**信息科学与工程学院**

**2017－2018学年第二学期**

实 验 报 告

课程名称： 微处理器原理与应用

实验名称： 实验2.2

专 业 班 级 通信工程 二班

学 生 学 号 201800121050

学 生 姓 名 孟麟芝

实 验 时 间 2020年2月26日

实验报告

【实验目的】

1. 掌握控制转移类指令的使用
2. 掌握分支结构程序的编写
3. 掌握简单的循环结构程序的编写
4. 掌握一些简单中断程序的使用

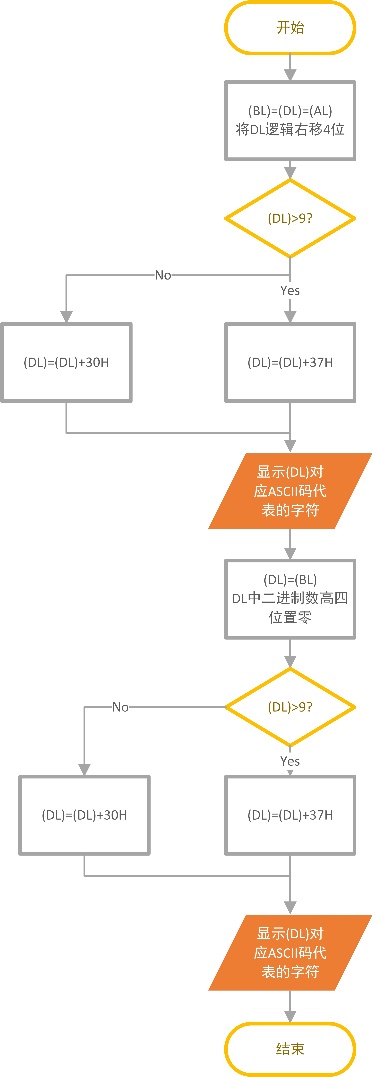
【实验要求】

1. 编写一个程序，将AL寄存器中的两位十六进制数显示出来
2. 编写一个程序，判别键盘上输入的字符；若是1-9字符，则显示之；若为A-Z或a-z字符，均显示＇c＇；若是回车字符(其ASCII码为0DH)，则自动结束程序，若为其它字符则不显示，循环等待新的字符输入。
3. 完成两个设计，截图显示实验结果，并贴出程序流程图

【实验具体内容】

【第一个实验】

1.程序流程图：



2.程序源代码：

CODE SEGMENT

ASSUME CS:CODE ;这里使用伪指令定义了一个段为CODE，assume使之与CS段相关联

START:MOV AL,3EH ;start设置了程序的入口，防止数据与程序同时存在一个段内时出现错误

MOV BL,AL

MOV DL,AL ;这两步对AL的存放可以实现将两位十六进制数的处理分开

MOV CL,4 ;把4赋给CL

SHR DL,CL ;将DL逻辑右移(CL)=4位（即不考虑符号），右移后第四位放入CF，高四位补0

CMP DL,9 ;使用cmp指令，通过结果不同对标志寄存器改变不同实现分支结构

JBE NEXT1 ;若(DL)<=9则直接进入NEXT1，配合后面加30H，可对应’0’~’9’的ASCII码

ADD DL,7 ;若(DL)>9则DL加7，则DL内为10D~15D时，配合后面加30H，已为65D~70D，对应’A’~’F’的ASCII码

NEXT1:ADD DL,30H ;如上所述，便于对应ASCII码

MOV AH,2

INT 21H ;这两步使用int指令，显示AL中二进制数高四位 ASCII 码

MOV DL,BL ;把(AL)再次放入DL中操作

AND DL,0FH ;进行逻辑与运算，将DL中二进制数高四位置零，第四位保持不变

CMP DL,9

JBE NEXT2 ;与CMP配合，当(DL)<=9进入NEXT2，与下面四行程序构成对低位的显示输出，与上面对高位的显示原理完全相同

ADD DL,7

NEXT2:ADD DL,30H

MOV AH,2

INT 21H ;显示低位 ASCII 码

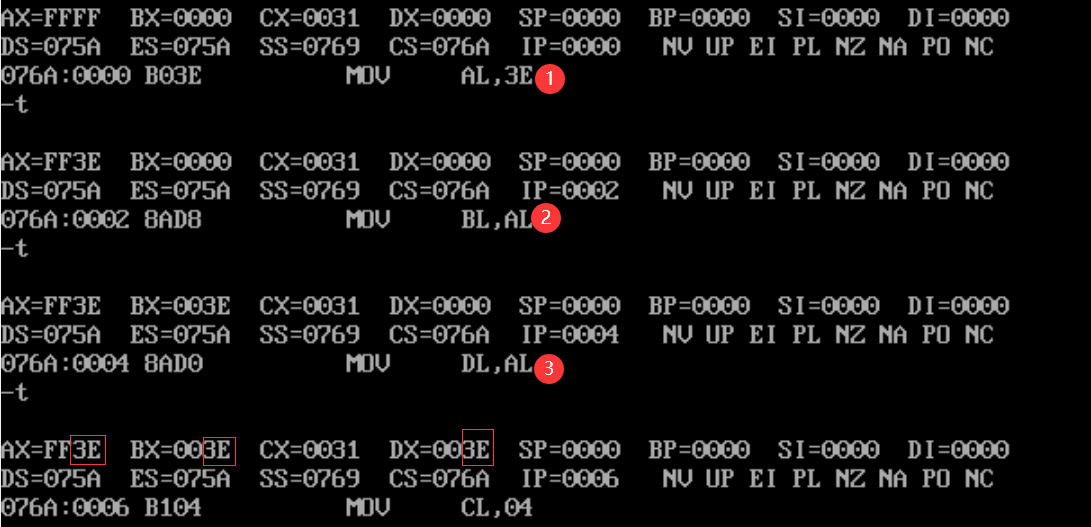
MOV AH,4CH ;

INT 21H ;

CODE ENDS ;返回DOS

END START

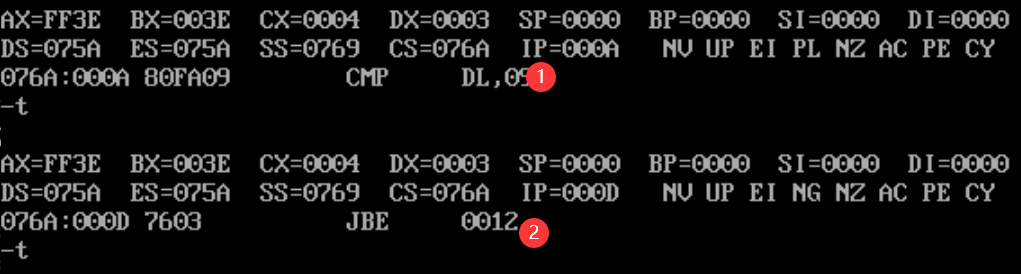
3.逐步调试验证



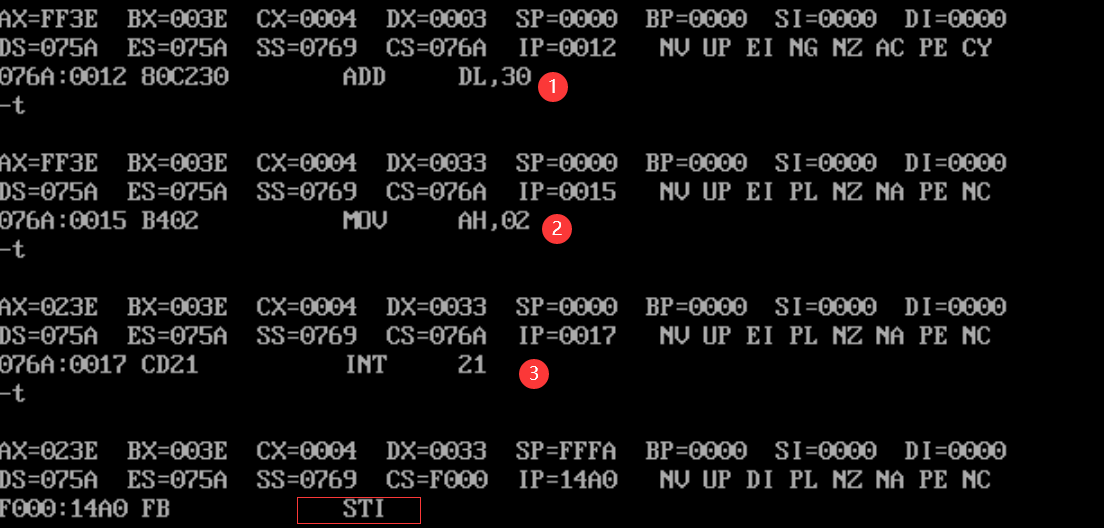
经过前三步，预设好的(AL)=3EH已被放入BL,DL



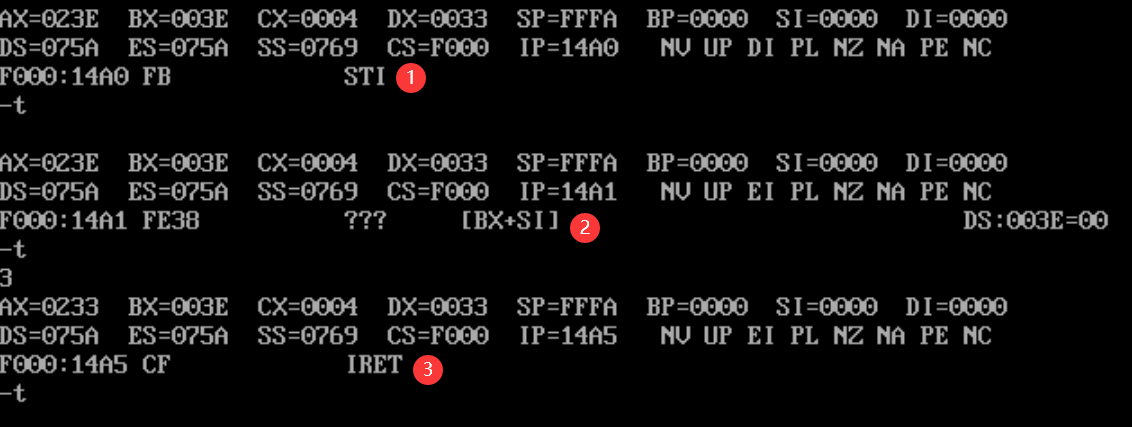
这两步实现了对DL中二进制数的逻辑右移四位，保留了(DL)的高位



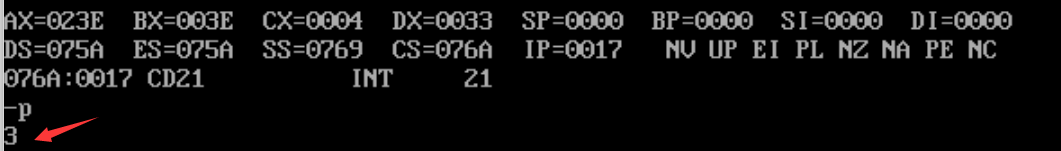
这两步对DL中的数判断是否大于9，以方便匹配不同的ASCII码，0012对应的是源程序中NEXT1第一条指令的偏移地址



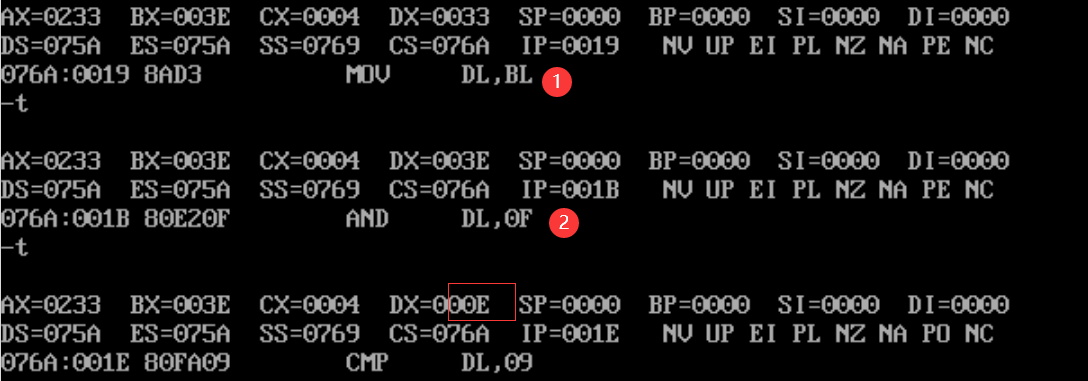
上面判断DL高位为小于9的数（3），故直接加30H，对应了’3’的ASCII码



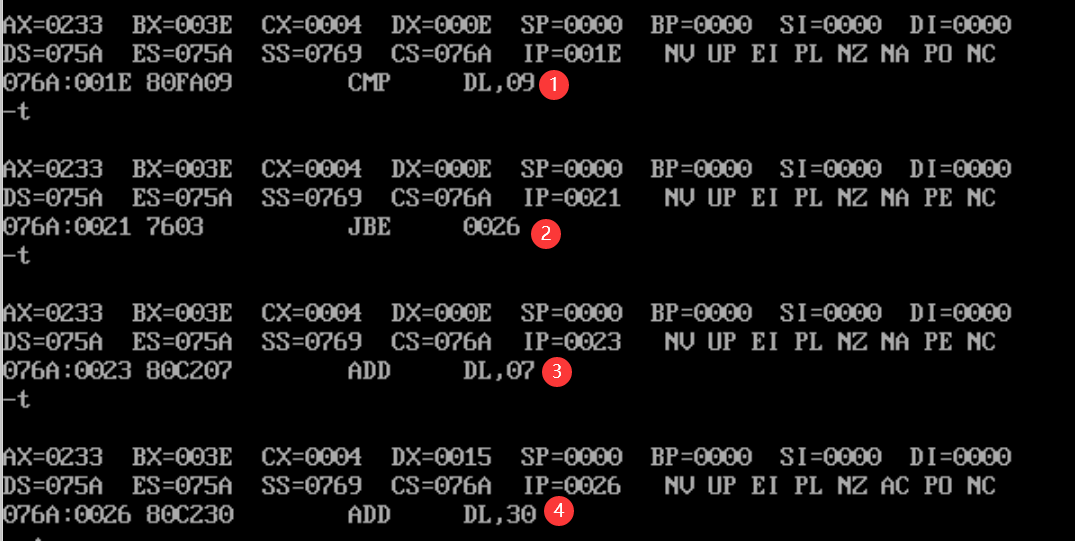
在这里，int21用-t命令调试是会出现问题的，如图所示，第一步STI允许中断，第二步则是未知指令，第三步又进行了返回，虽然程序也可以继续向下运行，不过已经导致了未知的操作。这里应该使用-p命令调试



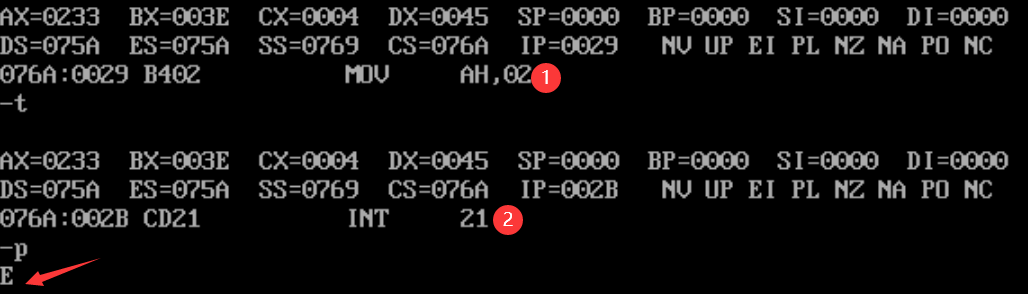
这样就看到了我们想要的结果



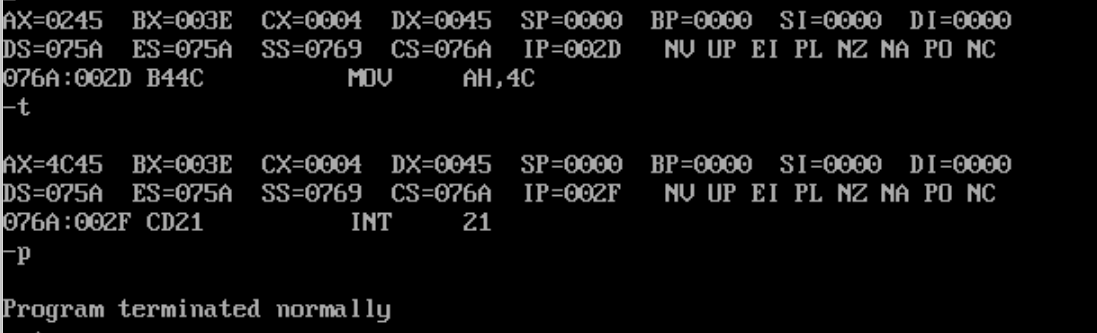
再将(BL)=3EH放入DL，且将DL四进制数高四位置0，可见保留了低位



比较得(DL)中得数大于10（为EH），三四步加37H使之对应’E’得ASCII码



再次利用int21中断程序输出，还是要注意使用-p指令



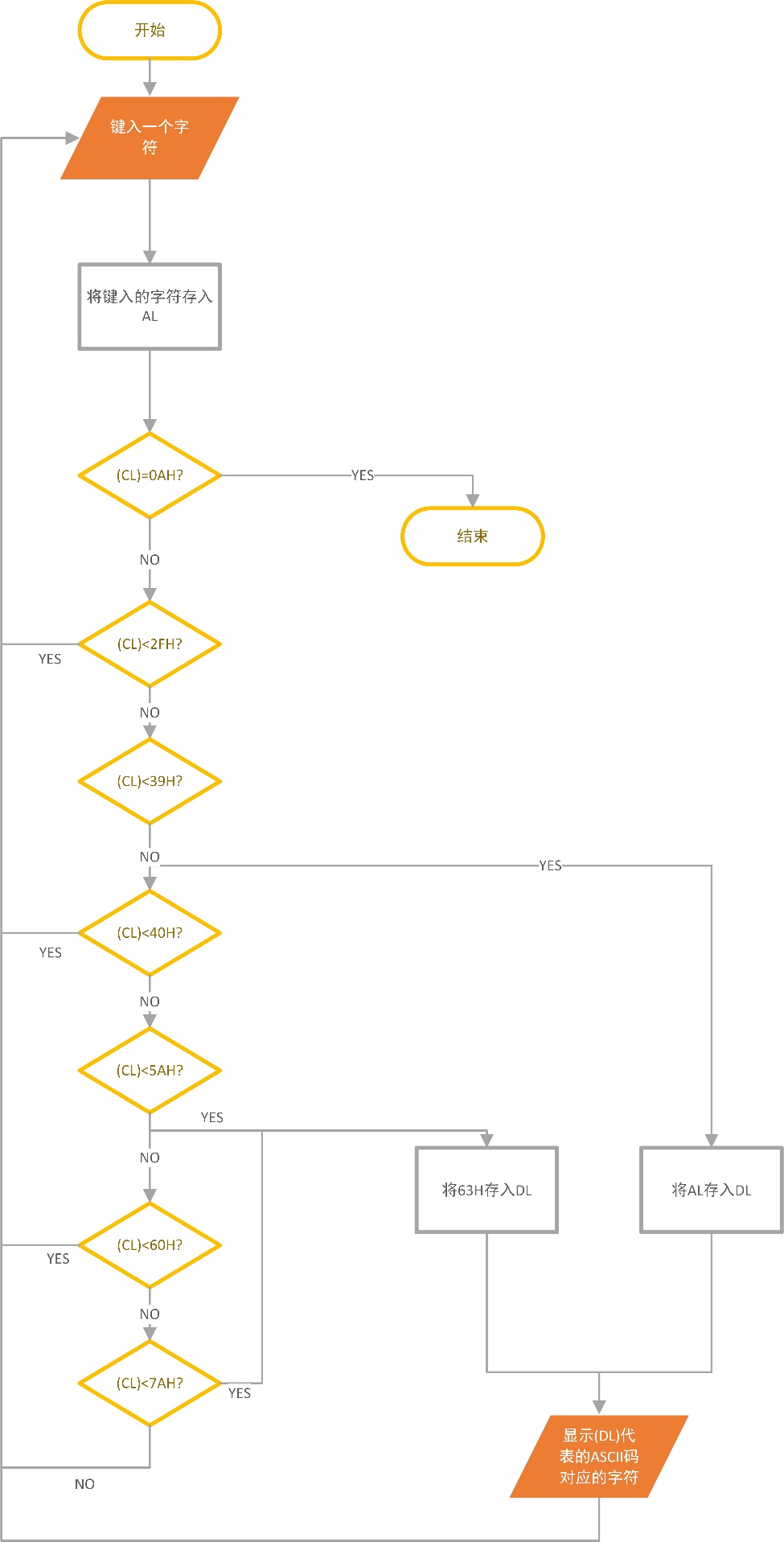
调试完毕，可以直接运行程序查看一下效果



也是符合预期的

【第二个实验】

1.程序流程图：



2.程序源代码

CODE SEGMENT

ASSUME CS:CODE

START:RESTART:

MOV AH,07H

INT 21H

CMP AL,0DH

JE ENDD;回车则结束程序

CMP AL,2FH

JBE RESTART;除去回车，ASCII码在0~47则重新开始

CMP AL,39H

JBE NUM;ASCII码在48~57，进入数字处理块

CMP AL,40H

JBE RESTART;ASCII码在58~64则重新开始

CMP AL,5AH

JBE ALPHA;ASCII码在65~90，进入字母处理块

CMP AL,60H

JBE RESTART;ASCII码在91~96则重新开始

CMP AL,7AH

JBE ALPHA;ASCII码在97~122，进入字母处理块

JMP RESTART;ASCII码在123~126则重新开始

NUM:MOV DL,AL

JMP DISPLAY

;实现对输入数字的显示

ALPHA:MOV DL,63H

JMP DISPLAY

;63H对应'c'的ASCII码，实现输入字母显示'c'

DISPLAY:MOV AH,02H

INT 21H

JMP RESTART

;使用int指令输出(DL)对应字符，输出结束后重新开始

ENDD:MOV AH,4CH

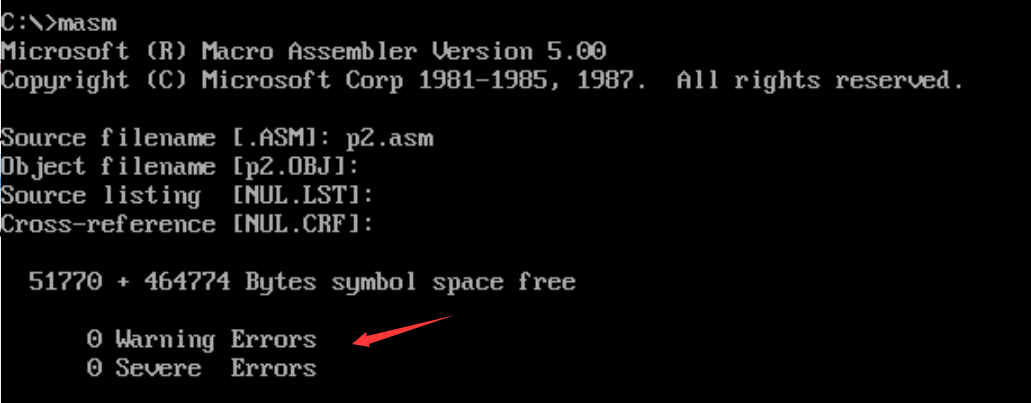
INT 21H

CODE ENDS

END START

3.程序运行验证

（1）对p2.asm进行编译链接





无栈段的提醒是通过常规对程序的检验发出的，可以忽略



可见已生成可执行文件

（2）程序运行验证



经过验证，输入数字直接显示，输入大小写字母均显示为’c’，而输入其他字符，无变化



敲击回车实现退出程序（要注意，CTRL+CR对应的ASCII码为0AH，二者有区别）

【实验心得】

1. CTRL+CR对应的ASCII码为0AH，CR对应的为0DH,二者有区别
2. 对于复杂程序的设计可以使用模块化的思想
3. 对INT21H要使用-p指令执行，-t指令执行后会引发未知的操作
4. 出于对程序的常规检查，masm编译器会对无栈段的代码发出warning，若无需使用栈段则可以忽略
5. 要注意逻辑右移和算术右移的区别，逻辑右移直接在高位补0，算术右移考虑了符号位，为负数时，右移后最高位为1，正数则与逻辑右移无区别
6. INT21程序使用时要注意好发生条件和操作结果

【思考】

1.DS和ES有无存放偏移地址的寄存器？如有得话如何查看？

一、DS为数据段，ES为附加段，在一般情况下二者偏移地址EA可由以下几种方式给出

（1）直接寻址

EA=[idata]

（2）间接寻址

EA=[BX] EA=[SI] EA=[DI]

（3）寄存器相对寻址

EA=[BX+idata] EA=[SI+idata] EA=[DI+idata]

（4）基址变址寻址

EA=[BX+SI] EA=[BX+DI]

（5）相对基址变址寻址

EA=[BX+SI+idata] EA=[BX+DI+idata]

二、涉及中断等汇编指令时，二者有区别

如：在INT21中，一些调用参数由DS:DX决定，并不能通过设置ES为DATA segment，将调用参数改变成由ES:DX决定，除非修改INT21中断程序

2.下面程序中若用JL如何实现？程序复杂度有何区别？

求 AX 中数据的绝对值：

CMP AX, 0

JGE NONEG

NEG AX

NONEG: MOV RESULT, AX

答：用JL实现，应为

CMP AX,0

JL YNEG

JMP S

YNEG:NEG AX

S:MOV RESULT,AX

可见，对负数处理步骤数无区别，对正数需要多一步才得到结果