



教育部高职高专规划教材
Jiaoyubu Gaozhi Gaozhuan Guihua Jiaocai

微机接口与汇编语言实训

武 新 编

高等教育出版社
HIGHER EDUCATION PRESS



教育部高职高专规划教材

微机接口与汇编语言实训

武 新 编

高等教育出版社

内容提要

本书是教育部高职高专规划教材,是与教育部高职高专规划教材《汇编语言程序设计》和《微机接口技术》配套的实际训练指导书,旨在提高高职高专计算机专业学生的程序设计能力和实践能力。

本书共分两章,第1章为上机操作所需的软、硬件配置和所需软件的基本使用方法介绍。第2章为实验内容部分,共27个实验,约需40学时。实验内容由浅入深,包括使用汇编语言编写一些基本程序和对硬件接口的编程,还包括C语言对硬件的编程方法。每个实验均给出了实验目的、实验内容,并提出了问题和要求。本书可作为高等职业学校、高等专科学校、成人高等学校及本科院校举办的二级职业技术学院的计算机或其他相关专业学生的教材,还可供程序开发人员或自学者参考。

图书在版编目(CIP)数据

微机接口与汇编语言实训/武新编. —北京:高等教育出版社,2001

教育部高职高专规划教材

ISBN 7-04-009822-9

I. 微… II. 武… III. ①微型计算机—接口—高等学校:技术学校—教材②汇编语言—程序设计—高等学校:技术学校—教材 IV. ①TP364.7②TP313

中国版本图书馆CIP数据核字(2001)第00923号

微机接口与汇编语言实训

武新 编

出版发行 高等教育出版社

社 址 北京市东城区沙滩后街55号

邮政编码 100009

电 话 010-64054588

传 真 010-64014048

网 址 <http://www.hep.edu.cn>

<http://www.hep.com.cn>

经 销 新华书店北京发行所

印 刷 北京二二〇七工厂

开 本 787×1092 1/16

版 次 2001年9月第1版

印 张 6

印 次 2001年9月第1次印刷

字 数 130 000

定 价 8.30元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

出版说明

教材建设工作是整个高职高专教育教学工作中的重要组成部分。改革开放以来,在各级教育行政部门、学校和有关出版社的共同努力下,各地已出版了一批高职高专教育教材。但从整体上看,具有高职高专教育特色的教材极其匮乏,不少院校尚在借用本科或中专教材,教材建设仍落后于高职高专教育的发展需要。为此,1999年教育部组织制定了《高职高专教育基础课程教学基本要求》(以下简称《基本要求》)和《高职高专教育专业人才培养目标及规格》(以下简称《培养规格》),通过推荐、招标及遴选,组织了一批学术水平高、教学经验丰富、实践能力强的教师,成立了“教育部高职高专规划教材”编写队伍,并在有关出版社的积极配合下,推出一批“教育部高职高专规划教材”。

“教育部高职高专规划教材”计划出版500种,用5年左右时间完成。出版后的教材将覆盖高职高专教育的基础课程和主干专业课程。计划先用2~3年的时间,在继承原有高职、高专和成人高等学校教材建设成果的基础上,充分汲取近几年来各类学校在探索培养技术应用性专门人才方面取得的成功经验,解决好新形势下高职高专教育教材的有无问题;然后再用2~3年的时间,在《新世纪高职高专教育人才培养模式和教学内容体系改革与建设项目计划》立项研究的基础上,通过研究、改革和建设,推出一大批教育部高职高专教育教材,从而形成优化配套的高职高专教育教材体系。

“教育部高职高专规划教材”是按照《基本要求》和《培养规格》的要求,充分汲取高职、高专和成人高等学校在探索培养技术应用性专门人才方面取得的成功经验和教学成果编写而成的,适用于高等职业学校、高等专科学校、成人高校及本科院校举办的二级职业技术学院和民办高校使用。

教育部高等教育司

2000年4月3日

前 言

本教材是教育部高职高专规划教材，是与《汇编语言程序设计》和《微机接口技术》两本教材配套的实训课程的指导书。通过本课程的学习，可提高高职高专计算机专业学生的程序设计能力和实际运用计算机的能力。

“汇编语言程序设计”及“微机接口技术”都是实践性很强的课程，只有多实践、多做练习，才能真正掌握微机的原理及接口原理，并为深入学习微机，充分利用、开发和进行微机维护打下坚实的基础。

全书共分两章，第1章为上机操作所必需的软、硬件配置和所需软件的基本使用方法介绍。重点介绍了DEBUG调试程序、Q编辑程序、调试汇编语言程序的步骤和方法以及C语言中的一些函数。第2章实验内容分为两部分，第一部分为汇编语言的基本编程训练，目的是让学生掌握汇编语言编程的基本方法和技巧；第二部分为对接口编程，考虑到在实际应用中经常使用高级语言或高级语言和汇编语言混合编程开发应用程序，为拓宽学生的思路，使其将来能更快适应生产第一线的需要，在实验中除使用汇编语言外，还给出了C语言对接口的编程实例，并给出了C语言调用汇编语言的方法和程序。

教材针对高职高专计算机专业学生的特点，在作者多年教学实践的基础上总结而成。实验内容由浅入深，通俗易懂。每个实验均给出了实验目的、实验内容，并提出了问题和要求，以便让学生在实验前、后进行思考总结。书中所给实例都在PC机上调试通过，并具有一定的代表性。

洛阳工学院的梁文林教授对全稿进行了详尽的审阅，并提出许多宝贵意见，在此表示衷心的感谢。

由于作者水平有限，书中难免有不妥与疏漏之处，敬请读者批评指正。

编 者

2000年12月

目 录

第 1 章 上机操作预备知识	1
1.1 在 PC 系列机上运行汇编源程序所必备的软件	1
1.2 在 PC 系列机上运行汇编源程序的全过程	1
1.2.1 建立与修改汇编源程序文件（注意源文件扩展名必须是.ASM）	1
1.2.2 对源文件汇编产生相应的目标文件（目标文件的扩展名为.OBJ）	2
1.2.3 连接目标文件生成可执行文件（扩展名为.EXE）	5
1.2.4 运行可执行文件（扩展名为 .EXE）	6
1.3 Q 编辑命令	7
1.4 调试程序 DEBUG	7
1.4.1 运行 DEBUG	7
1.4.2 DEBUG 命令的一些共同特点	8
1.4.3 DEBUG 命令	8
1.5 实验中出现的 C 语言函数	11
第 2 章 实验内容	13
2.1 汇编语言基本实验操作方法练习	13
实验一 上机操作方法练习一	13
实验二 上机操作方法练习二	16
2.2 汇编语言基本程序设计实验	19
实验一 求和程序	19
实验二 寻址方式练习	20
实验三 多位 BCD 码加减法	21
实验四 四则混合运算编程	22
实验五 查表转换	23
实验六 查找程序	25
实验七 统计程序	28
实验八 排序程序	31
实验九 子程序	32
实验十 串操作	34
实验十一 宏调用	35
实验十二 模块化程序设计	36
实验十三 子程序库的建立	38
实验十四 综合练习	41

2.3 接口实验	48
实验一 软中断程序.....	48
实验二 硬中断程序（键盘支持程序）	50
实验三 打印驱动程序.....	55
实验四 键盘模拟电子琴.....	57
实验五 时钟驻留程序.....	59
实验六 串行通信.....	63
实验七 直接写 VRAM 画图形	66
实验八 鼠标程序.....	70
实验九 磁盘操作.....	74
实验十 文件操作.....	75
实验十一 C 语言与汇编语言混合编程	79
附 录.....	83
附录 1 汇编错误代码信息表.....	83
1. 错误代码.....	83
2. 未编号的错误信息.....	87
附录 2 IBM PC ASCII 码表	89

第 1 章 上机操作预备知识

CPU 是中央处理器 (Central Processing Unit) 的简称, 由大规模集成电路 (LSI) 制成, 常见的字长为 8 位的 CPU 有 Intel 公司的 8080 和 8085, 16 位字长的 CPU 有 8088, 8086, 80286, 32 位字长的有 80386, 80486, Pentium 等。

CPU 的主要功能是对数据进行运算、判断和传送, 以及对存储器、I/O 端口和 CPU 本身进行控制。本章主要介绍在 PC 系列机上调试、运行汇编语言源程序和用汇编语言及 C 语言对微机接口编程的操作方法和过程。

1.1 在 PC 系列机上运行汇编源程序所必备的软件

在 PC 系列机上运行汇编源程序所需的常用软件有:

- (1) DOS, PDOS 或 UCDOS 等;
- (2) 编辑程序: Q, EDIT, SK, WORDSTAR 等;
- (3) 宏汇编程序: MASM;
- (4) 交叉参考程序: CREF (调试大程序时使用);
- (5) 连接程序: LINK;
- (6) 库管理程序: LIB;
- (7) 调试程序: DEBUG;
- (8) 建立.COM 文件: EXEIBIN (可选项)。

其中 1, 2, 3, 5, 7 是必备的。最好将上述软件拷贝到一个软盘上或硬盘的同一子目录下, 这样使用起来比较方便。本书中, 凡是要求用户从键盘输入的字符, 均在字符下加下划线, 以示区别系统的提示符, 并用 “✓” 表示回车。

1.2 在 PC 系列机上运行汇编源程序的全过程

在 PC 机上运行汇编源程序必须经过下面四步。

1.2.1 建立与修改汇编源程序文件 (注意源文件扩展名必须是.ASM)

A>UCDOS✓

如果源程序中有汉字可先启动汉字系统, 如 UCDOS 等; 没有汉字则不必启动。

A>Q A.ASM✓

进入全屏编辑状态，接着一个字符一个字符地输入源程序。

```

STACK  SEGMENT✓
        DB  200 DUP(0) ✓
STACK  ENDS✓
DATA    SEGMENT✓
BUF     DB  'Hello, How Are You!$'✓
DATA    ENDS✓
CODE    SEGMENT✓
        ASSUME CS:CODE,DS:DATA,SS:STACK✓
START:  MOV  AX, DATA✓
        MOV  DS, AX✓
        LEA  DX, BUF✓
        MOV  AH, 9✓
        INT  21H✓
        MOV  AH, 4CH✓
        INT  21H✓
CODE    ENDS✓
        END  START✓

```

源程序输入完，按 Esc 键激活菜单。选 File 下的 Save 仅存盘不退出；选 Quit 下的 eXit 存盘退出；选 Quit 下的 Quit 不存盘退出。存盘退出后，机器给出 DOS 提示信息。

A>_

此时在盘上已建立了一个文件名为 A.ASM 的文件，用 DIR 命令可以看到。

1.2.2 对源文件汇编产生相应的目标文件（目标文件的扩展名为.OBJ）

命令格式：MASM [Source][,Object][,List][,Cross_ref][;]

其中：中括号中的内容为可选项，Source，Object，List，Cross_ref 位置处分别应打入源程序文件、目标程序文件、列表程序文件和交叉对照文件的盘符和文件名。当使用可选项时，如果四个文件都指定了或虽然没指定全部四个文件名，但却输入了分号，则使用系统默认名，而不再显示任何需要回答的提示信息；若文件名指定不全，也没有以分号结尾，则会出现一些需要回答的提示信息，请求回答。例如：

1. A>MASM A;✓

Microsoft (R) Macro Assembler Version 5.10

Copyright (C) Microsoft Copr 1981,1988 All rights reserved.

47584+288299 Bytes symbol space free

0 Warning Errors

0 Severe Errors

A>_

此命令对 A.ASM 文件进行汇编, 如果没有错误, 则仅产生一个 A.OBJ 文件。如果还需要生成列表文件 A.LST 和交叉文件 A.CRF 则按如下方法键入:

2. A>MASM A ✓

Microsoft (R) Macro Assembler Version 5.10

Copyright (C) Microsoft Copr 1981,1988 All rights reserved.

Object filename [A.OBJ]: ✓

Source listing [NUL.LST]: A✓

Cross-reference [NUL.CRF]: A✓

47584+288299 Bytes symbol space free

0 Warning Errors

0 Severe Errors

A>_

如果汇编后产生的.OBJ, .LST, .CRF 文件不建立在当前盘上或文件名不使用 A 时, 在出现提示符时可键入: 盘符: [路径][文件名]

例如: 在 B 盘上生成 F.OBJ 文件, 则在出现如下提示信息时, 按下述方法回答。

.....

Object filename [A.OBJ]: B:F.OBJ✓

汇编后建立的.LST 文件可用于打印, 以便了解汇编后的情况。

A>TYPE A.LST✓

Microsoft(R)Macro Assembler Version 5.10 6/8/00 20:13:20

Page 1-1

```

1 0000                                STACK SEGMENT
2 0000 00C8[                          DB 200 DUP(0)
3      00
4      ]
5
6 00C8                                STACK ENDS
7 0000                                DATA SEGMENT
8 0000 48 65 6C 6C 6F 2C  BUF DB 'Hello, How are you! $'
9      20 48 6F 77 20 61
10     72 65 20 79 6f 75
11     21 20 24
12 0015                                DATA ENDS
13 0000                                CODE SEGMENT
14                                ASSUME CS:CODE,DS:DATA,SS:STACK
15 0000 B8----R                      MOV AX,DATA
16 0003 8E D8                        MOV DS,AX

```

```

17 0005 8D 16 00 00      LEA DX,BUF
18 0009 B4 09            MOV AH,9
19 000B CD 21            INT 21H
20 000D B4 4C            MOV AH,4CH
21 000F CD 21            INT 21H
22 0011                  CODE ENDS
23                        END

```

Microsoft(R) Macro Assembler Version 5.10 6/8/00 20:13:20

Symbols 1-1

Segments and Groups:

Name	Length	Align	Comine	Class
CODE	0011	PARA	NONE	
DATA	0015	PARA	NONE	
STACK.....	00C8	PARA	NONE	

Symbols:

Name	Type	Value	Attr
BUF.....	L BYTE	0000	DATA
@CPU.....	TEXT	0101H	
@FILENAME.....	TEXT	H	
@VERSION.....	TEXT	51	

17 Source Lines

17 Total Lines

9 Symbols

47594+423089 Bytes symbol space free

0 Warning Errors

0 Severe Errors

A>_

在 Page 1-1 中最左边第一列为行号，第二列为偏移地址，偏移地址右边为各段的值，其中代码段的值为机器指令的代码。其中 15 行的“...R”是由于段的浮动性，使得数据段的首地址无法确定，因而只能使用“...R”表示。

汇编后建立的.CRF 文件不能单独使用，若需要了解源程序中的符号（段名、变量、标号、常量、过程名）在定义和引用时的情况，还要调用 CREF 文件对.CRF 文件进行处理，生成.REF 文件后，才能打印（显示）输出，其过程如下：

A>CREF✓

Microsoft (R) Cross-Reference Utility Version 5.10

Copyright (C) Microsoft Corp 1981,198. All rights reserved.

Cross-reference [.CRF] : A✓

```

Listing [A.REF]: ✓
6 Symbol
A>TYPE A.REF✓
Microsoft Cross-Reference Version 5.10 Wed Jun 08 20:14:18 2000
Symbol Cross-Reference      (# definition,+modification)  Cref
@CPU..... 1#
@VERSION..... 1#
BUF..... 8# 17
CODE..... 13# 14 22
DATA..... 7# 12 14 15
STACK..... 1# 6 14
6 Symbols
A>_

```

文件 A.REF 内加“#”号的数字表示符号定义所在的行，未加“#”号的数字，如 17 或 14, 22 等为该符号所在的引用行。

1.2.3 连接目标文件生成可执行文件（扩展名为.EXE）

命令格式：LINK [Objlst][,Runfile][,Mapfile][,Liblist][;]

连接程序的输入文件是目标文件.OBJ 和库文件.LIB，输出文件是可执行文件.EXE 和地址分配列表文件.MAP，它们之间的关系如图 1-1 所示。

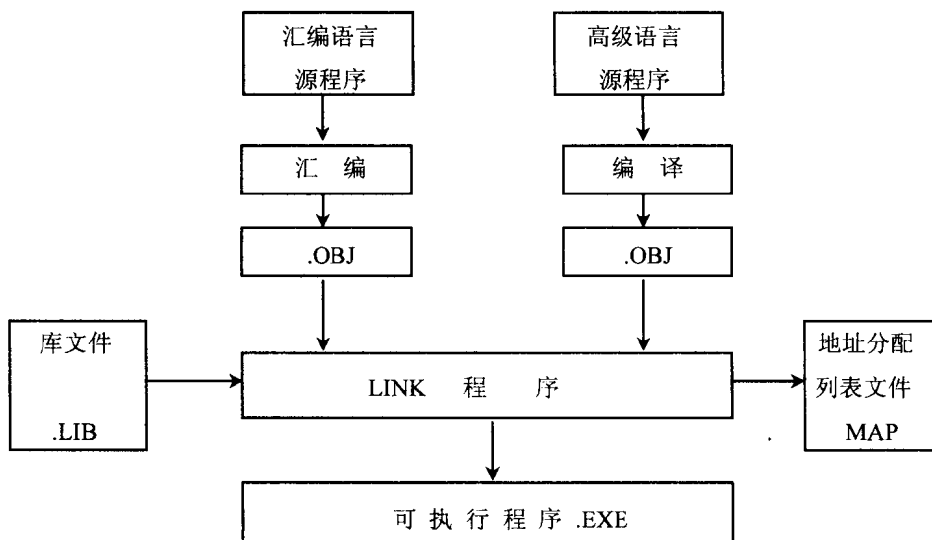


图 1-1 LINK 程序的输入输出文件

1. A>LINK A;✓

Microsoft (R) Overlay Linker Version 3.64

Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

A>_

此命令对 A.OBJ 连接后, 如没有错误, 生成 A.EXE 文件。或者:

2. A>LINK✓

Microsoft (R) Overlay Linker Version 3.64

Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

Object Modules [A.OBJ]:A✓

Run File [A.EXE]:✓

List file [UNL.MAP]:A✓(或✓, 不产生.MAP 文件)

Libraries [A.LIB]:✓

A>_

.MAP 为地址分配列表文件, 如不需要时, 可直接按✓。 .MAP 的内容如下:

A>TYPE A.MAP✓

Start	Stop	Length	Name	Class
00000H	000C7H	000C8H	STACK	
000D0H	000E4H	00015H	DATA	
000F0H	00110H	00011H	CODE	

A>_

如果程序由多个模块: A.OBJ, B.OBJ, C.OBJ 组成, 则可用 LINK 命令生成一个可执行文件。

A>LINK A+B+C✓

Microsoft (R) Overlay Linker Version 3.64

Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

A>_

生成的可执行文件名为 A.EXE

1.2.4 运行可执行文件 (扩展名为 .EXE)

经过 1.2.1, 1.2.2, 1.2.3 后均无错误, 即可在 DOS 状态下直接打入文件名运行程序。

A>A✓

Hello, How Are You!

A>_

设计一个较复杂、较长的汇编源程序时, 免不了会出现这样或那样的错误, 很难一次通过。这时, 就需要使用调试程序 Debug 帮助我们对源程序进行调试, 具体调试方法见 1.4。在上述四步中, 不论哪一步出现错误都应回到第一步重新修改源程序, 然后再进行 1.2.2, 1.2.3, 1.2.4 步的操作。

1.3 Q 编辑命令

(1) 在 DOS 状态下键入: Q <文件名> 就进入全屏幕编辑状态, 如

A>Q A.ASM✓

- (2) 在编辑状态下, 即可逐个字符地直接输入源程序。
- (3) 按<Esc>键激活菜单, 可移动光标进行选择, 再按一次回到原状态。
- (4) 按<F1>键给出帮助信息。
- (5) <F7>定义块头, <F8>定义块尾。
- (6) <Ctrl>+K,C 拷贝块, <Ctrl>+K,V 移动块。
- (7) <Ctrl>+Y 删除光标所在的行。
- (8) <Ctrl>+U 恢复刚删除的行。
- (9) <Ctrl>+N 在光标所在位置插入一空行。
- (10) <Insert>进入插入状态, 再按一次进入改写状态, 交替改变。
- (11) 删除一个字符。
- (12) <F2>文件存盘。
- (13) <F4>复制当前行。
- (14) <Alt>+X 文件存盘退出。
- (15) <Alt>+Q 文件不存盘退出。
- (16) ↑、↓、←、→移动光标。
- (17) DOS 状态下看命令的帮助信息, 可键入: <文件名>/?

例: A>DEBUG/?✓

Run Debug, a program testing and editing tool.

DEBUG [[drive:][path]filename [testfile-parameters]]

[drive:][path]filename Specifies the file you want to test.

testfile-parameters Specifies command-line information required by the file you want to test.

After Debug starts, type? to display a list of debugging commands.

(18) 在 Debug 状态下看帮助信息可使用?G, ?U 等命令。

1.4 调试程序 DEBUG

1.4.1 运行 DEBUG

DEBUG 程序可用来显示、修改或执行一个文件, 是调试汇编语言程序的重要工具。在 DOS 状态下键入 DEBUG [文件名.扩展名]命令, 即可进入 DEBUG 状态, 如

A>DEBUG A.EXE✓

“_”是DEBUG提示符。DEBUG只能对扩展名为.COM, .EXE的文件进行调试。

1.4.2 DEBUG 命令的一些共同特点

- (1) 命令及参数可以是大写、小写或两者的组合。
- (2) 只有按下✓键, 命令才有效。
- (3) 按下<Ctrl>+<Break>键, 则停止命令的执行, 返回DEBUG的提示符“_”。
- (4) 如果命令产生大量输出, 可按下<Ctrl>+<Num Lock>键, 使显示停止滚动以便阅读, 再按其他键时继续滚动显示。
- (5) 所有输入和显示的数, 均为十六进制数, 其后不跟H。

1.4.3 DEBUG 命令

1. A 汇编命令

功能: 把IBM PC的宏汇编语句直接汇编到内存。

格式: A [地址]

说明: 键入的汇编语句被汇编到地址指定的相应内存单元中, 如没有指定地址, 则被汇编到CS: 0100内存区内。键入所有语句后, 当再次出现提示符时, 按下✓键, 返回到DEBUG提示符。

2. C 比较命令

功能: 比较两个内存块的内容。

格式: C [源地址范围][目标地址]

说明: 比较两个内存块, 如果内容不同则显示出两者的地址和内容。地址范围为: 起始地址 结束地址, 或: 起始地址 L 处理的字节数。

3. D 显示内存命令

功能: 显示部分内存的内容。

格式: D [地址]/[范围]

说明: 显示分两部分, 一部分是字节的十六进制数形式, 另一部分是ASCII码形式, 不可见字符用“.”表示。

4. E 修改内存命令

功能: 修改内存单元的内容。

格式: E 地址 [内容表]

说明: 具有两种操作: 1) 用内容表中的内容替换指定地址的一个或多个字节的内容; 2) 允许顺序显示和修改字节内容, 不修改的可按空格。如只键入一个地址偏移量, 则缺省的段在DS中。按✓键结束命令。

5. G 执行程序命令

功能：执行正在调试的程序，当执行到指定断点时，停止执行，并显示寄存器、标志和下一条要执行的指令。

格式：G [=地址] [地址 [地址...]]

说明：[=地址]规定了执行的起始地址，缺省时则以 CS: IP 当前内容为开始地址，输入时要特别注意加“=”号。第二、三个参数的地址为断点地址，最多可设 10 个断点，每当程序执行遇到断点时都会停下来，键入 G 继续执行。

6. H 十六进制运算命令

功能：计算两个十六进制数的和与差。

格式：H 值1 值2

例如：H 0F 8✓

17 07

7. I 输入命令

功能：从指定端口输入一个字节，并显示之。

格式：I 端口地址

例如：I 2F8✓

6B （表示 2F8 端口的内容为 6B）

8. L 装入内存命令

功能：把文件或磁盘扇区装入内存。

格式：L [地址 [驱动器 扇区号 扇区数]]

说明：驱动器：0 表示 A 盘，1 表示 B 盘，2 表示 C 盘，依此类推。一条 L 命令能装入的最大扇区数是 80H。地址表示装入内存的起始地址，缺省时装入 CS: 100 开始的内存。

L 命令把文件装入内存时，起始内存地址由命令中的地址给出，常与 N 命令联合使用。例如：N MYPROG✓

L✓

结果为从缺省磁盘把名为 MYPROG 的文件取出并存入 CS: 100 开始的内存单元中。

9. N 指定名字命令

功能：为待调试程序提供所需要的文件名。

格式：N [盘符:] [路径] 文件名[.EXE]

说明：若在启动 DEBUG 程序时没有指定文件名，那么在 L 命令装入文件之前必须先使用 N 命令。

10. O 输出命令

功能：把字节发送到指定的输出端口。

格式: O [端口地址] [字节]

例如: O 2F8 4F✓

把字节 4F 发送到输出端口 2F8。

11. P 处理命令

功能: 在下一条指令前停止。

格式: P [=地址] [值]

说明: 执行子程序, 循环命令, 中断或重复字符串命令后, 发送一条 P 命令返回到下一条命令处, 其语法与 T 命令相同。

12. Q 退出命令

功能: 结束 DEBUG 程序, 返回 DOS。

格式: Q

说明: 用 Q 命令时内存中正在运行的文件不存盘, 必须用 W 命令才能保存。

13. R 寄存器命令

功能: 显示和修改寄存器内容和标志。

格式: R 寄存器名

说明: 显示单个寄存器内容, 例如: R AX✓

显示全部寄存器内容, 例如: R✓

显示标志寄存器内容, 例如: RF✓

标志为:	溢出	方向	中断	符号	零	辅助进位	奇偶	进位
置位	OV	DN	EI	NG	ZR	AC	PE	CY
复位	NV	VP	DI	PL	NZ	NA	PO	NC

14. T 跟踪命令

功能: 逐条跟踪程序。

格式: T [=地址] [值]

说明: 执行自指定地址开始的一条或给定值条指令, 每条指令执行完都会显示出所有寄存器和标志位的内容, 未给定地址则以 CS:IP 开始执行。

15. U 反汇编命令

功能: 从指定的地址反汇编 32 字节的汇编语句, 同时, 显示出它们的地址、十六进制值以及对应的汇编语句。

格式: U [地址]/[范围]

说明: 地址、范围均缺省时, 则以前条 U 命令反汇编的最后一条指令后的单元作为起始地址, 若以前未用过 U 命令则以 CS: 100 作为起始地址。

16. W 写入磁盘命令

功能: 把调试的文件写入磁盘。