汇编语言程序设计实验报告

实验时间 2024 **年** 5 **月** 7 **日** **第** 11 **周** 计算机与信息 **学院** **学号** 202110120428

姓名 毛颂凯 **指导老师** 肖明 **得分**_____

实验名称：汇编语言程序设计实验

实验目的：

1) **熟悉汇编语言集成实验环境 MASM2015 的使用方法**
2) **熟练掌握编写汇编语言源程序的基本方法和基本框架。**
3) **学会编写顺序、分支和循环结构的程序设计。**
4) **掌握程序指令 JZ、JNZ、JA、JG、LOOP 等转移指令及应用。**

一、实验环境及材料：

1、硬件环境配置：

CPU：11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz 2.30 GHz

内存：16.0 GB

硬盘：HFM512GD3JX013N

2、软件名称、版本号及参数配置：

操作系统：Windows 11 家庭中文版

仿真软件：VS code、MASM2015、Proteus

二、实验内容：

1、验证性实验思考题

(1) 编辑如下源程序，保存项目为 EXAM1，在 MASM2015 中直接运行观测实验现象。

(2)

问题一：若将指令 INT 21H 错打成 INT 21，会有什么问题?

将指令 INT 21H 错误地打成 INT 21，会导致程序在执行时无法正确执行，本应该会打印 BUF 字符串，最终并没有打印，程序就结束了。因为在 8086 汇编中，INT 21H 是调用 DOS 的系统调用功能，而 INT 21 并不是一个有效的中断向量号。

问题二：若去掉第 10 行和第 11 行的指令，可以吗？为什么？

不可以，去掉 MOV AX, DATAS 和 MOV DS, AX 这两条指令，那么代码段中的指令就无法正确地访问数据段中的数据。这是因为在 8086 汇编中，数据段的段地址需要加载到数据段寄存器 DS 中，才能够正确地访问数据段中的变量或数据。

当使用指令 MOV AX, DATAS 将数据段 DATAS 的段地址加载到 AX 寄存器时，实际上是在告诉计算机将数据段 DATAS 视为当前要操作的数据段。然后，通过 MOV DS, AX 将 AX 中的段地址复制到 DS 寄存器，这样 DS 寄存器就指向了数据段 DATAS，代码段中的指令才能够正确地访问数据段中的变量或数据。因此，去掉这两条指令将导致代码段无法正确地访问数据段中的数据，在本程序中会输出乱码。

(3) 编辑如下源程序，保存项目为 EXAM2，在 MASM2015 中直接运行观测实验现象，在 MASM2015 中调试，观测 DATA 段的变化。

问题一：程序的功能是什么？

这段程序的功能是从字符串"A1"中提取小写字母，然后将其存储到另一个字符串"A2"中，并最终将"A2"字符串输出显示出来。

问题二：N 求的是什么？

N 是字符串 A1 的大小。

问题三：程序中 SI 和 DI 起什么作用？

在程序中，SI 和 DI 是两个指针寄存器。SI 用于指向源字符串"A1"，而 DI 用于指向目标字符串"A2"。在循环中，SI 逐个读取"A1"中的字符，然后通过 DI 将符合条件的小写字母存储到"A2"中。

问题四：DATA 段中 A2 是如何变化的？

数据段中，A2 是通过声明为 N 个字节的重复符号"$"而变化的。

“$$$$$$$$” → “b$$$$$$$$” →“bc$$$$$$$”→“bcd$$$$$$$”→“bcda$$$$$$”→ “bcdag$$$$$”

→ “bcdagh$$$$” (在实际内存单元中存的是 ASCII 码)

(4) 编辑如下源程序，保存项目为 EXAM3，在 MASM2015 中直接运行观测实验现象。

问题一：程序的功能是什么？

输出 BIN 的十进制。

问题二：PWTAB 在程序中起到什么作用？

用于每次取 BIN 十进制的每一位上的数字是多少，例如取最高位就要除以 10000，得到商和余数，商就是最高位的数字；取次高位，用余数再除以 1000，按着这样就可以依次取出 BIN 十进制的每一位上的数字是多少。

问题三：第 22 行指令 OR　AL,30H 有什么作用？

将 AL 寄存器中的值转换为对应的 ASCII 字符。在 ASCII 码表中，数字 0 到 9 的 ASCII 码值是 30H 到 39H，所以通过将 AL 的值与 30H 进行 OR 运算，可以将 AL 中的值转换为对应的 ASCII 数字字符。

2、设计性实验代码和分析

(1) 编写程序任意输入两个字符串并判断它们是否相同，若相同则输出 YES，否则输出 NO。（文件命名为“汇编语言程序设计实验 1.asm”）

思路：首先比较字符串的大小，如果相等，在比较字符串的每一位是否相等。所以直接从两个字符串的偏移地址 ＋1 开始比较，循环次数为字符串大小 ＋1。

```
DATAS SEGMENT
    STRING1 DB 80,?,80 DUP(?)
    STRING2 DB 80,?,80 DUP(?)
    YES DB 0DH, 0AH,"YES$"
    NO DB 0DH, 0AH,"NO$"
    ENDL DB 0DH,0AH,"$"
DATAS ENDS

STACKS SEGMENT
    DW 100H DUP(?)
STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX

    LEA DX,STRING1
    MOV AH,0AH
    INT 21H

    LEA DX,ENDL
    MOV AH,09H
    INT 21H

    LEA DX,STRING2
    MOV AH,0AH
    INT 21H

    LEA SI,STRING1
    INC SI

    LEA DI,STRING2
    INC DI

    MOV CX,0
    MOV CL,[SI];
    INC CX

CYCLE:
    MOV AL,[SI]
    MOV BL,[DI]
    CMP AL,BL
    JNE PRINT_NO
    INC SI
    INC DI
LOOP CYCLE

PRINT_YES:
    MOV DX,OFFSET YES
    MOV AH,09H
    INT 21H
    JMP OVER

PRINT_NO:
    MOV DX,OFFSET NO
    MOV AH,09H
    INT 21H
```

```
    JMP OVER

OVER:
    MOV AH,4CH
    INT 21H

CODES ENDS
    END START
```

结果验证

输入： 123asd123、123asd123；输出：YES

输入： 123456、125556；输出：NO

(2) 变量 BUF 中存放着 10 个有符号的字节数据，编程将这 10 个数按从小到大排序，不要求输出。

（文件命名为“汇编语言程序设计实验 2.asm”）

思路：

1. 循环比较和交换：从第一个元素开始，依次比较相邻的两个元素，如果顺序不对（比如较大的元素在前面），则交换它们的位置。
2. 一轮循环结束后：最大的元素会移动到最后一个位置。
3. 重复执行：针对剩余未排序的元素，重复上述比较和交换的过程，直到所　有元素都排好序。

下表从 0 开始，第一层循环 $i\ (n-1\sim 0)$，第二层循环 $j(0\sim i-1)$

```
DATAS SEGMENT
    BUF DB 9,8,7,6,5,4,3,2,1,0
    ENDL DB 0DH,0AH,"$"

DATAS ENDS

STACKS SEGMENT
    DW 100H DUP(?)
STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS

START:
    MOV AX,DATAS
    MOV DS,AX

    MOV CX,9
    MOV SI,OFFSET BUF

;冒泡排序
LOOP1:
    MOV DI,CX       ; 将 CX 的值赋给 DI，用于内循环计数
    MOV BX,0

LOOP2:
    MOV AL,[SI + BX]
    CMP AL,[SI + BX + 1]
    JL NEXT
    XCHG AL,[SI + BX + 1]   ;大的数往高地址处移动
    MOV [SI + BX],AL

NEXT:
    INC BX
    CMP BX,DI
    JL LOOP2
    LOOP LOOP1

    MOV AH,4CH
    INT 21H

CODES ENDS
    END START
```

结果

可以看到，已经是从小到大排序了。