

高等院校信息技术规划教材

微型计算机原理与 接口技术实验指导 (第2版)

陈燕俐 许建 李爱群 周宁宁 编著



清华大学出版社

高等院校信息技术规划教材

微型计算机原理与 接口技术实验指导（第2版）

陈燕俐 许建 李爱群 周宁宁 编著

清华大学出版社

北京

内 容 简 介

本书是《微型计算机原理与接口技术》(孙力娟等编著,清华大学出版社出版)一书的配套实验教材。本教材结合课程内容,针对汇编语言程序设计及接口技术编排了内容丰富的软硬件实验项目和指导性的实验例题,主要内容有汇编语言程序设计实验、微型计算机教学实验系统及系统软件介绍、微型计算机接口实验。本书的硬件实验以清华大学科教仪器厂研发的“TPC-486EM 32 位微机原理与接口技术实验系统”为实验平台。

本书内容丰富,大量的实验实例和实验项目扩展了教科书的内容,既可作为高等院校“汇编语言程序设计”、“微机原理与接口技术”等课程的实验教材,也可供自学者及从事计算机应用的工程技术人员参考。

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

微型计算机原理与接口技术实验指导/陈燕俐等编著. --2 版. --北京:清华大学出版社,2016

高等院校信息技术规划教材

ISBN 978-7-302-42201-3

I. ①微… II. ①陈… III. ①微型计算机—理论—高等学校—教学参考资料 ②微型计算机—接口技术—高等学校—教学参考资料 IV. ①TP36

中国版本图书馆 CIP 数据核字(2015)第 279030 号

责任编辑:白立军

封面设计:常雪影

责任校对:焦丽丽

责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 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

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:三河市春园印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:10.75 字 数:252 千字

版 次:2010 年 7 月第 1 版 2016 年 1 月第 2 版 印 次:2016 年 1 月第 1 次印刷

印 数:1~4000

定 价:25.00 元

产品编号:066740-01

前言

“微型计算机原理与接口技术”是高等院校计算机专业及电类相关专业计算机硬件课程体系中的一门重要的专业基础课,是一门理论与实际结合十分紧密、实践性很强的课程。实验是微机接口教学过程中十分重要的环节,是全面提高学生素质的有效途径。

本书是与教材《微型计算机原理与接口技术》(书号: 9787302408956)配套的实验教程,目的是使学生通过实验加深对理论课程的理解,培养学生的编程能力和实际动手能力。本书分为软件实验和硬件实验两部分,硬件实验以清华大学科教仪器厂研发的“TPC-486EM 32 位微机原理与接口技术实验系统”为实验平台。本书共分为 6 章,内容如下。

第 1 章为汇编语言语法实验,介绍汇编语言源程序的格式和框架、汇编语言程序的开发过程,以及汇编语言语法练习实验,为学生下一步的软硬件实验打下基础。第 2 章和第 3 章是汇编语言程序设计实验实例和内容,实验内容丰富,涵盖了一般汇编语言程序设计和微机原理教学中所要求做的所有软件实验。第 4 章介绍了 Win32 汇编程序的框架、Win32 汇编语言程序开发过程,以及 Win32 窗口程序、字符串显示程序、消息处理程序实验。第 5 章对“TPC-486EM 32 位微机原理与接口技术实验系统”的结构和功能、上位机系统软件的使用进行介绍。第 6 章为硬件接口实验,覆盖了目前高等院校微机接口实验教学大纲的主要内容,包括保护模式程序设计。综合性实验要求学生能熟练掌握各种常用接口芯片的结构和功能,能综合运用接口芯片达到实验要求。

本书按照实验说明、实验目的和要求、实验示例和实验项目进行每一个实验的组织,每种实验的实验示例都给出了源程序清单和注释,这些程序都经过调试和运行;涉及硬件的还给出了具体的实验原理和硬件连线。本书还提供了大量的实验题目,教师可以根据本校的教学特点和要求选择相应的实验内容。

本书由陈燕俐、许建、李爱群和周宁宁编著,其中引言、第 3 章和第 4 章由陈燕俐编写,陈燕俐、许建、李爱群合写了第 1 章,陈燕



俐、周宁宁合写了第2章,许建编写了第5章,许建、陈燕俐、李爱群、周宁宁合写了第6章,由陈燕俐负责全书的统稿工作。本书的编写还得到南京邮电大学孙力娟教授和洪龙教授的热情鼓励、悉心指导和积极帮助。他们审阅了全书,提出了许多宝贵建议,使得本书更加完善,在此表示衷心的感谢。本书在编写过程中,还得到清华大学科教仪器厂和清华大学出版社的大力支持,也参考了相关的书籍(已在参考文献中列出),在此一并致以诚挚的谢意。

由于编者的水平有限,书中难免有错漏之处,恳请读者提出批评意见。

编 者

2015年7月

目录

Contents

第 1 章 汇编语言语法实验	1
1.1 汇编语言程序开发过程	1
1.2 汇编语言程序编程练习	4
1.3 汇编语言语法实验	8
第 2 章 结构化程序设计实验	10
2.1 顺序程序设计	10
2.2 分支程序设计	12
2.3 循环程序设计	16
2.4 子程序设计	22
2.5 宏指令设计	27
第 3 章 应用程序设计实验	30
3.1 数制及代码转换程序设计	30
3.2 数值计算程序设计	35
3.3 字符串操作程序设计	39
3.4 图形显示程序设计	45
3.5 磁盘文件管理程序设计	54
第 4 章 Win32 汇编程序设计实验	61
4.1 Win32 汇编语言程序开发过程	61
4.2 Win32 汇编语言程序编程练习	64
4.3 Win32 窗口程序设计	67
4.4 字符串显示程序设计	70
4.5 消息处理程序设计	74



第5章 TPC-486EM 32位微机原理与接口技术实验系统	79
5.1 TPC-486EM 微机教学实验系统概述	79
5.1.1 功能特点	79
5.1.2 教学实验系统结构	81
5.2 上位机系统软件的使用说明	82
5.2.1 概述	82
5.2.2 软件的详细使用说明	84
5.2.2 软件使用举例	89
第6章 硬件接口实验	90
6.1 定时器/计数器实验	90
6.2 中断实验	95
6.3 串行通信实验	102
6.4 并行接口实验	111
6.5 微机接口人机界面交互程序实验	117
6.6 D/A 转换实验	127
6.7 A/D 转换实验	131
6.8 存储器扩充实验	135
6.9 DMA 实验	142
6.10 保护模式实验	151
6.11 综合性实验	160
参考文献	166

汇编语言语法实验

1.1 汇编语言程序开发过程

汇编语言(Assembly Language)是唯一能够利用计算机所有硬件特性,并能直接控制硬件的编程语言。汇编语言编写的程序称为汇编语言程序,汇编语言程序必须翻译成机器语言程序(即目标代码程序),才能在机器上运行。

1. 汇编语言源程序的格式

一个完整的汇编语言源程序在结构上必须做到4点。

- (1) 用方式选择伪指令说明执行该程序的 CPU 类型。
- (2) 用段定义语句定义每一个逻辑段。
- (3) 用 ASSUME 语句说明段约定。
- (4) 用汇编结束语句说明源程序结束。

实模式下汇编程序有两种编程格式:一种格式生成扩展名为 EXE 的可执行文件,称为 EXE 文件的编程格式;另一种格式可以生成扩展名为 COM 的可执行文件,称为 COM 文件的编程格式。

典型的 EXE 编程格式如下:

.486	;方式选择伪指令说明执行该程序的 CPU 类型
DATA SEGMENT USE16	;定义作为数据段的逻辑段,段名 DATA
	;定义变量
:	
DATA ENDS	;数据段结束
CODE SEGMENT USE16	;定义作为代码段的逻辑段,段名 CODE
ASSUME CS:CODE,DS:DATA	;段约定
BEG: MOV AX,DATA	;数据段的段基址赋给段寄存器
MOV DS,AX	
...	;程序代码
MOV AH,4CH	;程序结束,返回 DOS

```

INT 21H
CODE ENDS ;代码段结束
END BEG ;源程序结束,程序的开始点为 BEG 指令

```

典型的 COM 编程格式如下:

```

.486 ;方式选择伪指令说明执行该程序的 CPU 类型
CODE SEGMENT USE16 ;定义作为代码段的逻辑段,段名 CODE
    ASSUME CS:CODE ;段约定
    ORG 100H ;偏移地址为 100H 的单元必须是程序的启动指令
BEG: JMP START ;跳过数据区
    ... ;定义程序使用的数据,也可设置在代码段的末尾
START: ;程序代码
    :
    MOV AH,4CH ;程序结束,返回 DOS
    INT 21H
CODE ENDS ;代码段结束
END BEG ;源程序结束,程序的开始点为 BEG 指令

```

2. 汇编语言的开发过程

汇编语言程序设计的实验环境对计算机的配置要求比较低,普通的个人计算机一般都可以满足。常用的汇编语言开发工具有 Borland 公司的 TASM 和 Microsoft 公司的 MASM。其中,MAASM 是 Microsoft Macro Assembler 的缩写,是 Microsoft 公司为 x86 微处理器家族开发的汇编开发环境,该程序自面世以来已经推出了多个不同版本,如 MASM 4.0、MASM 6.11、MASM 12 等。考虑到程序的兼容性,初学者也可以使用“MASM for Windows 集成实验环境”在 Windows 环境下进行汇编语言的学习,该程序是由安阳工学院开发的,集编辑、汇编链接、调试为一体的 MASM 集成环境。

汇编语言程序的开发过程如图 1.1 所示。这个过程主要由编辑、汇编、链接和调试几个步骤构成。

1) 源程序的编辑

编辑就是调用编辑程序编辑源程序,生成一个扩展名为 ASM 的文本源文件。DOS 提供的 EDIT.EXE 或其他屏幕编辑软件都能完成编辑任务。

2) 源程序的汇编

为了使汇编语言编写的程序能在机器上运行,必须利用汇编程序(Assembly Program,如 Microsoft 公司的 MASM 或 Borland 公司的 TASM)对源程序进行翻译,生成扩展名为 OBJ 的目标文件。

汇编语言源程序包含指令性语句(即符号指令)和指示性语句(即伪指令)两类语句。符号指令和机器指令具有一一对应的关系,伪指令是为汇编程序提供汇编信息,为链接程序提供链接信息,在汇编后并不产生目标代码。

在汇编过程中,如果汇编程序检查到源程序中有语法错误,则不生成目标代码文件,并给出错误信息。根据用户需要,汇编程序还可生成列表文件(LST 文件)和交叉参考文

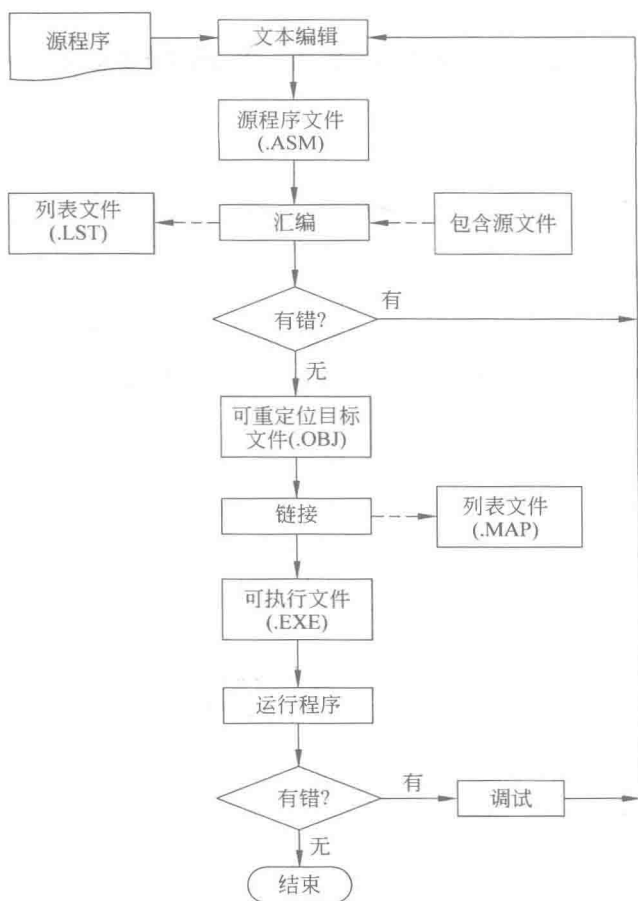


图 1.1 汇编语言程序的开发过程

件(XRF 文件)。

3) 目标程序的链接

链接就是利用链接程序(如 Microsoft 公司的 LINK 或 Borland 公司的 TLINK)将用户目标程序和库文件进行链接、定位,生成扩展名为 EXE 的可执行文件。链接时,如果在目标文件或库中找不到所需的链接信息,则链接程序会发出错误信息,而不生成可执行文件。根据用户需要,链接程序还可生成内存映射文件(MAP 文件)。

4) 动态调试

有时用户生成的 EXE 文件运行后,并没有按照设计的意图运行,这就需要对程序进行调试。根据具体情况,调试的过程也不尽相同。一般地,可利用调试工具(各版本 DOS 所带的 DEBUG 或 Borland 公司的 Turbo Debugger)对生成的可执行文件进行调试,找出错误。再对源程序进行修改……即重复地进行编辑、汇编、链接、调试,直到生成完全正确的可执行文件。



1.2 汇编语言程序编程练习

1. 实验说明

在 1.1 节的基础上掌握汇编语言程序设计过程。

2. 实验目的和要求

掌握汇编语言源程序的编辑、汇编、目标文件的链接和可执行文件的调试执行全过程;掌握文本编辑软件、MASM、LINK 和 DEBUG 的使用方法以及汇编语言的语法规则。

3. 实验示例

【例 1.1】 显示 5 行 HELLO。

```
;FILENAME: EXA121.ASM
.486
DATA        SEGMENT  USE16
MSG         DB        'HELLO'
            DB         0,0,0                ;①

DATA        ENDS
CODE        SEGMENT  USE16
            ASSUME    CS:CODE,DS:DATA
BEG:        MOV      AX,DATA
            MOV      DS,AX
            MOV      ES,AX                ;②
            MOV      CX,5
LL1:        MOV      MSG+5,0DH            ;③
            MOV      MSG+6,0AH            ;④
            MOV      MSG+7,'$'            ;⑤
            CALL     DISP
            MOV      MSG+5,0                ;⑥
            MOV      MSG+6,0                ;⑦
            MOV      MSG+7,0                ;⑧
            LOOP     LL1
            MOV      AH,4CH
            INT      21H
DISP        PROC
            MOV      AH,9
            MOV      DX,OFFSET MSG
            INT      21H
            RET
```

```
DISP      ENDP
CODE      ENDS
          END      BEG
```

以上是该程序的源文件,执行后,在屏幕上显示 5 行 HELLO,语句①~⑧是为了演示 DEBUG 而设置的。下面以此例来介绍汇编语言源程序的开发过程。

1) 启动 DOS 命令窗口

如果计算机安装的是 Windows 操作系统,则用户有以下两种方法启动 DOS 命令窗口。

方法 1: 在 Windows“开始”菜单中执行“运行”命令,在“运行”对话框中输入 cmd,单击“确定”按钮启动 DOS 命令窗口,如图 1.2 所示。

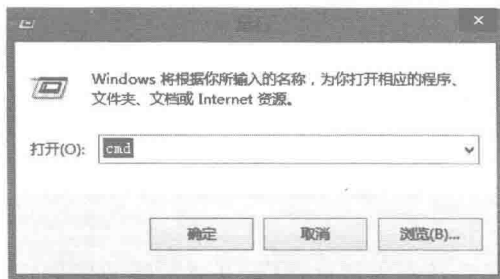


图 1.2 Windows 系统的“运行”对话框

方法 2: 在 Windows“开始”菜单中执行“程序”→“附件”→“命令提示符”命令,也能够启动 DOS 命令窗口。

用户进入 DOS 命令窗口后,应输入“进入子目录”命令进入当前汇编目录(即开发工具的相关文件已复制在此目录下),例如:

```
>c:↵          (↵表示回车键)
>cd masm↵
```

2) 编辑

采用文本编辑软件编辑汇编语言源程序,注意保存时,文件的扩展名必须是 asm。如果采用的是 Windows 环境下的如“记事本”等的编辑工具,保存时的“保存类型”选项必须选择“所有文件”,如图 1.3 所示。

源程序以及汇编、链接后的目标程序和可执行程序,可以存放在开发工具所在的目录,如 C:\masm,也可以集中存放在用户建立的一个文件夹中,例如 D:\myfile 中。但这时,所有涉及这些文件的路径前缀不能省略。

如果 exal21.asm 保存在 C:\masm 目录,则命令格式(命令中不区分大小写)为

```
C:\masm>edit exal21.asm↵
```

如果欲将 exal21.asm 保存在 D:\myfile 中,则命令格式为

```
C:\masm> edit D:\myfile\exal21.asm↵
```

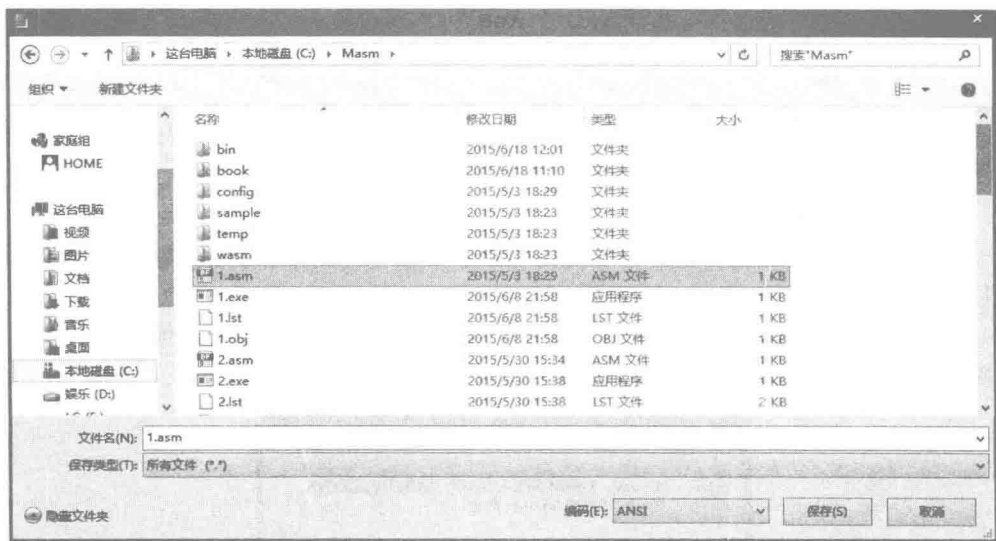


图 1.3 记事本的“另存为”对话框

3) 汇编

汇编操作能够将源程序转换为目标程序,并显示错误信息。

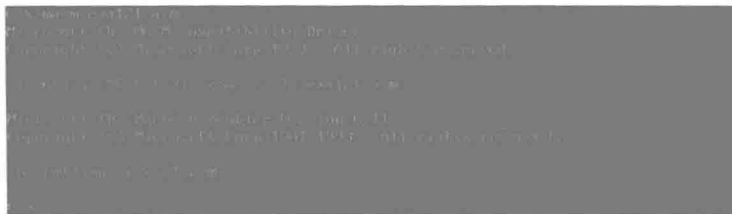
如果 exal21.asm 保存在 C:\masm 目录,则命令格式为

```
C:\masm>masm exal21
```

如果 exal21.asm 保存在 D:\myfile 目录,并且欲将 exal21.obj 也保存在此目录,则命令格式为

```
C:\masm> ! masm D:\myfile\exal21.asm D:\myfile\exal21.obj
```

如果系统给出源程序中的错误信息(错误原因和错误行号),则需要采用编辑软件修改源程序中的错误,直到汇编正确为止。编译结果如图 1.4 所示。



```
C:\masm>link exa121
```

执行该操作后,显示结果如图 1.5 所示。

```

[SN:106 1990:1 of.]

This is off 0th segment. Unavailable linker. Version 5.31.909 and 13.196.
Copyright (C) Microsoft Corp 1984-1990. All rights reserved.

Bad file (E2043.csc1)
Bad file (bad map)
libraries (1.10b):
Bad file (Bad file: bad def 1)
LINK : warning L4061: no stack segment.
END

```

图 1.5 链接结果

其中的链接过程还会生出一个扩展名为 MAP 的文件,它可以给出每个段的地址分配情况和长度;在链接的过程中还与一个库文件(扩展名为 LIB)相关。一般的汇编程序不需要库文件,只有与某些高级语言(如 C 语言)接口时才需要库文件,此时只需要输入库名即可。在该例中可以选择直接按 Enter 键,跳过这些文件即可。

5) 运行 EXE 可执行程序。

EXE 文件是可执行文件,在 Windows 环境下直接双击 EXE 文件图标就可执行,也可在 DOS 命令行提示符下直接输入可执行文件名后按 Enter 键执行。例如:

C:\masm>exa121

运行结果如图 1.6 所示。

```
HELLO  
HELLO          c  
HELLO  
HELLO  
HELLO  
HELLO
```

Press any key to continue.

图 1.6 运行结果

6) 调试程序

如果程序运行错误,则可启动调试软件 Debug 对程序进行调试,找出错误原因。

C:\masm>Debug exa121 (exa121.asm 保存在 C:\masm 目录)

或

C:\masm>Debug D:\myfile\exa121 (exa121.asm 保存在 D:\myfile 目录)

常用参数如下:

-u 反汇编

-r 显示寄存器内容

-g 执行到断点处 后面+行号 例如-G9

-d 查看数据



-q 退出返回操作系统

Debug 过程中参数为-u 的结果如图 1.7 所示。

-u			
076B:0006 1304807	MOV	DI, 076B	
076B:0007 11200	MOV	DI, 02	
076B:0008 11200	MOV	DI, 05	
076B:0009 1005200	MOV	DI, 0005	
076B:000A 06069000	MOV	EDI, ESI [0005:0000]	
076B:000B 06060000	MOV	EDI, ESI [0005:0000]	
076B:000C 06060700	MOV	EDI, ESI [0005:0007]	
076B:000D 1012000	CALL	CALL	
076B:000E 06060700	MOV	EDI, ESI [0005:0007]	

图 1.7 Debug 过程中参数为-u 的结果

4. 实验项目

【实验 1.1】 汇编语言编程过程的练习。

请将例 1.1 的源程序通过一个编辑软件输入计算机并加以保存,并命名为 EXA121.ASM。然后调用 MASM 和 LINK 完成编译和链接,生成可执行文件 EXA121.EXE。试着在当前目录下运行程序 EXA121.EXE。最后用 Debug 将 EXA121.EXE 调入 Debug 的调试界面,掌握调试过程。

1.3 汇编语言语法实验

1. 实验说明

在 1.2 节的基础上进一步掌握汇编语言程序开发过程。

2. 实验目的和要求

进一步学习汇编语言源程序的编辑、汇编、目标文件的链接和可执行文件的执行全过程;掌握编辑软件、MASM、LINK 和 Debug 的使用方法;掌握汇编语言的语法规则。

3. 实验项目

【实验 1.2】 排除语法错误。

下面给出的是一个通过比较法完成 8 位二进制数转换成十进制数送屏幕显示功能的汇编语言源程序,但有很多语法错误。要求实验者按照原样对源程序进行编辑,汇编后,根据 MASM 给出的错误信息对源程序进行修改,直到没有语法错误为止。然后进行链接,并执行相应的可执行文件。正确的执行结果是在屏幕上显示 $25+9=34$ 。

【程序清单】

```
;FILENAME: EXA131.ASM
.486
DATA SEGMENT NT USE16
SUM DB ?,?,
```

```
MESG      DB      '25+9='
           DB      0,0
N1         DB      9,F0H
N2         DW      25
DATA       ENDS
CODE       SEGMENT      USE16
           ASSUME CS: CODE, DS: DATA
BEG:       MOV     AX,      DATA
           MOV     DS,      AX
           MOV     BX,      OFFSET SUM
           MOV     AH,      N1
           MOV     AL,      N2
           ADD     AH,      AL
           MOV     [BX],    AH
           CALL    CHANG
           MOV     AH,      9
           MOV     DX,      OFFSEG MEST
           INT     21H
           MOV     AH,      4CH
           INT     21H
CHANG:     PROC
LAST:      CMP     [BX],10
           JC      NEXT
           SUB     [BX],10
           INC     [BX+7]
           JMP     LAST
NEXT:      ADD     [BX+8],SUM
           ADD     [BX+7],30H
           ADD     [BX+8],30H
           RET
CHANG:     ENDP
CODE       ENDS
END        BEG
```

结构化程序设计实验

结构化程序设计是指具有结构性的编程方法。采用结构化程序设计方法编程,旨在提高所编程序的质量。自顶向下、逐步精化方法有利于在每一抽象级别上尽可能保证所编程序的正确性;按模块组装方法编程以及所编程序只含顺序、分支和循环3种程序则可使程序结构良好、易读、易理解和易维护,并易于保证及验证程序的正确性。任一流程图均可利用循环和嵌套等价地改写成只含顺序、分支和递归的程序,并且每种程序只有一个入口和一个出口。汇编语言程序设计的主要方法包括顺序、分支、循环、子程序和宏指令的设计等。

2.1 顺序程序设计

1. 实验说明

顺序程序是一种最简单也是最基本的结构形式,是最简单的序列结构,程序上没有用到分支和循环,没有控制转移类指令,它的执行流程与指令的排列顺序完全一致,顺序程序设计是所有程序设计的基础。

2. 实验目的和要求

掌握顺序程序的编程方法。

3. 实验示例

【例 2.1】 采用顺序编程方法,实现在屏幕上显示字符串“Enjoy programming in MASM”。

【程序流程图】

程序流程图如图 2.1 所示。

【程序清单】

```
;FILENAME: EXA211.ASM  
.486  
DATA    SEGMENT USE16
```

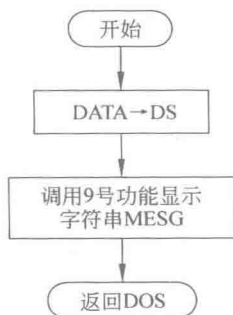


图 2.1 例 2.1 程序流程图