



《汇编语言程序设计》

实验指导书

（第二版）

金陵科技学院
JINLING INSTITUTE OF TECHNOLOGY

2019/2/1

目录

上机实验要求	1
实验 1 汇编语言开发过程及调试	1
实验 2 数据寻址与传送指令	3
实验 3 数据运算指令	5
实验 4 顺序程序和转移指令	7
实验 5 分支与循环程序设计	9
实验 6 子程序设计	11
实验 7 windows 程序设计	13
实验 8 与 C++ 混合编程	15
可选实验 DOS 应用程序与浮点指令	17
附录	19

上机实验要求

上机前作好充分准备，包括程序框图、源程序代码、开发过程、对运行结果的分析等。

注意及时在硬盘上存储程序，以避免文件丢失。

每个上机实验安排在 2 个学时内完成。

上机后应写出实验报告、并及时提交。

实验 1 汇编语言开发过程及调试

实验序号：1

实验名称：汇编语言开发过程及调试

适用专业：软件工程/信息安全

学时数： 2 学时

一、实验目的和要求

1. 熟练掌握汇编语言的语句格式和程序框架。
2. 汇编语言的开发过程。
3. 理解断点调试、单步调试的作用和方法

二、实验仪器和设备

1. 硬件环境：微机 CPU 486 以上，500MB 以上硬盘，32M 以上内存；
2. 软件环境：装有 MASM 6.15、Debug、LINK、EDIT、CREF.EXE 和 EXR2BIN.EXE 等应用程序。

三、实验内容

1. 创建 MASM 目录、复制开发软件，或者安装开发软件，熟悉进入和退出 32 位控制台的方法。
2. 熟悉汇编语言的语句格式和程序框架，录入教材例 1-1 程序（EG0101.ASM）。
3. 熟悉汇编语言的开发过程，汇编连接教材例 1-1 程序、生成可执行文件并执行，查阅列表文件。
4. 参考教材附录 A，通过调试例 1-1 程序，熟悉调试程序 WinDbg 的使用。
5. 理解断点调试、单步调试的作用和方法，按照附录 A 调试例 3-6 程序，调试习题 3-22 程序、观察程序运行结果。

四、实验结果与心得

实验 2 数据寻址与传送指令

实验序号：2

实验名称：使用各种寻址方式访问存储器

适用专业：软件工程/信息安全

学时数： 2 学时

一、实验目的和要求

1. 掌握汇编语言进行常量表达和变量定义的方法。
2. 掌握数据寻址方式。
3. 掌握 MOV、XCHG、PUSH 和 POP 等常用传送指令。
4. 掌握配套教材的输入输出子程序库的应用。

二、实验仪器和设备

1. 硬件环境：微机 CPU 486 以上，500MB 以上硬盘，32M 以上内存；
2. 软件环境：装有 MASM 6.15、Debug、LINK、EDIT、CREF.EXE 和 EXR2BIN.EXE 等应用程序。

三、实验内容

1. 参考例 1-1 程序、按照习题 2.14 要求在数据段定义 ASCII 可显示字符的数值，然后进行汇编、连接生成可执行文件，并显示出正确的 ASCII 表。
2. 机验证教材例 2-11 和例 2-12 程序。
3. 验证例 3-3 程序，并体会其中存储器操作数寻址方法的运用。
4. 实现习题 3.9 程序

四、实验结果与心得

实验3 数据运算指令

实验序号：3

实验名称：数据运算指令

适用专业：软件工程/信息安全

学时数：2 学时

一、实验目的和要求

1. 熟悉状态标志的意义。
2. 掌握常用加法、减法等数据运算指令。
3. 掌握逻辑运算、移位等常用位操作类指令。
4. 了解不同编码间的转换作用。

二、实验仪器和设备

1. 硬件环境：微机 CPU 486 以上，500MB 以上硬盘，32M 以上内存；
2. 软件环境：装有 MASM 6.15、Debug、LINK、EDIT、CREF.EXE 和 EXR2BIN.EXE 等应用程序。

三、实验内容与过程

1. 判断习题 3.12 每条指令执行后 AL 的数值和相关标志状态，写出结果。然后将其编辑成为一个完整的汇编语言源程序，汇编连接、生成可执行文件。可以利用本书配套的输入输出子程序库，在每条指令之后，调用其中的 DISPRF（先调用）和 DISPHD，显示 6 个状态标志位和 EAX 内容，并核对事先判断的结果。

2. 验证例 3-5 程序，实现习题 3.15 程序。
3. 验证例 3-8 程序，实现习题 3.22 程序。
4. 实现习题 3.19 程序。

四、实验结果与心得

实验 4 顺序程序和转移指令

实验序号 4

实验名称：顺序程序和转移指令

适用专业：软件工程/信息安全

学时数： 2 学时

一、实验目的和要求

1. 理解顺序程序结构。
2. 掌握无条件转移指令。
3. 掌握带符号数的比较转移指令：JL, JLE, JG, JGE。

二、实验仪器和设备

1. 硬件环境：微机 CPU 486 以上，500MB 以上硬盘，32M 以上内存；
2. 软件环境：装有 MASM 6.15、Debug、LINK、EDIT、CREF.EXE 和 EXR2BIN.EXE 等应用程序。

三、实验内容与过程

1. 验证例 4-2 程序，实现习题 4.7 程序。
2. 实现习题 4.8 程序。
3. 验证例 4-6 程序，实现习题 4.12 程序。
4. 按照附录 A 调试例 4-5 程序，调试习题 4.12 程序，并注意状态标志、并跟踪指令执行顺序（即，程序指针 EIP 数值变化）。

四、实验结果与心得

实验 5 分支与循环程序设计

实验序号：5

实验名称：分支与循环程序设计

适用专业：软件工程/信息安全

学时数： 2 学时

一、实验目的和要求

1. 掌握单分支结构程序的特点和方法。
2. 掌握双分支结构程序的特点和方法。
3. 掌握循环指令和循环程序结构。

二、实验仪器和设备

1. 硬件环境：微机 CPU 486 以上，500MB 以上硬盘，32M 以上内存；
2. 软件环境：装有 MASM 6.15、Debug、LINK、EDIT、CREF.EXE 和 EXR2BIN.EXE 等应用程序。

三、实验内容与过程

1. 验证例 4-10 程序，实现习题 4.17 程序。
2. 验证例 4-11 程序，实现习题 4.16 程序。
3. 验证例 4-18 程序，实现习题 4.27 程序。
4. 综合分支和循环程序结构，实现习题 4.29 程序。
5. 实现习题 4.30 程序。
6. 利用 WinDbg 调试上述习题。

四、实验结果与心得

实验 6 子程序设计

实验序号：6

实验名称：中断程序

适用专业：软件工程/信息安全

学时数： 2 学时

一、实验目的和要求

1. 掌握子程序指令和子程序结构。
2. 熟悉子程序的参数传递的方法。
3. 熟悉子程序模块、子程序库和子程序库包含生成可执行文件的多模块程序结构开发方法。

二、实验仪器和设备

1. 硬件环境：微机 CPU 486 以上，500MB 以上硬盘，32M 以上内存；
2. 软件环境：装有 MASM 6.15、Debug、LINK、EDIT、CREF.EXE 和 EXR2BIN.EXE 等应用程序。

三、实验内容与过程

1. 实现习题 5.8 和 5.13 程序。
2. 实现习题 5.14 各项要求。
3. 利用 WinDbg 调试上述程序。

四、实验结果与心得

实验 7 windows 程序设计

实验序号：7

实验名称：windows 程序设计

适用专业：软件工程/信息安全

学时数： 2 学时

一、实验目的和要求

1. 熟悉 Windows 应用程序接口 API 调用原理。
2. 掌握常用控制台函数的调用方法。
3. 熟悉消息窗口函数的调用方法。
4. 了解 MASM32 开发环境。

二、实验仪器和设备

1. 硬件环境：微机 CPU 486 以上，500MB 以上硬盘，32M 以上内存；
2. 软件环境：装有 MASM 6.15、Debug、LINK、EDIT、CREF.EXE 和 EXR2BIN.EXE 等应用程序。

三、实验内容与过程

1. 实现习题 6-4 程序。
2. 实现习题 6.6 程序。
3. 将主存内容显示子程序加入 32 位输入输出子程序库 IO32.LIB，相应修改包含文件 IO32.INC，并利用一个调用程序验证该子程序的正确。
4. 实现习题 6.8 程序
5. 验证例 6-11 程序，实现习题 6.16 程序。
6. 利用 WinDbg 调试上述程序。

四、实验结果与心得

实验 8 与 C++ 混合编程

实验序号：8

实验名称：与 C++ 混合编程

适用专业：软件工程/信息安全

学时数：2 学时

一、实验目的和要求

1. 熟悉嵌入汇编和模块连接的混合编程和开发过程。

二、实验仪器和设备

1. 硬件环境：微机 CPU 486 以上，500MB 以上硬盘，32M 以上内存；
2. 软件环境：装有 MASM 6.15、Debug、LINK、EDIT、CREF.EXE 和 EXR2BIN.EXE 等应用程序。

三、实验内容

1. 验证例 7-1 和例 7-2 程序。
2. 实现习题 7.9 各项要求，并按照规定编写汇编语言程序。

四、实验结果与心得

可选实验 DOS 应用程序与浮点指令

实验序号: XXX

实验名称: DOS 应用程序与浮点指令

适用专业: 软件工程/信息安全

学时数: 2 学时

一、实验目的和要求

1. 掌握 16 位 DOS 应用程序的编写。
2. 熟悉串操作指令。
3. 掌握输入输出指令应用。
4. 熟悉浮点数据格式和常见浮点指令及其编程。

二、实验仪器和设备

1. 硬件环境: 微机 CPU 486 以上, 500MB 以上硬盘, 32M 以上内存;
2. 软件环境: 装有 MASM 6.15、Debug、LINK、EDIT、CREF.EXE 和 EXR2BIN.EXE 等应用程序。

三、实验内容

1. 验证例 8-2 程序、实现习题 8.6 程序。
2. 验证例 8-4 程序、实现习题 8.11 程序。
3. 验证例 8-7 程序、实现习题 8.14 程序。
4. 验证例 9-4 程序和例 9-5 程序, 实现习题 9.6 程序

四、实验结果与心得

附录

汇编语言常用出错信息

汇编程序在对源程序的汇编过程中,若检查出某语句有语法错误,随时在屏幕上给出出错信息。如操作人员指定的列表文件名(即.LST),汇编程序亦将在列表文件中出错的下面给出出错信息,以便操作人员即时查找错误,给予更正。MASM5.0 出错信息格式如下:

源程序文件行: **WARNING/ERROR** 错误信息码: 错误描述信息

其中,错误描述信息码由五个字符组成。第一个是字母 A,表示汇编语言程序出错;接着有一个数字指明出错类别:‘2’为严重错误,‘4’为严肃警告,‘5’为建议性警告,最后三位为错误编号。

错误编号	错误描述
0	Block nesting error 嵌套出错。嵌套的过程、段、结构、宏指令或重复块等非正常结束。例如在嵌套语句中有外层的结束语句,而无内层的结束语句
1	Extra characters on line 一语句行有多余字符,可能是语句中给出的参数太多
2	Internal error-Register already defined 这是一个内部错误。如出现该错误,请记下发生错误的条件,并使用 Product Assistance Request 表与 Microsoft 公司联系
3	Unkown type specifer 未知的类型说明符。例如类型字符拼错,把 BYTE 写成 BIT, NEAR 写成 NAER 等
4	Redefinition of symbol 符号重定义。同一标识符在两个位置上定义。在汇编第一遍扫描时,在这个标识符的第二个定义位置上给出这个错误
5	Symbol is multidefined 符号多重定义。同一标识符在两个位置上定义。在汇编第二遍扫描时,每当遇到这个标识符都给出这个错误
6	Phase error between passes 两次扫描间的遍错。一个标号在二次扫描时得到不同的地址值,就会给出这种错误。若在启动 MASM 时使用/D 任选项,产生第一遍扫描的列表文件,它可帮助你查找这种错误
7	Already had ELSE clause 已有 ELSE 语句。在一个条件块里使用多于一个的 ELSE 语句
8	Must be in conditional block 没有在条件块里。通常是有 ENDIF 或 ELSE 语句,而无 IF 语句
9	Symbol not defined 符号未定义,在程序中引用了未定义的标识符
10	Syntax error 语法错误。不是汇编程序所能识别的一个语句
11	Type illegal in context 指定非法类型。例如对一个过程指定 BYTE 类型,而不是 NEAR 或 FAR
12	Group name must be unique 组名应是唯一的。作为组名的符号作为其他符号使用
13	Must be declared during pass 1 必须在第一遍扫描期间定义。在第一遍扫描期间,如一个符号在未定义前就引用,就会出现这种错误。
14	Illegal public declaration 一个标识符被非法的指定为 PUBLIC 类型
15	Symbol already defferent kind 重新定义一个符号为不同种类符号。例如一个段名重新被当作变量名定义使用
16	Reserved word used as symbol 把汇编语言规定的保留字作标识符使用
17	Forward reference illegal

错误编号	错误描述
	非法的向前引用。在第一遍扫描期间，引用一个未定义符号。
18	Operand must be register 操作数位置上应是寄存器，但出现了标识符
19	Wrong type of register 使用寄存器出错
20	Operand must be segment or group 应该给出一个段名或组名。例如 ASSUME 语句中应为某段寄存器和指定一个段名或组名，而不应是别的标号或变量名等
21	Symbol has no segment 不知道标识符的段属性
22	Operand must be type specifier 操作数应给出类型说明，如 NEAR、FAR、BYTE 等
23	Symbol already defined locally 已被指定为内部的标识符，企图在 EXTRN 语句中又定义外部标识
24	Segment parameters are changed 段参数被改变。如同一标识符定义在不同段内
25	Improper align/combin type 段定义时的定位类型/组合类型使用出错
26	Reference to multidefined symbol 指令引用了多重定义的标识符
27	Operand expected 需要一个操作数，只有操作符
28	Operator expected 需要一个操作符，但只有操作数
29	Divsion by 0 or overflow 除以 0 或溢出
30	Negative shift count\ 运算符 SHL 或 SHR 的移位表达式值为负数
31	Operand type must match 操作数类型不匹配。双操作数指令的两个操作数长度不一致，一个是字节，一个是字
32	Illegal use of external 外部符号使用出错
33	Must be record field name 应为记录字段名。在记录字段名位置上出现另外的符号
34	Must be record name or field name 应为记录名或记录字段名。在记录名或记录字段名位置上出现另外的符号
35	Operand must be size 应指明操作数的长度（如 BYTE、WORD 等）。通常使用 PTR 运算即可改正
36	Must be variable,label,or constant 应该是变量名、标号、或常数的位置上出现了其他信息
37	Must be stucture field name 应该为结构字段名。在结构字段名位置上出现了另外的符号
38	Lefe operand must segment 操作数的左边应该是段的信息。如设 DA1、DA2 均是变量名，下列语句就是错误的：“MOV AX, DA1: DA2”。DA1 位置上应使用某段寄存器名
39	One operand must constant 操作数必须是常数。
40	Operand must be in same segment or one constant “—”运算符用错。例如“MOV AL, —VAR”，其中 VAR 是变量名，应有一常数参加运算。又如两个不同段的变量名相减出错
41	Normal type operand expected 要求给出一个正常的操作数。
42	Constant expected 要求给出一个常数。
43	Operand must have segment 运算符 SEG 用错。
44	Must be associated with data

错误编号	错误描述
	在必须与数据段有关的位置上出现了代码段有关的项
45	Must be associated with code 在必须与代码段有关的位置上出现了数据段有关的项
46	Multiple base registers 同时使用了多个基址寄存器。如“MOV AX, [SI][BP]”
47	Multiple index registers 同时使用了多个变址寄存器。如“MOV AX, [SI][DI]”
48	Must be index or base register 指令仅要求使用基址寄存器或变址寄存器，而不能使用其他寄存器。
49	Illegal use of register 非法使用寄存器出错
50	Value is out of range 数值太大，超过允许值。例如：“MOV AL, 100H”
51	Operand not in current CS ASSUME segment 操作数不在当前代码段内。通常指转移指令的目标地址不在当前 CS 段内
52	Improper operand type 操作数类型使用不当。例如：“MOV VAR1, VAR2”。两个操作数均为存储器操作数，不能汇编出目标代码
53	Jump out of range by %ld byte 条件转移指令跳转范围超过-128~+127 个字节。出错厂、信息同时给出超过的字节数
54	Index displacement must be constant 变址寻址的位移量必须是常数
55	Illegal register value 非法的寄存器值。目标代码中表达寄存器的值超过 7
56	Immediate mode illegal 不允许使用立即数寻址。例如“MOV DS, CODE”其中 CODE 是段名，不能把段名作为立即数传送给段寄存器 DS
57	Illegal size for operand 使用操作数大小（字节数）出错。例如：使用双字的存储器操作数
58	Byte register illegal 要求用字寄存器的指令使用了字节寄存器。如 PUSH, POP 指令的操作数寄存器必须是字寄存器
59	Illegal uer of CS register 指令中错误使用了段寄存器 CS。如：“MOV CS, AX” CS 不能做目的操作数
60	Must be accumulator register 要求用 AX 或 AL 的位置上使用可其他寄存器。如 IN, OUT 指令必须使用累加器 AX 或 AL
61	Improper uer of segment register 不允许使用段寄存器的位置上使用了段寄存器。如“SHL DS, 1”
62	Missing or unreachable CS 试图跳转去执行一个 CS 达不到的标号。通常是指缺少 ASSUME 语句中 CS 与代码段相关联
63	Operand combination illegal 双操作数指令中两个操作数组合出错
64	Near JMP/CALL to different CS 试图用 NEAR 属性的转移指令跳转到不在当前段的一个地址
65	Label cannot have segment override 段前缀使用出错
66	Must have instuction agter prefix 在重复前缀 REP, REPE, REPNE 后面必须有指令
67	Cannot override ES for destination 串操作指令中目的操作数不能用其他段寄存器替代 ES
68	Cannot address with srgment register 指令中寻找一个操作数，但 ASSUME 语句中未指明哪个段寄存器与该操作数所在段有关联
69	Must be in segment block 指令语句没有在段内
70	Cannot use EVEN or ALIGN with byte alignment 在段定义伪指令的定位类型中选用 BYTE，这时不能使用 EVEN 或 ALIGN 伪指令
71	Forward needs override or FAR

错误编号	错误描述
	转移指令的目标没有在源程序中说明为 FAR 属性，可用 PTR 指定
72	Illegal value for DUP count 操作符 DUP 前的重复次数是非法的或未定义
73	Symbol id already external 在模块内试图定义的符号，它已在外部符号伪指令中说明
74	DUP nesting too deep 操作数 DUP 的嵌套太深
75	Illegak use of undefinde operand(?) 不定操作符 “?” 使用不当。例如 “DB 10H DUP (? +2)”
76	Too many valer for struc or record initialization 在定义结构变量或记录变量时，初始值太多
77	Angle brackets requored around initialized list 定义结构体变量时，初始值未用尖括号 (<>) 括起来
78	Directive illegal structure 在结构体定义中的伪指令使用不当。结构定义中的伪指令语句仅二种：分号 (;) 开始的注释语句和用 DB、DW 等数据定义伪指令语句
79	Override with DUP illegal 在结构变量初始值表中使用 DUP 操作符出错
80	Field cannot be overridden 在定义结构变量语句中试图对一个不允许修改的字段设置初值
81	Override id of wrong type 在定义结构变量语句中设置初值时类型出错
82	Circular chain of EQU aliases 用等值语句定义的符号名，最后又返回指向它自己。如： A EQU B B EQU A
83	Cannot emulate cooprocessor opcode 仿真器不能支持的 8087 协处理器操作码
84	End of file,not END directive 源程序文件无 END 文件
85	Data emitted with no segment 语句数据没有在段内