

# 实验一 仿真软件使用与汇编程序调试方法

## 一、实验目的

1. 学习 debug 的用法
2. 掌握相关命令的使用

## 二、实验设备和仪器

1. 微型计算机 (Intel x86 系列 CPU) 1 台
2. Windows /XP/Win7/win10 操作系统

## 三、实验内容及要求

- 1、查看寄存器中的内容。
- 2、查看内存中的内容

PC 主板中有一个生产日期，在内存 FFF00~FFFFFH 的某几个单元中。请找到这个生产日期并试图改变它。

- 3、向内存 B8100H 开始的单元中填写数据，如：

1, 2, 3, 4, 5, 6, 7, 8 等等。

4、使用 DEBUG，将下列的程序段写入内存，逐条执行，观察每条指令执行后 CPU 中相关寄存器中的内容变化

```
MOV AX, 4E20H
ADD AX, 1416H
MOV BX, 2000H
ADD AX, BX
MOV BX, AX
ADD AX, BX
MOV AX, 001AH
MOV BX, 0026H
ADD AL, BL
ADD AH, BL
ADD BH, AL
MOV AH, 0
ADD AL, BL
ADD AL, 9CH
```

## 四、实验原理及步骤

步骤如下：

- (1) 进入 Debug 的方法
- (2) 用 R 命令查看、改变 CPU 寄存器的内容

R 查看寄存器内容

改变寄存器的值：R 相关寄存器

- (3) 用 D 命令查看内存中的内容

查看内存中的内容方法：D 段地址：偏移地址

用 D 命令查看内存范围的方法：D 段地址：起始偏移地址 结尾偏移地址

- (4) 改写内存中的内容

方法 1：用 E 命令改写内存中的内容。比如将 2000:0~2000:9 单元中的内容分别写为 0、1、2、3、4、5、6、7、8、9。用命令：E 起始地址 数据 数据。。。的格式。  
即：e 2000:0 0 1 2 3 4 5 6 7 8 9。

方法 2：也可以逐个逐个的修改：E 2000:0 按回车。

- (5) 查看内存中原有的机器码所对应的汇编指令

方法：U SA:EA

- (6) 用 T 命令单步执行

- (7) 用 A 命令以汇编指令的形式在内存中写入机器指令。

方法一：A SA:EA

方法二：A

## 五、实验结果分析及实验报告

## 实验二 数据处理实验

### 一、实验目的

1. 掌握键盘输入字符的用法
2. 掌握顺序结构的设计方法
3. 进一步掌握调试工具的使用方法

### 二、实验设备和仪器

1. 微型计算机 (Intel x86 系列 CPU) 1 台
2. Windows /XP/Win7 操作系统

### 三、实验内容及要求

- 1、编程从键盘上输入一个字符并显示。
- 2、编程实现 2 的 4 次方。

### 四、实验原理及步骤

#### 1、单字符显示

ASSUME CS:CODE

CODE SEGMENT

MOV AH,1; 从键盘上输入一个字符

IN 21H

MOV DL,AL ;显示字符

MOV AH,2

INT 21H

MOV AH,4CH

INT 21H

CODE ENDS

END

#### 2、

CODE SEGMENT

ASSUME CS:CODE

START: MOV AX,2

ADD AX,AX

MOV AX,AX

MOV AX,AX

MOV AH,4CH

INT 21H

CODE ENDS

END START

### 五、实验结果分析及实验报告

## 实验三 分支程序设计

### 一、实验目的

1. 学习分支指令的用法
2. 掌握分支结构程序的设计方法

### 二、实验设备和仪器

1. 微型计算机 (Intel x86 系列 CPU) 1 台
2. Windows /XP/Win7 操作系统
3. 微软 MASM5.0 程序包

### 三、实验内容及要求

某班 20 人，统计成绩 90 分以上，80~89，70~79，60~69 分及 60 分以下的人数。当前数据段中 DATA1 开始的 20 个单元中，存放在某课程的考试成绩。统计人数存放在同一数据段的 DATA2 开始的 5 个单元中。

### 四、实验原理及步骤

DATA SEGMENT

DATA1 DB 50,61,62,78,65,89,90,76,88,91,52,68,78,95,81,82,75,82,55,87

DATA2 DB 5 DUP (0)

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE, DS:DATA

START:

MOV AX,DATA

MOV DS,AX

MOV CX,20

LEA SI, DATA1

LEA DI, DATA2

AGAIN: MOV AL,[SI]

CMP AL,90

JC NEXT1

INC BYTE PTR[DI]

JMP ST

NEXT1: CMP AL,80

JC NEXT2

INC BYTE PTR[DI+1]

JMP ST

NEXT2: CMP AL,70

JC NEXT3

INC BYTE PTR[DI+2]

JMP ST

```
NEXT3: CMP AL,60
        JC NEXT4
        INC BYTE PTR[DI+2]
        JMP ST
NEXT4:  INC BYTE PTR[DI+4]
ST:     INC SI
        LOOP AGAIN
        MOV AH,4C00H
        INT 21H
CODE ENDS
END START
```

## 五、实验结果分析及实验报告要求

## 实验四 循环程序设计

### 一、实验目的

1. 学习循环指令的用法
2. 掌握分支、循环结构程序的设计方法

### 二、实验设备和仪器

1. 微型计算机 (Intel x86 系列 CPU) 1 台
2. Windows /XP/Win7 操作系统
3. 微软 MASM5.0 程序包

### 三、实验内容及要求

利用循环编写 10 个数的排序程序。

### 四、实验原理及步骤

#### 1. 实验原理

循环程序是把一段程序代码重复执行多次的程序结构。循环程序包括：初始化、循环体和循环控制等三个部分。初始化部分用于对循环参数（如循环次数、控制条件、指针等）设置初值；循环体是被重复执行的程序段；循环控制部分用于决定是否退出循环。

#### 2. 步骤

- (1) 确定源程序的存放目录
- (2) 编写程序，建立 ASM 源文件

本程序中，为找到 10 个数中的最大值，方法是将第一个数与后面的第二个数相比较，如果比后面的数大，则交换；依次类推，第二个数与第三个数，第三个数与第四个数……，将所有的数比较一遍后，最大的数就会在数列的最后面。

程序框图后附。

- (3) 用 MASM.EXE 汇编源程序产生 OBJ 目标文件
- (4) 用 LINK.EXE 产生 EXE 可执行文件
- (5) 执行程序
- (6) 如出现错误，利用 TD.EXE 调试程序，重复 (3) ~ (5)，直到运行结果正确

DATA SEGMENT

BUF DW 3, 1, 2, 4, 5, 7, -6, 8, -8, 10

N=(\$-BUF)/2

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE, DS:DATA

START: MOV AX, DATA

MOV DS, AX

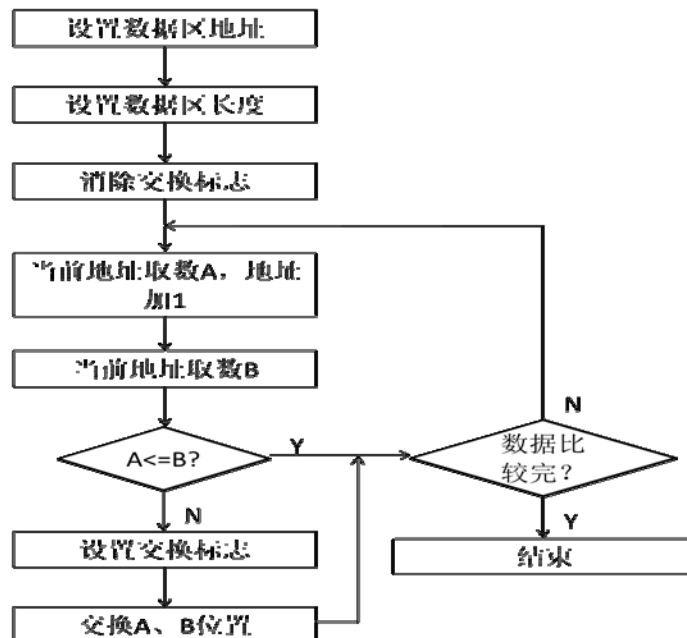
```

MOV CX, N
DEC CX
LOOP1: MOV DX, CX
MOV BX, 0
LOOP2: MOV AX, BUF[BX]
CMP AX, BUF[BX+2]
JGE L;
XCHG AX, BUF[BX+2]
MOV BUF[BX], AX
L: ADD BX, 2
DEC CX
JNE LOOP2
MOV CX, DX
LOOP LOOP1
MOV AH, 4CH
INT 21H
CODE ENDS

```

## END START 五、实验结果分析及实验报告要求

1. 了解数据排序的简单算法，如“冒泡排序法”。附程序框图：



## 实验五 系统中断调用

### 一、实验目的

1. 掌握中断程序的概念以及如何响应中断

### 二、实验设备和仪器

1. 微型计算机 (Intel x86 系列 CPU) 1 台
2. Windows /XP/Win7 操作系统
3. 微软 MASM5.0 程序包

### 三、实验内容及要求

编写程序：溢出的中断程序编写

### 四、实验步骤

```
assume cs:code
code segment
start:mov ax,cs
      mov ds,ax
      mov si,offset do0      ;设置 ds:si 指向源地址
      mov ax,0
      mov es,ax
      mov di,200h            ;设置 es:di 指向目的地址
      mov cx,offset do0end - offset do0;设置 cx 为传输长度
      cld                    ;设置传输方向为正
      rep movsb

      ;设置中断向量表

      mov ax,4c00h
      int 21h

do0:  jmp short do0start
      db "overflow!"

do0start:
      mov ax,cs
      mov ds,ax
      mov si,202h            ;设置 ds:si 指向字符串

      mov ax,0b800h
      mov es,ax
      mov di,12*160+36*2      ;设置 es:di 指向显存空间的中间位置

      mov cx,9                ;设置 cx 为字符串长度
s:    mov al,[si]
      mov es:[di],al
      inc si
```



```
        add di,2
        loop s

        mov ax,4c00h
        int 21h
do0end:nop

code ends
end start
```

## 五、实验结果分析及实验报告要求

- 1、完整的程序
- 2、分析该程序的中断向量地址以及如何响应中断程序。

# 实验六 判断闰年的程序设计

## 一、实验目的

1. 掌握子程序设计的基本方法，包括子程序的定义、调用和返回，子程序中如何保护和恢复现场，主程序与子程序之间如何传送参数
2. 学习如何进行数据转换的处理方法

## 二、实验设备和仪器

1. 微型计算机 (Intel x86 系列 CPU) 1 台
2. Windows /XP/Win7 操作系统
3. 微软 MASM5.0 程序包

## 三、实验内容及要求

编写程序，从键盘上输入一个四位数的年份，判断其是否为闰年。要求写出三个子程序：

1. 输入年份是否合法
2. 将输入的年份字符转化成数字
3. 判断该年份是否为闰年，判断方法为（1）能被 4 整除但不能被 100 整除，或（2）能被 400 整除

## 四、实验原理及步骤

### 1. 实验原理

在实际编程中，经常会遇到功能完全相同的程序段，或不在同一程序模块，或虽在同一模块而需重复执行，但又不是连续重复执行。为了避免重复编制同一段程序，节省存储空间，把程序段独立开来，附加少量额外语句，将其编制成公用子程序，供程序其他地方需要时调用，这种程序设计方法称为子程序设计。

在使用子程序时应注意以下三点：参数的传递、相应寄存器内容的保护、子程序的嵌套。

### 2. 步骤

（1）确定源程序的存放目录

（2）编写程序，建立 ASM 源文件

程序框图后附。

程序清单后附

（3）用 MASM.EXE 汇编源程序产生 OBJ 目标文件

（4）用 LINK.EXE 产生 EXE 可执行文件

（5）执行程序

（6）如出现错误，利用 TD.EXE 调试程序，重复（3）~（5），直到运行结果正确

程序清单：

```
DATA SEGMENT
```

```
ENTER1 DB ' Please input the year',0DH,0AH,'$'
```

```
ERR1 DB 0DH,0AH,' INPUT ERROR',0DH,0AH,'$'
```

```
YESLEAP DB 0DH,0AH,' THE YEAR IS LEAP',0DH,0AH,'$'
```

```
NOLEAP DB 0DH, 0AH, 'THE YEAR IS NOT LEAP', 0DH, 0AH, '$'
YEAR DB 5, ?, 5 DUP(?) ;输入年份字符所在的存储区，第一个 5 表示最多可输入 5 个字
符，紧接着? 存放实际的字符数，后面 5 个存储区存放实际输入的字符
DATA ENDS
```

```
ASSUME CS:CODE, DS:DATA
```

```
CODE SEGMENT
```

```
START:
```

```
    MOV AX, DATA
```

```
    MOV DS, AX
```

```
    CALL INPUT;输入年份字符串并判断是否合法子程序
```

```
    CALL CHARINTONUM ;将字符串变成数字子程序
```

```
    CALL LEAP      ;判断闰年的子程序
```

```
    MOV AX, 4C00H
```

```
    INT 21H
```

```
INPUT PROC;输入年份字符串并判断是否合法
```

```
    INPUTYEAR:  LEA DX, ENTER1    ;显示提示字符
```

```
                MOV AH, 09H
```

```
                INT 21H
```

```
                LEA DX, YEAR      ;输入年份
```

```
                MOV AH, 0AH
```

```
                INT 21H
```

```
                MOV CL, YEAR+1    ;字符个数存放在 cx 中
```

```
                MOV CH, 0
```

```
                LEA SI, YEAR+2
```

```
JUDGE1: CMP BYTE PTR [SI], 30H ;判断输入的字符是否在 '0'-'9' 之间
```

```
        JB ERROR
```

```
        CMP BYTE PTR [SI], 39H
```

```
        JA ERROR
```

```
        INC SI
```

```
        LOOP JUDGE1
```

```
        JMP R1
```

```
ERROR:  LEA DX, ERR1;显示输入格式错误
```

```

        MOV AH, 09H
        INT 21H
        JMP INPUTYEAR ;输入错误后重新输入
R1:     RET
INPUT ENDP

```

CHARINTONUM PROC;将输入的字符转化成数字年份，存放在 AX 寄存器中

```

        MOV CL, YEAR+1
        MOV CH, 0
        MOV AX, 0
        LEA SI, YEAR+2
S1:     MOV BX, 0
            MOV BL, 10
        MUL BX
        MOV BL, [SI]
        MOV BH, 0
        SUB BX, 30H
        ADD AX, BX
        INC SI
        LOOP S1
        ret
CHARINTONUM ENDP

```

LEAP PROC ;判断寄存器 AX 存放的年份是否为闰年

```

        MOV CX, AX
        MOV DX, 0
        MOV BX, 4 ;判断能被 4 整除
        DIV BX
        CMP DX, 0
        JNZ NO
        MOV AX, CX

```

```

        MOV DX, 0
        MOV BX, 100 ;判断能被 100 整除
        DIV BX
        CMP DX, 0
        JNZ YES
        MOV AX, CX
        MOV DX, 0
        MOV BX, 400 ;判断能被 400 整除
        DIV BX
        CMP DX, 0
        JNE NO
YES:    LEA DX, YESLEAP
        MOV AH, 09H
        INT 21H
        JMP EXITNUM
NO:     LEA DX, NOLEAP
        MOV AH, 09H
        INT 21H

EXITNUM: RET
LEAP ENDP

CODE ENDS
END START

```

## 五、实验结果分析及实验报告要求

- 1、画出程序框图
- 2、算法分析及子程序
- 3、程序调试

# 实验七 可编程并行接口实验

## 一、实验目的

1、了解并行通讯的基本原理，掌握并行接口芯片 8255 的工作原理和编程方法。

## 二、实验设备和仪器

- 1. 微型计算机（Intel x86 系列 CPU）1 台
- 2. Windows /XP/Win7 操作系统
- 3. 微软 MASM5.0 程序包
- 4. 实验箱

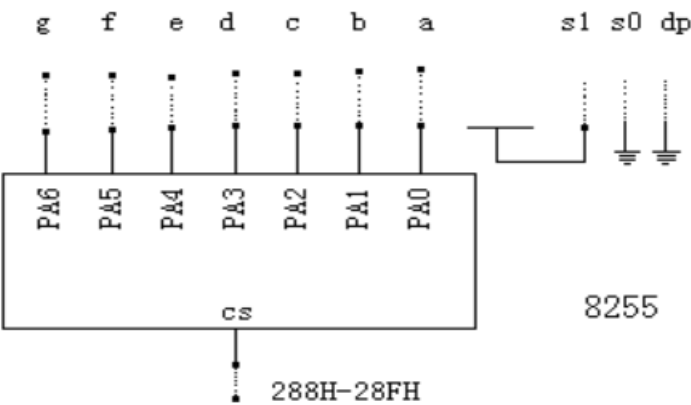
## 三、实验内容及要求

编写程序：编程实现从键盘上输入一个数字（0-9），将数字在七段数码管上显示出来。  
按数字键外的任意键退出并返回 DOS。

## 四、实验原理及步骤

### 1. 实验原理

1、将 8255 的 A 口 PA0~PA6 分别与七段数码管的段码驱动输入端 a~g 相连，位码驱动输入端 S1 接+5V，S0、dp 接地，CS 接 288H~28FH。



### 2. 数码管字形对应的段码表

显示字形	段码
0	3FH
1	06H
2	5BH
3	4FH
4	66H
5	6DH
6	7DH
7	07H

8	7FH
9	6FH

### 3. 步骤

(1) 确定源程序的存放目录

(2) 编写程序，建立 ASM 源文件

程序框图后附。

程序清单后附

(3) 用 MASM. EXE 汇编源程序产生 OBJ 目标文件

(4) 用 LINK. EXE 产生 EXE 可执行文件

(5) 执行程序

(6) 如出现错误，利用 TD. EXE 调试程序，重复 (3) ~ (5)，直到运行结果正确

程序清单：

DATA SEGMENT

ENTER DB 0AH, 0DH, '\$'

MSG DB 0AH, 0DH, 'PLEASE INPUT A NUMBER (0—9):', '\$'

seg7 db 3FH, 06H, 5BH, 4FH, 66H, 6DH, 7DH, 07H, 7FH, 6FH; 0-9 对应的段码

DATA ENDS

CODE SEGMENT

ASSUME DS:DATA, CS:CODE

START:MOV AX, DATA

MOV DS, AX

MOV DX, 028BH

MOV AL, 80H

OUT DX, AL

INPUT: MOV DX, OFFSET MSG

MOV AH, 09H

INT 21H

MOV AH, 01H;将数字回显

INT 21H

CMP AL, 30H;将数字和 30H (0) 作比较

JC FINISH ;若 AL 小于'0' (ASCII 值为 48) ;则跳转到 FINISH

CMP AL, 39H;将数字和 39H (9) 作比较

JA FINISH;若 AL 大于'9' (ASCII 值为 57) 则跳转到 FINISH

```
SUB AL, 30H
MOV AH, 0
MOV SI, AX
MOV DX, 0288H
MOV AL, [SEG7+SI]
OUT DX, AL
JMP INPUT                ;跳回 INPUT
FINISH:
MOV AH, 4CH              ;结束
INT 21H
CODE ENDS                ;代码段结束
END START                ;程序结束
```

## 五、实验结果分析及实验报告要求

- 1、画出程序框图
- 2、算法分析
- 3、程序调试