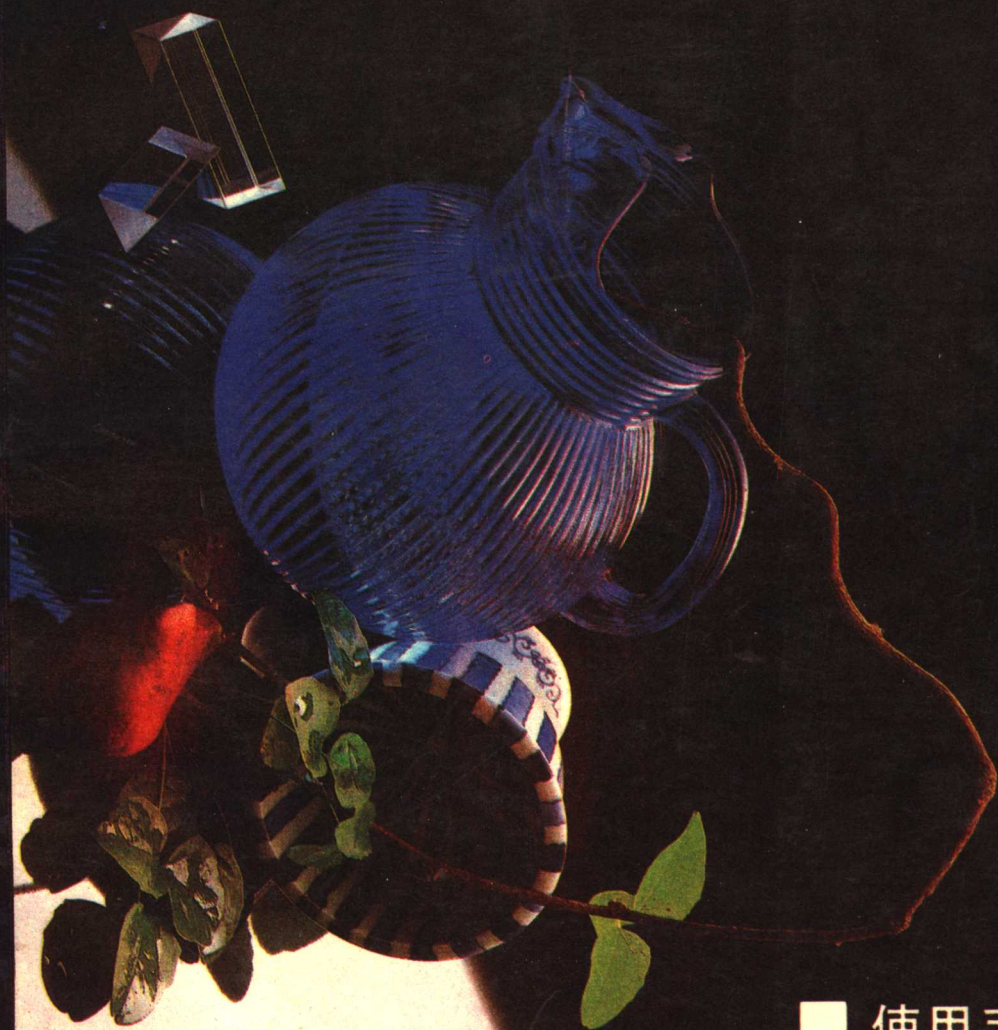


Turbo Assembler

汇编大全

(下册)



■ 使用手册

中国科学院希望高级电脑技术公司

一九九〇年九月



(共三册)

Turbo Assembler

汇 编 大 全

(下册)

■ 用户手册 (上册)

■ 程序设计技巧 (中册)

■ 使用手册 (下册)

中国科学院希望高级电脑技术公司

一九九〇年九月

前言

Turbo Assembler 是 Borland 公司推出的可与 MicroSoft 公司的 MASM 相匹敌的汇编语言软件。从某种角度来看, Turbo Assembler 的总体性能要比 MASM 好。Turbo Assembler 的速度是 MASM 所远不能比拟的。Turbo Assembler 与 MASM 完全兼容,带有可选的 Ideal 模式。

Turbo Assembler 支持 Turbo Debugger 源级调试器,方便、高效的调试工具会帮助你减轻调试汇编原会有的负担。汇编的繁琐你将不再会感觉到,代之以汇编的高速、对计算机的完全控制。

Turbo 系列高级语言的使用者若想优化代码、提高速度、使用低级功能,请使用 Turbo Assembler, 因为 Turbo Assembler 使 Turbo 系列高级语言与汇编的接口简化。

<<Turbo Assembler 大全>>分成三个部分: <<Turbo Assembler 用户手册>>, <<Turbo Assembler 程序设计技巧>>, <<Turbo Assembler 参考手册>>。

本书介绍 Turbo Assembler 使用的算符、预定义符及伪指令。

由于编译者水平有限,加上时间仓促,错误难免,欢迎指正。

文韬

1990.8.

简 介

本书是与 Turbo Assembler 软件包相配套的两本书中的第三本。如果已经读过本套书中的前二本(《用户手册》,《高级程序设计技巧》),那么一定会想阅读本书以了解更深一层的东西。

本参考手册为你提供了 Turbo Assembler 直接而详尽的说明。如果觉得仍需要了解汇编语言基础方面的内容,可参阅前二本,里面有更多、更深入的讲述。

§ 0.1 硬件和软件需求

Turbo Assembler 可在 IBM PC 类计算机上运行,包括 XT, AT, PS/2 及所有真正兼容机种。Turbo Assembler 需要 MS-DOS 2.0 以上的版本和至少 256K 内容。

Turbo Assembler 可产生用于 8086, 80186, 80286 和 80386 处理器的指令。它也可产生用于 8087, 80287 和 80387 协处理器的浮点指令。

§ 0.2 从本手册中能得到什么

本手册要讨论如下内容:

第一章:预定义符号,告诉你有关 Turbo Assembler 的等价符。

第二章:操作符,描述了 Turbo Assembler 提供的各种操作符。

第三章:伪指令集,按字母序提供了 Turbo Assembler 全部伪指令的详尽信息。

附录 A: Turbo Assembler 语法概要,以修改的巴科斯范式描述了 Turbo Assembler 表达式(包括 MASM 和 Ideal 两种方式)。

附录 B:兼容问题,包括了 MASM 和 Turbo Assembler MASM 方式差别的内容。

附录 C: Turbo Assembler 的扩充和改进,详细地讲述了 Turbo Assembler 对 MASM 的扩充和改进。

附录 D: Turbo Assembler 实用程序,描述了可与本软件包配套使用的六种实用程序:MASM、TLINK、TLIB、GREP、OBJXREF 和 TCREF。

附录 E: Turbo Assembler 出错信息,描述了使用 Turbo Assembler 可能产生的全部信息,包括信息性消息、致命错误信息、警告信息和出错信息。

§ 0.3 一些约定

当我们谈论 IBM PC 机或其兼容机时,我们是指使用 8088, 8086, 80186, 80286 和 80386 芯片(所有这些芯片可统称为 80x86)的任何类计算机。当我们谈论 PC-DOS 或 MS-DOS 时,我们是指其 2.0 以上版本的 PC 操作系统。

目 录

简介	1
§ 0.1 硬件和软件需求	1
§ 0.2 从本手册中能得到什么	1
§ 0.3 一些约定	1
第一章 预定义符号	1
@code	1
@CodeSize	2
@Cpu	2
@curseg	3
@data	3
@DataSize	3
??date	3
@fardata	3
@fardata?	4
@FileName	4
??filename	4
??time	4
??version	4
@WordSize	5
第二章 操作符	6
§ 2.1 算术精度	6
§ 2.2 操作符优先级	6
()	7
*	7
+ (双目)	8
+ (单目)	8
- (双目)	8
- (单目)	9
.	9
/	9
:	10
?	10
【】 操作符	11
AND	11
BYTE	11
DATAPTR	12

DUP	12
DWORD	12
EQ	12
FAR	13
FWORD	13
GE	13
GT	14
HIGH	14
LARGE	14
LE	15
LENGTH	15
LOW	16
LT	16
MASK	17
MOD	17
NE	17
NEAR	18
NOT	18
OFFSET	18
OR	19
PROC	19
PTR	19
PWORD	20
QWORD	20
SEG	20
SHL	21
SHORT	21
SHR	21
SIZE	21
SMALL	22
SYMTYPE	23
TBYTE	23
THIS	23
.TYPE	24
TYPE	24
UNKNOWN	25
WIDTH	26
WORD	26
XOR	26
§ 2.3 特殊的宏操作符	27

&.....	27
< >	27
!	28
%.....	28
;	28
第三章 伪指令集	30
样本伪指令	30
.186.....	30
.286.....	31
.286C.....	31
.286P.....	31
.287.....	31
.386.....	32
.386C.....	32
.386P.....	32
.387.....	32
.8086.....	33
.8087.....	33
:	33
=.....	34
ALIGN.....	34
.ALPHA.....	35
ARG.....	35
ASSUME.....	37
%BIN.....	37
CATSTR.....	38
.CODE.....	38
CODESEG.....	38
COMM.....	38
COMMENT.....	39
%CONDS.....	39
.CONST.....	40
CONST.....	40
.CREF.....	40
%CREF.....	41
%CREFALL.....	41
%CREFREF.....	41
%CREFUREF.....	41
%CTLS.....	42
.DATA.....	42

.DATA?	42
DATASEG	43
DB	43
DD	43
%DEPTH	44
DF	45
DISPLAY	45
DOSSEG	46
DP	46
DQ	46
DT	47
DW	47
ELSE	48
ELSEIF	48
EMUL	49
END	49
ENDIF	49
ENDM	50
ENDP	50
ENDS	50
EQU	51
.ERR	52
ERR	52
.ERR1	52
.ERR2	52
.ERRB	53
.ERRDEF	53
.ERRDIF	53
.ERRDIFI	54
.ERRE	54
.ERRIDN	54
.ERRIDNI	55
.ERRIF	55
.ERRIF1	55
.ERRIF2	55
.ERRIFB	56
.ERRIFDEF	56
.ERRIFDIF	56
.ERRIFDIFI	56
.ERRIFE	56

.ERRIFIDN	56
.ERRIFIDNI	56
.ERRIFNB	56
.ERRIFNDEF	57
.ERRNB	58
.ERRNDEF	57
.ERRNZ	57
.EVEN	57
.EVENDATA	58
.EXITM	58
.EXTRN	59
.FARDATA	60
.FARDATA?	60
FARDATA	61
GLOBAL	61
GROUP	62
IDEAL	63
IF	63
IF1	63
IF2	64
IFB	64
IFDEF	65
IFDIF,IFDIFI	65
IFE	65
IFIDN,IFIDNI	66
IFNB	66
IFNDEF	67
%INCL	67
INCLUDE	67
INCLUDELIB	68
INSTR	68
IRP	68
IRPC	69
JUMPS	69
LABEL	70
.IALL	70
.LFCOND	70
%LINUM	71
%LIST	71
.LIST	71

LOCAL	71
LOCALS	73
MACRO	74
%MACS	74
MASM	74
MASM51	75
.MODEL	75
MODEL	76
MULTERRS	78
NAME	78
%NEWPAGE	79
%NOCONDS	79
%NOCREF	79
%NOCTL	79
NOEMUL	80
NOINCL	80
NOJUMPS	80
%NOLIST	81
NOLOCALS	81
%NOMACS	81
NOMASM51	82
NOMULTERRS	82
%NOSYMS	82
%NOTRUNC	83
NOWARN	83
ORG	83
%OUT	84
P186	84
P286	84
P286N	84
P286P	85
P287	85
P386N	85
P386P	85
P387	85
P8086	85
P8087	85
PAGE	86
%PAGESIZE	86
%PCNT	86

PNO87	87
%POPLCTL	87
PROC	87
PUBLIC	89
PURGE	89
%PUSHLCTL	89
QUIRKS	90
.RADIX	90
RADIX	90
RECORD	90
REPT	91
.SALL	91
SEGMENT	92
.SEQ	93
.SFCOND	93
SIZESTR	94
.STACK	94
STACK	94
STRUC	94
SUBSTR	96
SUBTTL	96
%SUBTTL	96
%SYMS	97
%TABSIZ	97
%TEXT	97
.TFCOND	97
TITLE	98
%TITLE	98
%TREUNC	98
UDATASEG	98
UFARDATA	99
UNION	99
USES	100
WARM	100
.XALL	101
.XCREF	101
.XLIST	101
附录 A Turbo Assembler 语法概要	103
§ A.1 词法	103
§ A.2 MASM 方式下的表达式语法	105

§ A.3 Ideal 方式下的表达式语法	107
附录 B 兼容问题	110
§ B.1 环境变量	110
§ B.2 MicroSoft 二进制浮点数格式	110
§ B.3 Turbo Assembler Quirks 方式	110
§ B.3.1 字节移入/出段寄存器	111
§ B.3.2 对远程标号或过程出错的近程转移	111
§ B.3.3 使和=和 EQU 伪指令时类型信息的丢失	111
§ B.3.4 段调整检查	111
§ B.3.5 带符号立即数的算术和逻辑指令	112
§ B.3.6 MASM 5.1 特性	112
§ B.3.7 Masm 5.1/Quirks 方式的特性	113
附录 C Turbo Assembler 的扩充和改进	115
§ C.1 扩展的命令行语法	115
§ C.2 GLOBAL 伪指令	115
§ C.3 局部符号	115
§ C.4 条件转移扩展	115
§ C.5 Ideal 方式	115
§ C.6 UNION 伪指令/STRUC 嵌套	116
§ C.7 EMUL 和 NOEMUL 伪指令	116
§ C.8 明确的段重载	116
§ C.9 常量段	116
§ C.10 在 386 方式下扩展的 LOOP 指令	116
§ C.11 扩展的列表控制	117
§ C.12 可选的伪指令	117
§ C.13 预定义变量	117
§ C.14 Masm 5.0 和 Masm 5.1 的改进	117
§ C.15 改进的 SHL 和 SHR 处理	117
附录 D Turbo Assembler 实用程序	118
§ D.1 独立的 MAKE 实用程序	118
§ D.1.1 一个快速示例	118
§ D.1.1.1 创建一个 make 文件	119
§ D.1.1.2 使用一个 make 文件	120
§ D.1.2 创建 make 文件	121
§ D.1.2.1 Make 文件的组成	121
注释	121
显式规则	122
隐含规则	123
命令表	125
宏	126

伪指令·····	129
文件包含伪指令·····	129
条件伪指令·····	130
出错伪指令·····	132
!error 任何文本·····	132
消除定义伪指令·····	132
§ D.1.3 使用 MAKE·····	132
§ D.1.3.1 命令行语法·····	132
§ D.1.3.2 中止 MAKE 的说明·····	133
§ D.1.3.3 BUILTINS.MAK 文件·····	133
§ D.1.3.4 MAKE 是如何查找 make 文件的·····	133
§ D.1.3.5 TOUCH 实用程序·····	133
§ D.1.3.6 MAKE 命令行选择项·····	134
§ D.1.4 MAKE 出错信息·····	134
§ D.1.4.1 致命错误·····	134
§ D.1.4.2 一般错·····	135
§ D.2 Turbo Link·····	136
§ D.2.1 调用 TLINK·····	136
§ D.2.2 使用应答文件·····	137
§ D.2.3 TLINK 选择项·····	138
§ D.2.3.1 /x,/m,/s 选择项·····	138
§ D.2.3.2 /l 选择项·····	139
§ D.2.3.3 /i 选择项·····	139
§ D.2.3.4 /n 选择项·····	140
§ D.2.3.5 /c 选择项·····	140
§ D.2.3.6 /d 选择项·····	140
§ D.2.3.7 /e 选择项·····	140
§ D.2.3.8 /t 选择项·····	140
§ D.2.3.9 /v 选择项·····	140
§ D.2.3.10 /s 选择项·····	141
§ D.2.4 一些限制·····	141
§ D.2.5 出错消息·····	141
§ D.2.5.1 致命错·····	141
§ D.2.5.2 非致命错·····	142
§ D.2.5.3 警告·····	143
§ D.3 TLIB: Turbo 库管理员·····	143
§ D.3.1 使用目标模块库的优点·····	143
§ D.3.2 TLIB 命令行的组成·····	144
§ D.3.3 操作表(Operations)·····	144
文件名和模块名·····	145

TLIB 的各种操作·····	145
创建库·····	146
§ D.3.4 使用应答文件·····	146
§ D.3.5 改进的操作: /c 选择项·····	146
§ D.3.6 例子·····	146
§ D.3.7 创建一扩展词典: /E 选择项·····	147
§ D.4 GREP: 一种文件查找实用程序·····	147
§ D.4.1 GREP 选择项·····	147
§ D.4.1.1 优先级次序·····	148
§ D.4.2 查找串·····	149
§ D.4.2.1 正则表达式中的操作符·····	149
§ D.4.3 文件说明·····	149
§ D.4.4 带说明的例子·····	149
§ D.5 OBJXREF: 目标模块交叉引用实用程序·····	152
§ D.5.1 OBJXREF 命令行·····	152
§ D.5.1.1 命令行选择项·····	152
控制选择项·····	152
报告选择项·····	153
§ D.5.2 应答文件·····	153
§ D.5.2.1 自由形式的应答文件·····	153
§ D.5.2.2 连接器应答文件·····	153
§ D.5.2.3 /D 命令·····	154
§ D.5.2.4 /O 命令·····	154
§ D.5.2.5 /N 命令·····	154
§ D.5.3 OBJXREF 报告样本·····	154
§ D.5.3.1 按公用名报告·····	155
§ D.5.3.2 按模块报告 (/RM)·····	156
§ D.5.3.3 按引用报告 (/RR)(缺省方式)·····	156
§ D.5.3.4 按外部引用报告 (/RX)·····	156
§ D.5.3.5 按模块长度报告 (/RS)·····	157
§ D.5.3.6 按类报告 (/RC)·····	157
§ D.5.3.7 按未引用符号名报告 (/RV)·····	157
§ D.5.3.8 冗长报告 (/RV)·····	158
§ D.5.4 使用 OBJXREF 的例子·····	158
§ D.5.5 OBJXREF 出错信息和警告·····	158
§ D.5.5.1 出错信息·····	158
§ D.5.5.2 警告·····	158
§ D.6 TCREF: 源模块交叉引用实用程序·····	159
§ D.6.1 应答文件·····	159
§ D.6.2 与 TLINK 的兼容·····	159

§ D.6.2.1	开关	160
§ D.6.2.2	全局(或连接器级)报告	160
§ D.6.2.3	局部(或模块级)报告	160
附录 E	出错信息	161
§ E.1	信息性信息	161
§ E.2	警告和出错信息	161
§ E.3	致命错信息	177

第一章 预定义符号

Turbo Assembler 提供了许多可以在程序中使用的预定义符号。这些符号在源文件中的不同地方可能有不同的值。它们类似于使用 EQU 伪指令定义的等价符；每当 Turbo Assembler 在源文件中遇到一预定义符号的，它就用该预定义符号的当前值来取代该符号。

这些预定义的符号包括文本(串)等价符，数等价符和一些等价别名。串预定义值可用于任何能使用字符串的地方，例如用 DB 伪指令来初始化一系列数据字节：

```
NOW DB ??time
```

数预定义值能用于任何可使用一个数的地方：

```
IF ??version GT 100h
```

别名值使预定义符号成为其所代表值的一个同义词，可在任何允许使用普通符号名的地方使用相应预定义符号名：

```
ASSUME cs:@code
```

所有预定义符号均可在 MASM 和 Ideal 两种模式下使用。

如果在汇编时使用了 /ml 选择项，一定要严格地按照以下对这些符号的描述来使用预定义符号。

下面的规则适用于以 at 标志(@)开头的预定义符号：构成符号名一部分的每个词的头一个字母是大写字母(段名除外)；这个词的其余部分是小写字母。例如：

```
@CodeSize
```

```
@FileName
```

```
@WordSize
```

例外是有关段的重定义符号。以 at 标志(@)开头的段名总是由小写字母组成。例如：

```
@curseg
```

```
@fardata
```

以两个问号(??)起始的符号，其字母都是小写的。例如：

```
??data
```

```
??version
```

@code

功能 段名.CODE 的等价别名

说明 当使用简化的段伪指令(如.MODEL 等)时，该等价名允许你 ASSUME 和段重载等表达式中使用代码段名。

例子：.CODE

```
MOVE ax , @code
```

```
MOVE dx , ax
```

```
ASSUME ds:@code
```


@CodeSize

功能 表明代码存贮模型的数等价符。

说明 对使用近程(near)代码指针的小型 and 压缩模型, @CodeSize 被置成 0, 对其它所有使用远程(far)代码指针的存贮模型则将其设置为 1。

可用该符号来控制指向函数的指针按存贮模型来进行汇编。

例子 IF @CodeSize EQ 0

procptr DW PROC1 ;指向近程过程的指针

ELSE

procptr DO PROC1 ;指向远程过程的指针

ENDIF

@Cpu

功能 返回当前处理器信息的数值等价符。

说明 由 @Cpu 返回的值是将处理器类型按位来编码:

位	描述
0	允许8086指令
1	允许80186指令
2	允许80286指令
3	允许80386指令
7	允许特权指令 (80286和80386)
8	8087处理器指令
10	80287处理器指令
11	80387处理器指令

未定义的位是保留供以后使用的。在使用 @Cpu 时将它们屏蔽掉以使程序和以后的 Turbo Assembler 版本兼容。由于 8086 处理器类是向上兼容的, 当使用如 .286 伪指令来允许一处理器类型时, 则低级的处理器类型 (8086 ,80186) 自动允许使用。

注释: 本等价符只提供在汇编期间使用 .286 及有关伪指令所选择的处理器的信息, 并不指明程序在运行期间的处理器类型。

例子 IPUSH = @Cpu AND 2 ;对 186 以上处理器允许立即数压入栈

IF IPUSH

PUSH 1234

ELSE

mov ax , 1234

push ax

ENDIF