**实验报告**

**实验序号：3 实验项目名称： 编写子程序**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学　　号 |  | 姓 名 |  | 班 级 |  |
| 实验地点 |  | 指导教师 |  | 实验成绩 |  |
| **一、实验目的**  1）学会使用标志寄存器；  2）熟练使用通过检测比较结果来转移的转移指令。  **二、实验环境(设备**)  编辑器：Visual Studio 2022  汇编程序：MASM.EXE  调试程序：DEBUG.EXE  连接装配程序：LINK.EXE  **三、实验内容**      **四、实验结果及其分析**  **【说明】完整的实验代码附在文档最后。**  **1.程序分析**  **（1）如何判断字符串是否遍历结束：**因为这是一段字符串，因此可以通过观察是否遇到0来判断字符串是否遍历结束。  **（2）如何判断字符串中的字符是否是小写字母：**因为小写字母‘a’~‘z’是连续的ASCII码，因此可以通过比较单个字符串的ASCII码是否在‘a’和‘z’的区间内来判断是否属于小写字母。  **2.代码与结果测试**  **（1）代码**  **① datasg数据段程序代码：**    **② codesg代码段程序代码：**    **（2）结果测试**  ① 在执行前使用D DS:0命令查看字符串中的字符：可以看到，此时既有大写字符，又有小写字符。    ② 在DOS界面输入g，直接运行程序：    ③ 程序运行结束后，再使用D DS:0命令查看字符串中的字符：可以看到，此时小写字符已全部转化为大写字符。    **五、实验心得与讨论**  这次实验实现的功能是将小写字符串改为大写，程序本身不是很长、功能也不是很复杂，但是在具体编写的时候还是要用上很多知识点，要考虑的东西也很多。比如，我需要能够对各种转移指令非常了解并且能够灵活应用，此外，还要对字母的ASCII的规律非常熟悉以及对栈的使用。但是，通过实现该功能并不断处理和解决遇到的问题，我对这些知识点的掌握和使用也更加熟悉了。 | | | | | |
| **评语：**  1．按时完成实验；  2．实验内容和过程记录完整；  3．有实验的心得或讨论；  **老师：xx** | | | | | |

附完整代码：

assume cs:codesg

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*data数据段·开始\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

datasg segment

db "Beginner's All-purpose Symbolic Instruction Code.",0 ;注意0结尾

datasg ends

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*data数据段·结束\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*代码段·开始\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

codesg segment

begin: mov ax, datasg

mov ds, ax ;初始化ds

mov si, 0 ;ds：si指向字符串的首地址

call letterc ;调用子程序

mov ax, 4c00H

int 21H

; ----------------------------------------------

; 子程序letterc功能：将以0结尾的字符串中的小写字母转变成大写字母

; 入口参数：ds:si指向字符串首地址

; 返回值：无

;-----------------------------------------------

letterc: push ax

comp: mov al, [si] ;将字符读入al中

cmp al, 0 ;检测字符是否为0，字符串结尾检测

je exit ;如果为0，跳转到exit标号，退出

cmp al, 'a' ;与小写字母a比较

jb next\_char ;如果小于字符a，跳转到下个字符

cmp al, 'z' ;与小写字母z比较

ja next\_char ;如果大于字符z，跳转到下个字符

and byte ptr [si], 11011111B ;条件都不满足，开始转换为大写字母

;++++++下一个字符，递增si，跳转到comp标号++++++

next\_char: inc si

jmp short comp

;++++++字符串遍历结束，恢复寄存器，并返回主调程序+++++++

exit: pop ax ;恢复寄存器变量值

ret

codesg ends

end begin

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*代码段·结束\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*