

南京邮电大学

# 实 验 报 告

( 2024/2025 学年 第二学期 )

课程名称	计算机组成基础		
实验名称	汇编语言语法练习及分支循环程序设计		
实验时间	2025	年 4 月 1 日	
指导单位	计算机学院、软件学院、 网络空间安全学院		
指导教师	崔凯燕		

学生姓名	于明宏	班级学号	B23041011
学院(系)	计软网安院	专 业	信息安全

## 实 验 报 告

实验名称	汇编语言语法练习及分支循环 程序设计			指导教师	崔凯燕
实验类型	验证	实验学时	2	实验时间	2025.4.1
<p><b>一、实验目的和要求</b></p> <ol style="list-style-type: none"> <li>1. 掌握汇编语言源程序的格式；</li> <li>2. 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；</li> <li>3. 掌握编辑软件、TASM/MASM、TLINK 和 TD 的使用方法、掌握汇编语言的语法规则。</li> <li>4. 实验内容： <ol style="list-style-type: none"> <li>(1) 使用非命令行编译方式（使用汇编程序开发集成环境），编写循环结构的汇编程序，实现在屏幕打印各自学号姓名 10 行。</li> <li>(2) 【实验 1.3.1】排除语法错误：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 TASM 给出的信息对源程序进行修改，知道没有语法错误为止。然后进行链接，并执行相应可执行文件。正确的执行结果在屏幕上显示：25+9=34。</li> </ol> </li> </ol>					
<p><b>二、实验环境</b></p> <p>Windows XP/Win 7 32 位 + TASM/TNPP/轻松汇编/未来汇编软件</p>					

### 三、实验过程描述与结果分析

#### (1) 循环结构汇编程序开发

程序段为：

```
DATA SEGMENT USE16
    MSG DB 'B23041011 Minghong Yu', 0DH, 0AH, '$'
    COUNT DB 10
DATA ENDS

CODE SEGMENT USE16
    ASSUME CS:CODE, DS:DATA
    BEG:
        MOV     AX, DATA
        MOV     DS, AX
        MOV     CX, 10

    PRINT_LOOP:
        MOV     DX, OFFSET MSG
        MOV     AH, 9
        INT     21H

        LOOP    PRINT_LOOP

        MOV     AH, 4CH
        INT     21H

CODE ENDS
    END BEG
```

运行结果为：

```

Assembling file:  D:\test.asm  to  test.OBJ
Error messages:   None
Warning messages: None
Passes:          1
Remaining memory: 468k

D:\>TLINK D:\test
Turbo Link  Version 7.1.30.1. Copyright (c) 1987, 1996 Borland International
Warning: No stack

D:\>D:\test
B23041011 Minghong Yu
B23041011 Minghong Yu
B23041011 Minghong Yu
B23041011 Minghong Yu
B23041011 Minghong Yu
B23041011 Minghong Yu
B23041011 Minghong Yu
B23041011 Minghong Yu
B23041011 Minghong Yu
B23041011 Minghong Yu
D:\>_

```

## (2) 排除语法错误

要求通过比较法完成 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，要求按照原样对源程序进行编辑，汇编后，根据 TASM 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。

错误代码：

```

.486
DATA SEGMENT USE16
    SUM DB ?,?,
    MSG 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,OFFSET 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

```

### 修改后代码:

```

DATA SEGMENT USE16
      SUM  DB ?,?
      MSG  DB '25+9=','0,0','$'
      N1   DB 9
      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,BYTE PTR N2
ADD    AH,AL
MOV    [BX],AH
CALL   CHANG
MOV    AH,9
MOV    DX,OFFSET MSG
INT     21H
MOV    AH,4CH
INT     21H
```

CHANG PROC

LAST:

```
CMP    BYTE PTR [BX],10
JC      NEXT
SUB    BYTE PTR [BX],10
INC    BYTE PTR [BX+7]
JMP    LAST
```

NEXT:

```
MOV    AL,[BX]
ADD    [BX+8],AL
ADD    BYTE PTR [BX+7],30H
ADD    BYTE PTR [BX+8],30H
RET
```

CHANG ENDP

CODE ENDS

END BEG

运行结果为:

```
Z:\>mount d ./code
Drive D is mounted as local directory ./code/

Z:\>d:

D:\>set PATH=C:\TASM

D:\>TASM D:\test.asm
Turbo Assembler Version 4.1 Copyright (c) 1988, 1996 Borland International

Assembling file:  D:\test.asm to test.OBJ
Error messages:   None
Warning messages: None
Passes:          1
Remaining memory: 468k

D:\>TLINK D:\test
Turbo Link Version 7.1.30.1. Copyright (c) 1987, 1996 Borland International
Warning: No stack

D:\>D:\test
25+9=34
D:\>_
```

四、实验小结（包括问题和解决方法、心得体会、意见与建议等）

在实验过程中，因未正确初始化数据段寄存器（DS）和字符串结尾符号缺失，导致循环程序无法正常显示结果。通过补充 MOV AX, DATA 与 MOV DS, AX 初始化数据段，并在字符串末尾添加\$符号，最终实现屏幕循环输出学号姓名 10 次。此外，在修改语法错误时发现，变量定义格式错误（如 N1 DB 9,F0H）、操作数类型不匹配（如未用 BYTE PTR 转换字类型数据）以及标签拼写错误（MEST 误写为 MSG）是常见问题，逐一修正后程序成功显示“25+9=34”。

通过实验深刻体会到汇编语言对语法细节的严苛要求，例如寄存器初始化、中断调用格式和内存操作类型明确性。调试工具（如 TD）帮助快速定位运行时错误，而分模块测试的方法能有效减少错误累积。未来编程中需更注重逐行验证和提前规划代码逻辑，避免因低级错误耗费调试时间，同时也巩固了对分支循环结构和数据转换的理解，为复杂程序开发打下基础。

五、指导教师评语

成 绩		批阅人		日 期	
-----	--	-----	--	-----	--