

# 天津商业大学学生实验报告

开课实验室：403 机房

开课时间 2014 年 4 月 22 日

实验报告 2014 年 4 月 22 日

学院名称	信息工程学院	年级、专业、班	软件工程 1201	学号	20125041	姓名	王靖伟	同组姓名	无
课程名称	计算机组成原理	实验项目名称	实验二 分支结构程序设计			指导教师	高珊		
实验类型	验证 <input type="checkbox"/> 综合 <input type="checkbox"/> 设计 <input type="checkbox"/> 创新 <input type="checkbox"/>							成绩	
教师评语	教师签名：_____ 年 月 日								
实验报告内容一般包括以下几个内容：1、目的要求 2、仪器用具及材料（仪器名称及主要规格、用具名称） 3、实验内容及原理（简单但要抓住要点，写出依据原理） 4、操作方法与实验步骤 5、数据图表格（照片） 6、实验过程原始记录 7 数据处理及结果（按实验要求处理数据、结论） 8、作业题 9、讨论（对实验中存在的问题、进一步的想法等进行讨论）									
<p><b>实验报告内容：</b></p> <p><b>一、实验目的</b></p> <p>1、熟悉 Masm for windows 集成实验环境 2012，掌握汇编语言常用指令的使用方法。</p> <p>2、掌握字符串的输入输出方法、分支结构程序设计的步骤。</p> <p>3、掌握 Debug 调试程序的基本方法，并能够根据各专用寄存器的存储结果判断分支结构程序中各分支程序运行的正确性。</p> <p><b>二、实验环境</b></p> <p>Masm for windows 集成实验环境 2012</p> <p><b>三、实验内容</b></p> <p>1、字符串输入输出方法编程</p> <p>2、编写一个判断两个字符串是否匹配的程序。</p> <p><b>四、实验步骤</b></p> <p>1、字符串输入输出方法</p> <p style="margin-left: 40px;">DOS 系统调用 INT 21H 提供了字符及字符串的 I/O 功能，例如：</p> <p style="margin-left: 80px;">01H：从键盘读入一个字符</p> <p style="margin-left: 80px;">02H：显示一个字符</p> <p style="margin-left: 80px;">09H：显示一个字符串</p>									

注 1. 每个实验项目一份实验报告。2. 实验报告第一页学生必须使用规定的实验报告纸书写，附页用实验报告附页纸或 A4 纸书写，字迹工整，曲线要画在坐标纸上，线路图要整齐、清楚（不得徒手画）。3. 实验教师必须对每份实验报告进行批改，用红笔指出实验报告中的错、漏之处，并给出评语、成绩，签全名、注明日期。4. 待实验课程结束以后，要求学生把实验报告整理好，交给实验指导教师，加上实验课学生考勤及成绩登记表（见附件 2）、目录和学院统一的封面（见附件 3）后，统一装订成册存档。

# 天津商业大学学生实验报告附页

开课实验室：403 机房

开课时间 2014 年 4 月 22 日

实验报告 2014 年 4 月 22 日

0AH: 从键盘读入一个字符串

## (1) 从键盘读入一个字符

功能号：01H

入口参数：AL = 输入字符的 ASCII 码

功能：等待从键盘读入一个字符，将其 ASCII 码送入 AL，同时将该字符显示在屏幕上。

调用方法：

```
mov    ah, 1
int     21h
```

说明：输入一个字符后，不需要回车。若只键入回车，则 AL = 0DH。

## (2) 显示一个字符

功能号：02H

入口参数：DL = 要显示字符的 ASCII 码

功能：在当前光标位置显示 DL 中的字符，光标右移。

调用方法示例：

```
mov dl, 'A'    ; 显示字符'A'
mov ah, 2
int 21h
```

说明：显示字符 0DH 和 0AH 将产生回车和换行的作用。

## (3) 显示一个字符串

功能号：09H

入口参数：DS:DX = 欲显示字符串在内存的首地址，且字符串必须以 '\$' (24H) 作为结束符。

功能：在当前光标位置，显示由 DS:DX 所指的、以 '\$' 结尾的字符串，且光标右移。其中，'\$' 不算在显示的字符串之内。

调用方法示例：

```
String db 'Hello!', '$'; 显示字符串'Hello!'
mov ax, seg String
mov ds, ax
```

mov dx, offset String; 操作符 offset 在汇编语言中是由编译器处理的符号，它的功能是取得标号的偏移地址。

```
mov ah, 9
```

```
int 21h
```

#### (4) 从键盘读入一个字符串

功能号: 0AH

入口参数: DS:DX = 输入缓冲区首地址

输入缓冲区格式: 第 0 字节事先设置为缓冲区最多能容纳的字符个数 (包括回车); 第 1 字节将存放实际字符个数 (不包括回车), 由系统自动设置; 从第 2 字节开始存放实际输入的字符串, 最后为回车符。

功能: 从键盘读入一个字符串, 存入 DS:DX 所指的缓冲区。

调用方法示例:

```
buf db 10, ?, 10 dup (?)    ; 定义输入缓冲区

mov ax, seg buf
mov ds, ax
lea dx, buf
mov ah, 0ah
int 21h
```

2、编写一个判断两个字符串是否匹配的程序。

(1) 要求字符串保存在缓冲区中, 字符串可由键盘输入, 也可以采用固定的字符串声明方法。

(2) 两个字符串是否相等, 可通过寄存器的结果显示, 如两字符串相等, 则寄存器 AL=0, 否则 AL=1。也可以通过屏幕输出结果。

算法如下:

**字符串定义:**

BUF1 DB 'ABCDEFGH'; 此处输入数据段代码

COUNT1 EQU \$-BUF1; 定义了一个常量, 不占用内存单元, 代码段中使用它, 等价于使用一个立即数。

**使用的主要指令:**

两字符串赋值:

```
LEA SI, BUF1; 源串段寄存器
LEA DI, BUF2; 目标串段寄存器
MOV CX, COUNT1
```

比较两个字符串的长度

```
CMP CX, COUNT2; 比较指令, 两数进行相减比较, 不保存结果, 若相减结果为0 (相等), 则ZF=1
JNZ EXIT1; 条件转移指令, ZF=0 (不相等), 转至标号处执行
```

比较两个字符串的内容

```
CLD; DF=1
REPE CMPSB; REPE是重复前缀的说明, 在CF=0时退出; CMPSB字符串比较指令, 若相等, 则ZF=1
```

为AL赋值的语句

```
JNZ EXIT1
```

```
MOV AL, 0
    JMP EXIT2
EXIT1:
MOV AL, 1
EXIT2:
MOV AH, 4CH
INT 21H
```

五、实验结论

1、定义两个相同字符串，输出结果为 y（两字符串相同）。

```
BUF1 DB 'ABCDEFGH'
BUF2 DB 'ABCDEFGH'
```

图 1 在数据段定义两个相同的字符串

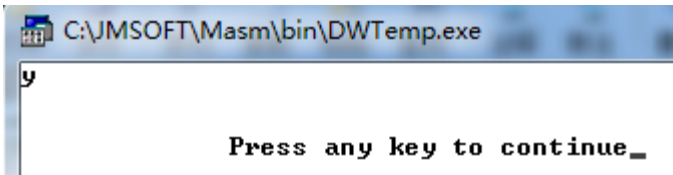


图 2 输出结果为 y（两字符串相同）

2、定义两个不相同字符串，输出结果为 n（两字符串不相同）。

```
BUF1 DB 'ABCDEFGH'
BUF2 DB 'Adadadq'
```

图 3 在数据段定义两个不相同的字符串

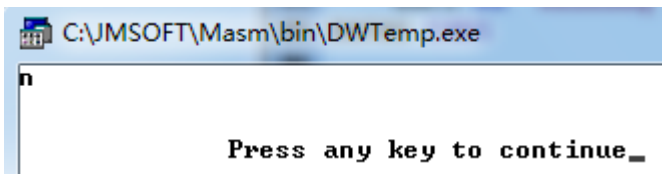


图 4 输出结果为 n（两字符串不相同）

六、实验分析

行数	代码
1	DATAS SEGMENT
2	;此处输入数据段代码
3	BUF1 DB 'ABCDEFGH'
4	BUF2 DB 'ABCDEFGH'
5	DATAS ENDS
6	
7	STACKS SEGMENT
8	;此处输入堆栈段代码
9	STACKS ENDS

```

10
11 CODES SEGMENT
12     ASSUME CS:CODES, DS:DATAS, ES:DATAS, SS:STACKS
13 START:
14     MOV AX, DATAS
15     MOV DS, AX
16     mov ES, AX
17     ;此处输入代码段代码
18
19     LEA SI, BUF1;源串段寄存器
20     LEA DI, BUF2;目标串段寄存器
21     CLD             ;DF=1
22     MOV CX, 8       ;循环比对的次数
23     REPE CMPSB;REPE是重复前缀的说明，在CF=0时退出
24                 ;CMPSB字符串比较指令，若相等，则ZF=1
25     JNZ EXIT1
26     MOV AL, 0
27     mov DL, 'y'
28     JMP EXIT2
29 EXIT1:
30     MOV AL, 1
31     mov DL, 'n'
32 EXIT2:
33     mov AH, 2
34     INT 21H
35     MOV AH, 4CH
36     INT 21H
37 CODES ENDS
38     END START

```