

可下载教学资料

<http://www.tup.tsinghua.edu.cn>

21世纪高等学校计算机**基础**实用规划教材

计算机硬件 技术基础（第二版） 实验与实践指导

曹岳辉 李 力 主编
陈彦伶 杨莉军 副主编

清华大学出版社



TP360.3/43=2C

2008

21世纪高等学校计算机**基础**实用规划教材

计算机硬件技术基础

(第二版)

实验与实践指导

曹岳辉 李 力 主编
陈彦伶 杨莉军 副主编

清华大学出版社
北京

内 容 简 介

本书是计算机入门课程《计算机硬件技术基础(第二版)》的配套实验教程。全书包含 17 个基本实验, 7 个综合实验, 并给出了部分习题的参考答案。

本书突出基础知识的复习与巩固, 注意程序设计能力的培养和提高。本书适用于大学非计算机专业学生学习计算机硬件技术, 也适用于教师和工程技术人员作为参考。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

计算机硬件技术基础(第二版)实验与实践指导/曹岳辉, 李力主编. —北京: 清华大学出版社, 2008. 7

(21 世纪高等学校计算机基础实用规划教材)

ISBN 978-7-302-17639-8

I. 计… II. ①曹… ②李… III. 硬件—高等学校—教学参考资料 IV. TP303

中国版本图书馆 CIP 数据核字(2008)第 073247 号

责任编辑: 魏江江 李 晔

责任校对: 李建庄

责任印制: 杨 艳

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

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

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者: 北京国马印刷厂

经 销: 全国新华书店

开 本: 185×260 印 张: 11.5 字 数: 277 千字

版 次: 2008 年 7 月第 1 版 印 次: 2008 年 7 月第 1 次印刷

印 数: 1~4000

定 价: 19.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010)62770177 转 3103 产品编号: 027907-01

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

本系列教材立足于计算机专业课程领域,以专业基础课为主、专业课为辅,横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

(1) 反映计算机学科的最新发展,总结近年来计算机专业教学的最新成果。内容先进,充分吸收国外先进成果和理念。

(2) 反映教学需要,促进教学发展。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,融合先进的教学思想、方法和手段,体现科学性、先进性和系统性,强调对学生实践能力的培养,为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现教学质量和教学改革成果的教材。

(4) 主张一纲多本,合理配套。专业基础课和专业课教材配套,同一门课程有针对不同层次、面向不同应用的多本具有各自内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选用。在制定教材规划时要依靠各课程专家在调查研究本课程教

材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主题。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平教材编写梯队才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪高等学校计算机基础实用规划教材

联系人:魏江江 weijj@tup.tsinghua.edu.cn

前 言

计算机硬件技术基础是大学计算机公共课中的一门重要课程,与计算机软件技术基础同属于计算机基础课程中的第二层次。本书为《计算机硬件技术基础(第二版)》配套的实验教程。

全书共分4章。第1章阐述了汇编语言上机环境;第2章提供17个基本实验;第3章提供7个综合实验;第4章给出了部分习题参考答案。

本书是作者总结计算机硬件技术基础的教学实践经验编写而成的。为配合计算机基础教学新一轮的“1+X”课程体系的改革,根据初学者特点,按照由浅入深,循序渐进的方式编写。全书内容丰富,示例较多,并给出了部分参考答案。

本教材由曹岳辉、李力任主编,陈彦伶、杨莉军任副主编。实验从选材到编写,都是集体讨论的结果,凝结了集体的智慧。参与本书编写的有曹岳辉、陈彦伶、杨莉军、李力、王晓玲、张葆青,全书由曹岳辉整理统稿,蒋外文教授任主审。

本书的编辑出版得到了许多同行专家、教师的支持,在此表示感谢。还要感谢清华大学出版社的工作人员对本书所做的大量工作。由于编者的水平有限和时间紧迫,因此错误和问题在所难免,真诚恳请批评指正。

编者

2008年5月于中南大学

相关课程教材推荐

ISBN	书 名	定价(元)
9787302116981	软件技术基础教程	33.00
9787302161837	嵌入式技术基础与实践	39.00
9787302167334	高级语言程序设计与应用教程	29.00
9787302172574	计算机网络管理技术	25.00
9787302168003	计算机组成与系统结构	34.00
9787302109013	微机原理、汇编与接口技术	28.00
9787302142867	XML 实用技术教程	25.00
9787302167327	微机组成与组装技术及应用教程	29.50
9787302119715	计算机硬件技术基础	23.00
9787302147640	汇编语言程序设计教程(第2版)	28.00
9787302131755	Java 2 实用教程(第三版)	39.00
9787302142317	数据库技术与应用实践教程——SQL Server	25.00
9787302143673	数据库技术与应用——SQL Server	35.00

以上教材样书可以免费赠送给授课教师,如果需要,请发电子邮件与我们联系。

教学资源支持

敬爱的教师:

感谢您一直以来对清华版计算机教材的支持和爱护。为了配合本课程的教学需要,本教材配有配套的电子教案(素材),有需求的教师可以与我们的联系,我们将向使用本教材进行教学的教师免费赠送电子教案(素材),希望有助于教学活动的开展。

相关信息请拨打电话 010-62776969 或发送电子邮件至 weijj@tup.tsinghua.edu.cn 咨询,也可以到清华大学出版社主页(<http://www.tup.com.cn> 或 <http://www.tup.tsinghua.edu.cn>)上查询和下载。

如果您在使用本教材的过程中遇到了什么问题,或者有相关教材出版计划,也请您发邮件或来信告诉我们,以便我们更好为您服务。

地址:北京市海淀区双清路学研大厦 A 座 708 计算机与信息分社魏江江 收
邮编:100084 电子邮件:weijj@tup.tsinghua.edu.cn
电话:010-62770175-4604 邮购电话:010-62786544

目 录

18	同知已眼日示显	4	区综合综
88	画成小出空可	2	区综合综
101	示显燃燃以地景	8	区综合综
133	(二)示显燃燃以地景	7	区综合综
131	案答学考	章	4
131	键实	2	
138	键实	3	
第1章 实验环境			1
1.1 调试程序 DEBUG			1
1.1.1 DEBUG 程序的调用			1
1.1.2 DEBUG 的主要命令			1
1.2 汇编系统 MASM			7
1.2.1 汇编系统的组成			7
1.2.2 汇编系统的应用			7
第2章 实验内容			11
实验 1 DEBUG 调试程序			11
实验 2 MASM 编辑与调试			16
实验 3 汇编语言分支程序设计			19
实验 4 汇编语言循环程序设计			23
实验 5 汇编语言子程序设计			26
实验 6 汇编语言中断调用			28
实验 7 TDN86/88 教学系统认识及程序设计			31
实验 8 8255A 输入输出实验			33
实验 9 8255A 并行口实验——键盘实验			35
实验 10 8255A 并行口实验——交通灯实验			37
实验 11 8259A 的中断实验(一)			39
实验 12 8259A 的中断实验(二)			41
实验 13 8259A 的中断实验(三)			41
实验 14 8254 定时/计数器应用实验(一)			42
实验 15 8254 定时/计数器应用实验(二)			43
实验 16 8254 定时/计数器应用实验(三)			44
实验 17 8 位 A/D 转换器 ADC0809 实验			45
第3章 综合实验			46
综合练习 1 电子琴的设计			46
综合练习 2 学习成绩管理			58
综合练习 3 打字练习			72

综合练习 4	显示日期与时间	81
综合练习 5	可爱的小动画	85
综合练习 6	霓虹灯模拟显示(一)	101
综合练习 7	霓虹灯模拟显示(二)	122
第 4 章 参考答案	137
实验 2	137
实验 3	138
实验 4	141
实验 5	143
实验 6	152
实验 7	156
实验 8	157
实验 10	158
实验 11	160
实验 12	161
实验 13	163
实验 14	165
实验 15	166
实验 16	167
实验 17	168
附录 汇编程序出错信息	169
参考文献	174

第 1 章

实验环境

1.1 调试程序 DEBUG

调试程序 DEBUG 是 DOS 提供的用于调试可执行程序的一个工具软件,它在汇编语言程序设计的學習过程中是最常用的调试工具。运行 DEBUG 程序后,用户可以通过 DEBUG 的命令输入汇编源程序,并通过单步、设置断点等方式提供非常有效的调试手段。

1.1.1 DEBUG 程序的调用

用户在不同操作系统下均可调用 DEBUG 程序。在 Windows 的“开始”菜单下运行 DEBUG 命令,或在 DOS 提示符下输入命令:

```
DEBUG[盘符][路径][文件名][参数 1][参数 2]
```

若 DEBUG 带有被调试程序的文件名,则将指定程序调入主存。当被调试程序为 EXE 文件时, BX, CX 包含被调试文件大小的字节数(BX 为高 16 位); CS:IP 和 SS:SP 根据程序设定分别指向代码段和栈段; DS=ES 指向当前可用的主存段,其他通用寄存器为 0; 状态标志全为 0。若被调试程序不为 EXE 文件,则 BX, CX 包含被调试文件大小的字节数(BX 为高 16 位); 所有段寄存器值相等,都指向当前可用主存段; SP 指示当前栈顶在这个段的尾部,其他通用寄存器均为 0; IP=0100H; 状态标志全为 0。

若 DEBUG 不带文件名,则可在需要时再用 N、L 命令调入被调试程序。此时所有段寄存器值相等,指向当前可用主存段; 除 SP 外,其他通用寄存器均为 0, SP 指示当前栈顶在此段尾部; IP=0100H; 状态标志全为 0。

DEBUG 程序调入后,根据有无被调试程序及其类型相应设置寄存器组的内容,在出现提示符“—”后,即可用 DEBUG 命令来调试程序。

1.1.2 DEBUG 的主要命令

在 DEBUG 提示符“—”后输入“?”并回车,将显示如图 1-1-1 所示的所有 DEBUG 指令。

DEBUG 命令均为一个字母后跟一个或多个参数。命令字母不区分大小写,命令和参数间可无分隔符; 参数只使用十六进制数,不带后缀字母。每个命令以回车确定生效,以 Ctrl+C 或 Ctrl+Break 组合键中止指令执行。若命令不符合 DEBUG 指令规范,则以 Error 提示并以“^”指示错误位置。DEBUG 命令有固定格式,其基本命令的格式及功能如表 1-1-1 所示。

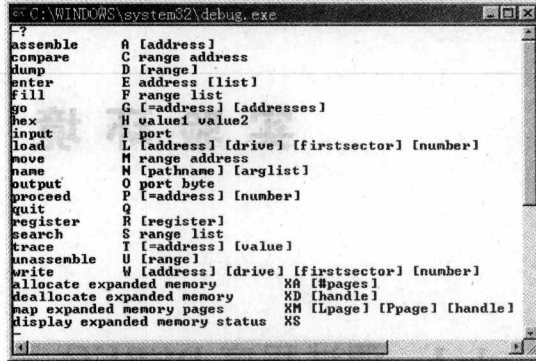


图 1-1-1 DEBUG 的主要命令

表 1-1-1 DEBUG 基本命令表

指 令	格 式	功 能	实 例
?	?	显示帮助信息	?
Assemble	A[地址]	汇编命令	A 100:20
Compare	C[区段 1][范围][区段 2]	比较两个内存块	C 100 L2 200
Dump	D[地址][范围]	显示内存内容	D 100 120
Enter	E 地址[数据表]	编辑内存内容	E 100 'XY',2
Fill	F 范围 数据表	修改内存内容	F 200 L8 9,'XYZ'
Go	G[=地址][,断点 1,...,断点 10]	带有任选断点执行	G 100
Hex	H 数字 1 数字 2	十六进制加减运算	H 44 33
Input	I 端口	指定端口输入/显示 1 个字节	I 70
Load	L[地址][驱动器][扇区号][扇区数]	装入文件或扇区	L 100 2 1 2
Move	M 范围 地址	传送内存区块	M 100 L10 110
Name	N[路径][文件名 1,...,文件名 n]	定义文件及参量	N pass.com
Output	O 端口 字节数据	输出字节到端口	O 71 10
Proceed	P[=地址][数值]	在下一指令上停止	P 2
Quit	Q	退出 DEBUG	Q
Register	R[寄存器/F]	显示修改寄存器/标志	R AX
Search	S 范围 数据	对字符进行检索	S 100 L70 'j'
Trace	T[=地址][数值]	单步追踪	T
Unassemble	U[地址/范围]	对指令进行反汇编	U 100
Write	W[地址][驱动器][扇区号][扇区数]	写入文件或扇区	W 100 0 1 1

1. 显示命令 d(dump)

d 命令用来显示主存储单元的内容,格式为:d[地址][范围]。地址参数表示指定存储单元的起始地址;范围参数有两种表示形式:一种是[起始地址 结束地址],另一种为[起始地址 L 字节长度]。未指定段地址时,d 命令默认显示 DS 段的内容。例:

- d 100 120 ; 显示数据段 ds: 100h~120h 的主存储单元内容

如图 1-1-2 所示,左边 0B37:0100 表示存储单元的“段基地址:偏移地址”;中间为十六进制数表示的每个字节单元的内容;右边为 ASCII 字符表示的每字节单元内容,“.”表示

不可显示的字符。

```

C:\WINDOWS\system32\debug.exe
-d 100 120
0B37:0100 E5 4E 5F 9D F9 C3 4E EB-51 80 CF 01 81 CD 00 80   .N...N.Q.....
0B37:0110 EB DA E1 46 EB AC DF 74-0D EB 45 00 34 00 26 0B   ...F...t..E.4.&.
0B37:0120 81
  
```

图 1-1-2 数据段 ds:100h~120h 内容

- d100 L20 ;显示 ds:100h 开始的 20h 个存储单元的内容(如图 1-1-3 所示)

```

C:\WINDOWS\system32\debug.exe
-d 100 L20
0B37:0100 E5 4E 5F 9D F9 C3 4E EB-51 80 CF 01 81 CD 00 80   .N...N.Q.....
0B37:0110 EB DA E1 46 EB AC DF 74-0D EB 45 00 34 00 26 0B   ...F...t..E.4.&.
  
```

图 1-1-3 ds:100h 开始的 20h 个数据

- d 102 ;省略范围参数,显示首地址 102h 开始的 80h 个单元的内容(如图 1-1-4 所示)

```

C:\WINDOWS\system32\debug.exe
-d 102
0B37:0100          5F 9D F9 C3 4E EB-51 80 CF 01 81 CD 00 80   ...N.Q.....
0B37:0110 E8 DA E1 46 EB AC DF 74-0D EB 45 00 34 00 26 0B   ...F...t..E.4.&.
0B37:0120 40 EB CE E9 2B 00 EB B4-DF 06 57 51 0E 07 BF A3   ...E.4k.u.....
0B37:0130 8F B9 06 00 81 CD 00 40-F2 AE 75 0B 81 E5 FF BF   ...;.....S&.
0B37:0140 B8 01 00 D3 E0 0B EB 59-5F 07 B0 00 AA 5F 9D F8   ...&:s...i.#...6
0B37:0150 C3 A0 41 FE 06 EB 99 C3-2E C7 06 55 91 00 00 2E   W.SWU...U.u.
0B37:0160 89 0E DF 91 2E 89 26 E1-91 2E 89 36 E3 91 FC 2E   C...k.u.e?..u
0B37:0170 89 0E
0B37:0180 89 0E
  
```

图 1-1-4 ds:102h 开始的 80h 个字节

- d ;省略参数,则以上次 d 命令显示的结束存储单元作为起始单元,显示连续 128 个字节的內容(如图 1-1-5 所示)

```

C:\WINDOWS\system32\debug.exe
-d
0B37:0180          48 91 2E C7 06 4A-91 00 00 2E C7 06 5D 91   H...J.....J.
0B37:0190 00 00 2E C7 06 4E 91 00-00 2E C7 06 1A 92 5B 5D   ...N.....[
0B37:01A0 2E C7 06 1C 92 7C 3C 2E-C7 06 1E 92 3E 2B 2E C7   ...k.....
0B37:01B0 06 20 92 3D 3B E8 83 09-73 13 B8 FF FF 53 26 8B   ...;.....S&.
0B37:01C0 1D 26 3A 0F 73 03 B8 02-00 5B E9 23 01 2E 89 36   ...&:s...i.#...6
0B37:01D0 57 91 53 57 55 8D 1E 5F-91 2E F6 06 56 91 20 75   W.SWU...U.u.
0B37:01E0 43 AC E8 0F 0A 72 3C E8-77 09 74 37 EB A4 89 75   C...k.u.e?..u
0B37:01F0 18 2E F6 06 56 91 20 75-05 EB 3F 09 EB 26 2E F6   .U.U.u.?.&..
0B37:0200 06 56
  
```

图 1-1-5 ds:182h 开始的 80h 个字节

- d cs: 0 ;显示指定代码段的存储单元内容(如图 1-1-6 所示)

```

C:\WINDOWS\system32\debug.exe
-d cs:0
0B37:0000 CD 20 FF 9F 00 9A EE FE-1D F0 4F 03 9B 05 8A 03   .....0.....
0B37:0010 9B 05 17 03 9B 05 1C 04-01 01 01 00 02 FF FF FF   .....N.....
0B37:0020 FF FF FF FF FF FF FF FF-FF FF FF FF 2F 05 4E 01   [.....7.....
0B37:0030 5B 0A 14 00 18 00 37 0B-FF FF FF FF 00 00 00 00   .....
0B37:0040 05 00 00 00 00 00 00-00 00 00 00 20 20 20   .....
0B37:0050 CD 21 CD 00 00 00 00-00 00 00 00 20 20 20   .....
0B37:0060 20 20 20 20 20 20 20-20 20 20 20 20 20 20   .....
0B37:0070 20 20 20 20 20 20 20-20 20 20 20 20 20 20   .....
  
```

图 1-1-6 代码段 0h 开始的 128 个字节单元

2. 修改命令 e(enter)

修改命令用于修改主存储单元的内容,格式为: e 地址[数据表]。
无数据表格式为逐个单元相继修改,直至回车结束。

- e cs:10 ;修改代码段 10h 起始的单元

系统显示指定代码段地址 10h 处的内容为 9B。若不需要修改,直接按空格键接着显示下一单元内容,或按“—”键显示上一单元内容;若需要修改,则在“.”后输入欲填入的值。这样用户可以不断修改相邻单元的内容,直到按回车键结束。此例中修改连续的 5 个单元内容为 01、02、03、04、05,如图 1-1-7 所示。修改完成后用 d 命令查看 10h 开始的 10 个单元,与图 1-1-6 对比可知,代码段 10h~14h 已被修改,其他部分没有变化。

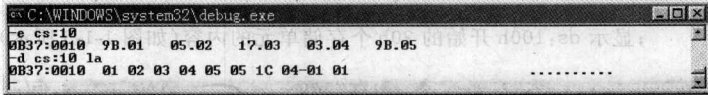


图 1-1-7 代码段 10h 开始的单元

带数据表格式为一次修改多个单元的方式,以指定数据表的内容依次修改指定存储单元为首地址的多个连续单元。此例数据表为: F4'abc',其中 F4、a、b、c 各占一个字节。用 d 命令显示可知, e 命令利用数据表修改了代码段 15h~18h 的 4 个单元,如图 1-1-8 所示。

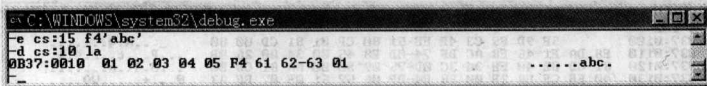


图 1-1-8 利用数据表修改多个连续单元

3. 寄存器命令 r(register)

r 命令用于显示和修改处理器 CPU 的寄存器,格式为: r[寄存器/f]。

- r ;显示所有寄存器内容和标志状态(如图 1-1-9 所示)

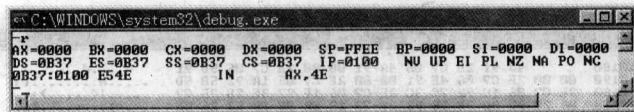


图 1-1-9 寄存器和标志状态

- r ax ;修改寄存器 ax 的内容

系统显示 ax 寄存器当前值为 0,并显示“:”。若需要修改,则输入欲修改的内容,回车确定生效,否则直接回车放弃修改(如图 1-1-10 所示)。

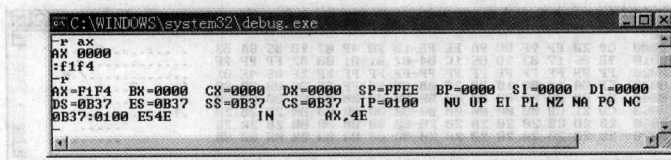


图 1-1-10 寄存器 ax 内容变化

- rf ;显示或修改标志位状态

系统显示各标志位状态,如需要修改,直接在“—”后按任意顺序输入需要修改的值,否则回车放弃修改(如图 1-1-11 所示)。

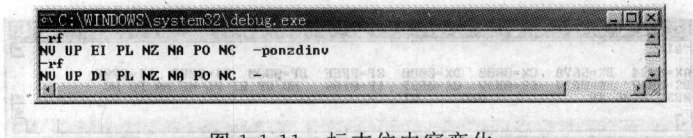


图 1-1-11 标志位内容变化

4. 汇编命令 a(assemble)

输入 a 命令后,可输入汇编指令,DEBUG 将它们汇编成机器代码,相继存放在指定地址开始的存储区中。格式为: a[地址]。

输入 a 命令后,系统显示如 0B37:0100 形式的段基地址: 偏移地址,等待用户输入新指令。输入指令后回车,出现新的一行,可继续输入指令直到回车结束汇编,返回 DEBUG 提示符状态。如图 1-1-12 所示的汇编命令还可以写为: -a 0B37:0100,与 -a100 效果相同。

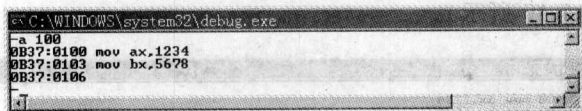


图 1-1-12 汇编命令

若 a 命令后省略地址参数,则上次 a 命令的最后一个存储单元为起始单元;若还未使用过 a 命令,则以当前 CS:IP 作为起始地址。

5. 反汇编命令 u(unassemble)

u 命令用于对指定地址或范围的存储单元进行反汇编,格式为: u[地址/范围]。

若 u 命令中指定了地址参数,则从指定地址开始反汇编 32 个字节的指令;若不带参数,则接着上一个 u 命令的最后一个单元开始反汇编 32 个字节;若没有使用过 u 命令,则从当前 CS:IP 开始反汇编。若 u 命令中指定了范围,则对指定范围内的存储单元进行反汇编。两种用法的反汇编命令执行结果如图 1-1-13 所示。

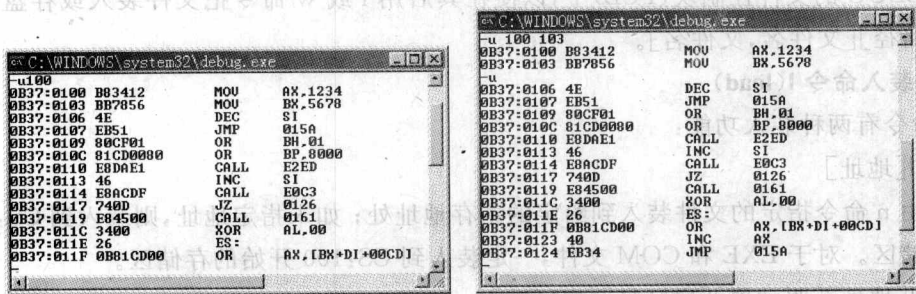


图 1-1-13 反汇编命令

6. 运行命令 g(go)

g 命令用于从指定地址处开始运行程序,直到遇到断点或程序正常结束。命令可以设置 0~10 个断点,格式为: g[=地址][,断点 1,...,断点 10]。

等号后的地址指定程序段运行的起始地址,如省略地址则从当前 CS:IP 开始运行。断点地址如果只有偏移地址,则默认是代码段 CS(如图 1-1-14 所示)。

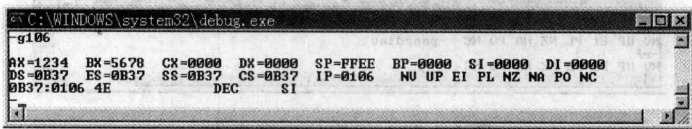


图 1-1-14 运行 0B37:0100~0B37:0106 处指令的结果

7. 跟踪命令 t(trace)

从指定地址起执行一条或多条指令后停下,显示所有寄存器内容、标志位状态及下一条指令。格式为: t[=地址][数值]。

如未指定地址,则从当前 CS:IP 开始执行。t 跟踪指令提供了一种逐条指令运行程序的方法,因此也被称为单步命令。t 命令利用处理器的单步中断,使用户可以细致地观察程序的执行情况(如图 1-1-15 所示)。

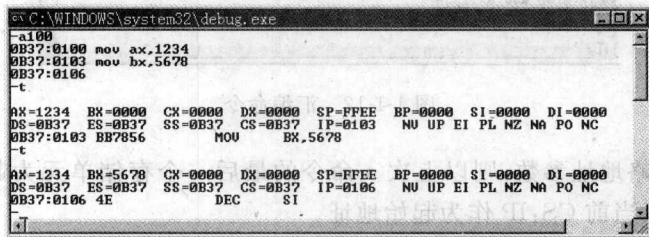


图 1-1-15 t 命令实例

t 命令及 g 命令必须指向正确的指令代码,否则会出现如死机等不可预测的结果。

8. 命名命令 n(name)

n 命令把一个或两个文件标识符(包含路径的文件全名)格式化在 CS:5CH 和 CS:6CH 两个 DEBUG 的文件控制块(FCB)中,以便在其后用 l 或 w 命令把文件装入或存盘。格式为: n[路径][文件名,文件名]。

9. 装入命令 l(load)

l 命令有两种装入功能:

1) l[地址]

将由 n 命令指定的文件装入到给定的主存地址处;如未指定地址,则装入到 CS:100 开始的存储区。对于 EXE 和 COM 文件,一定装入到 CS:100 开始的存储区。

2) l[地址][驱动器][扇区号][扇区数]

装入指定磁盘扇区范围的内容到指定的主存地址处,默认段地址是 CS。其中,驱动器 0 表示 A,1 表示 B,2 表示 C,...。例如,将硬盘 C 的 DOS 引导扇区内容装入 CS:100,然后查看,可使用下列命令:

-l 100 2 0 1

-d CS: 100

10. 写入命令 w(write)

与 l 命令一样,w 命令也有两种形式:

1) w[地址] 将指定起始地址的数据写入由 n 命令指定的文件；如未指定地址则从 CS:100 开始。写入文件的字节数先放入 BX、CX 中。用 w 命令保存的可执行程序的扩展名应为 COM，不能写入 EXE 和 HEX 扩展名的文件。

2) w[地址][驱动器][扇区号][扇区数]

将指定地址的数据写入磁盘的若干扇区(最多 80h)，若未指定段落地址则默认为 CS。

11. 退出命令 q(quit)

q 命令使 DEBUG 程序退出，返回 DOS。命令无存盘功能，如需存盘应先使用 w 命令。

1.2 汇编系统 MASM

1.2.1 汇编系统的组成

汇编系统 MASM 包括多个基本程序和文件，如表 1-2-1 所示。

表 1-2-1 MASM 中的程序和文件

文件名	说明	文件名	说明
ML.EXE	汇编程序	DOSXNT.EXE	MS-DOS 扩展文件
ML.ERR	汇编错误信息文件	LIB.EXE	子程序库管理文件
LINK.EXE	连接程序		

汇编程序 ML.EXE 将编辑好的汇编语言源程序文件(扩展名为 ASM)转换为 OBJ 模块文件，然后用连接程序 LINK.EXE 将一个或多个目标文件链接成一个可执行文件。

1.2.2 汇编系统的应用

1. 源程序的编辑

汇编语言程序可以用文本编辑器 edit、ultraedit 或者直接采用记事本 Notepad.exe 编辑，保存为扩展名为 .asm 的汇编语言源程序文件。

例：编辑文件名为 hello.asm 的源程序，以下为其源程序代码(简化段定义的程序格式)。

```

.model small ;小型程序一般采用小存储模式 small
.stack ;定义堆栈段
.data ;定义数据段
string db 'Hello,Everybody!',0dh,0ah,'$' ;在数据段定义需要显示的字符串
.code ;定义代码段
.startup ;说明程序起始点
mov dx,offset string ;指定字符串在数据段的偏移地址
mov ah,9
int 21h ;利用功能调用显示信息
.exit 0 ;程序结束点
end ;汇编结束
    
```

2. 源程序的汇编

汇编将源程序翻译成由机器代码组成的目标文件(.OBJ)。MASM 6.15 提供的汇编程

序是 ML.EXE,其命令带有许多参数。输入命令: ml /?,如图 1-2-1 所示。

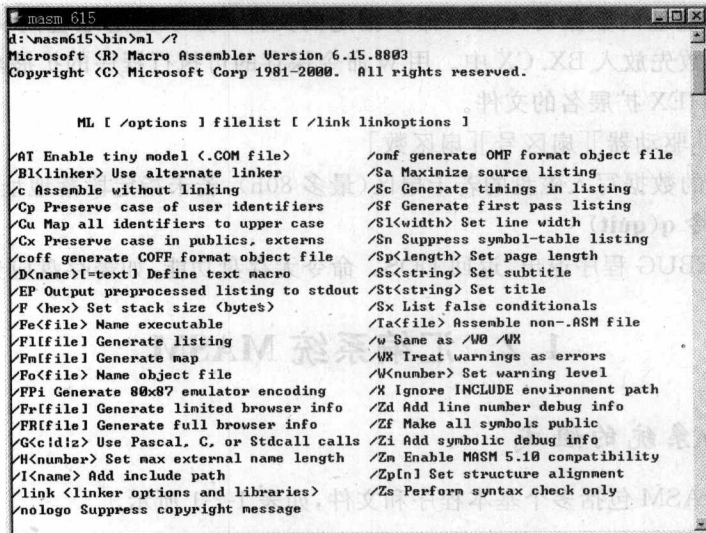


图 1-2-1 ML.EXE 程序参数一览

虽然 ML 程序参数众多,但许多并不经常用到。表 1-2-2 为 ML 的常用参数及其功能(区分大小写)。

表 1-2-2 ML 常用参数表

参数选项	功能
/AT	允许 tiny 存储模式(创建一个 COM 文件)
/c	只汇编源程序,不进行自动连接
/Fl 文件名	创建一个汇编列表文件
/Fr 文件名	创建一个可在 PWB 下浏览的. SBR 源浏览文件
/Fo 文件名	根据指定文件名生成模块文件,不采用默认名
/Fe 文件名	根据指定文件名生成可执行文件,不采用默认名
/Fm 文件名	创建一个连接映像文件,扩展名为 MAP
/I 路径名	设置需要包含进(INCLUDE)源程序的文件所在路径
/Sg	在生成的列表文件中列出由汇编程序产生的指令
/Sn	在创建列表文件时不产生符号表
/Zi	生成模块文件时加入调试程序所需要的信息
/Zs	只进行句法检查,不产生任何代码
/LINK	传递给连接程序 LINK 的参数

源程序编辑完成后,保存为扩展名为. ASM 的文件,然后用 ML 程序进行转换。

ml hello. asm /c ; 注意源程序必须带扩展名

如果源程序中没有语法错误,MASM 将自动生成一个目标文件 hello. obj,否则会给出相应错误信息。参数/c 表示仅利用 ML 进行汇编,不自动调用连接程序 LINK.EXE 进行连接。程序执行结果如图 1-2-2 所示。