

实验报告 输入输出

Hollow Man

一、实验环境

一台带有 MASM 软件的装有 Windows XP 系统的实验室计算机。

二、实验准备

用 Win+R 键打开“运行”，输入 cmd 并回车，打开“命令提示符”窗口程序。

在命令行中输入“cd /d D:\”切换到 D 盘根目录。

输入“MD JSL”创建 JSL 工作文件夹。

输入“cd JSL”切换到 JSL 工作目录

输入“copy C:\MASM*.”将程序文件拷贝进工作目录。

三、实验内容

1. 任务 1

将实验材料中所示代码敲入计算机，保存为 2.asm，使用 masm 和 link 进行编译链接，然后直接运行程序，得到以下运行结果：

```
D:\JSL>MASM 2.asm;
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

50746 + 415590 Bytes symbol space free

0 Warning Errors
0 Severe Errors

D:\JSL>LINK 2;

Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

LINK : warning L4021: no stack segment

D:\JSL>2
11223344556677889900$
D:\JSL>
```

使用 debug 中的 t 指令和 p 指令逐步调试：

```

D:\JSL>debug 2.exe
-t
AX=0100 BX=0000 CX=0014 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=13D5 ES=13D5 SS=13E5 CS=13E5 IP=0002  MV UP EI PL NZ NA PO NC
13E5:0002 CD21          INT     21
-p
1
AX=0131 BX=0000 CX=0014 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=13D5 ES=13D5 SS=13E5 CS=13E5 IP=0004  MV UP EI PL NZ NA PO NC
13E5:0004 3C24          CMP     AL,24
-t
AX=0131 BX=0000 CX=0014 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=13D5 ES=13D5 SS=13E5 CS=13E5 IP=0006  MV UP EI PL NZ AC PO NC
13E5:0006 7408          JZ      0010
-t
AX=0131 BX=0000 CX=0014 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=13D5 ES=13D5 SS=13E5 CS=13E5 IP=0008  MV UP EI PL NZ AC PO NC
13E5:0008 8AD0          MOV     DL,AL
-t

```

```

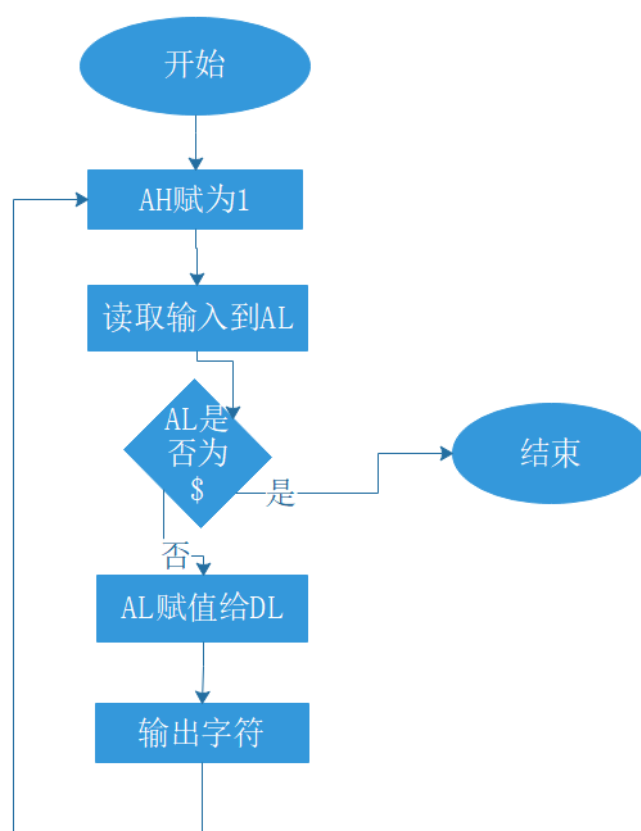
AX=0131 BX=0000 CX=0014 DX=0031 SP=0000 BP=0000 SI=0000 DI=0000
DS=13D5 ES=13D5 SS=13E5 CS=13E5 IP=000A  MV UP EI PL NZ AC PO NC
13E5:000A B402          MOV     AH,02
-t
AX=0231 BX=0000 CX=0014 DX=0031 SP=0000 BP=0000 SI=0000 DI=0000
DS=13D5 ES=13D5 SS=13E5 CS=13E5 IP=000C  MV UP EI PL NZ AC PO NC
13E5:000C CD21          INT     21
-p
1
AX=0231 BX=0000 CX=0014 DX=0031 SP=0000 BP=0000 SI=0000 DI=0000
DS=13D5 ES=13D5 SS=13E5 CS=13E5 IP=000E  MV UP EI PL NZ AC PO NC
13E5:000E EBF0          JMP     0000
-t
AX=0231 BX=0000 CX=0014 DX=0031 SP=0000 BP=0000 SI=0000 DI=0000
DS=13D5 ES=13D5 SS=13E5 CS=13E5 IP=0000  MV UP EI PL NZ AC PO NC
13E5:0000 B401          MOV     AH,01
-t
AX=0131 BX=0000 CX=0014 DX=0031 SP=0000 BP=0000 SI=0000 DI=0000
DS=13D5 ES=13D5 SS=13E5 CS=13E5 IP=0002  MV UP EI PL NZ AC PO NC
13E5:0002 CD21          INT     21
-

```

```
13E5:0002 CD21          INT     21
-p
$
AX=0124  BX=0000  CX=0014  DX=0031  SP=0000  BP=0000  SI=0000  DI=0000
DS=13D5  ES=13D5  SS=13E5  CS=13E5  IP=0004  NU UP EI PL NZ AC PO NC
13E5:0004 3C24          CMP     AL,24
-t
AX=0124  BX=0000  CX=0014  DX=0031  SP=0000  BP=0000  SI=0000  DI=0000
DS=13D5  ES=13D5  SS=13E5  CS=13E5  IP=0006  NU UP EI PL ZR NA PE NC
13E5:0006 7408          JZ      0010
-t
AX=0124  BX=0000  CX=0014  DX=0031  SP=0000  BP=0000  SI=0000  DI=0000
DS=13D5  ES=13D5  SS=13E5  CS=13E5  IP=0010  NU UP EI PL ZR NA PE NC
13E5:0010 B44C          MOV     AH,4C
-t
AX=4C24  BX=0000  CX=0014  DX=0031  SP=0000  BP=0000  SI=0000  DI=0000
DS=13D5  ES=13D5  SS=13E5  CS=13E5  IP=0012  NU UP EI PL ZR NA PE NC
13E5:0012 CD21          INT     21
-p
Program terminated normally
_
```

由图示运行结果可以看到，每次执行完指令后，CS:IP 自动指向了下一个命令地址，并且在每次 INT 21 调用完 DOS 输入一个非\$字符后，再次调用 INT 21 时会将这个字符输出。当输入\$字符时，再次调用 INT 21 时程序结束。

查看源代码，画出流程图：



由图示流程图可知，程序在循环中首先将 AH 赋值为 1，然后读取输入的字符赋值给 AL，随后进行判断 AL 是否为 '\$'，如果是则退出程序，不是则将 AL 赋值给 DL，调用 DOS

输出。

另外按照实验课上的题目要求，将自己编写的实现输出换行的改进程序保存为 2m.asm，使用 masm 和 link 进行编译链接，然后直接运行程序，得到以下运行结果：

1
1
2
2
3
3
4
4
5
5
6
6
\$

其代码如下：

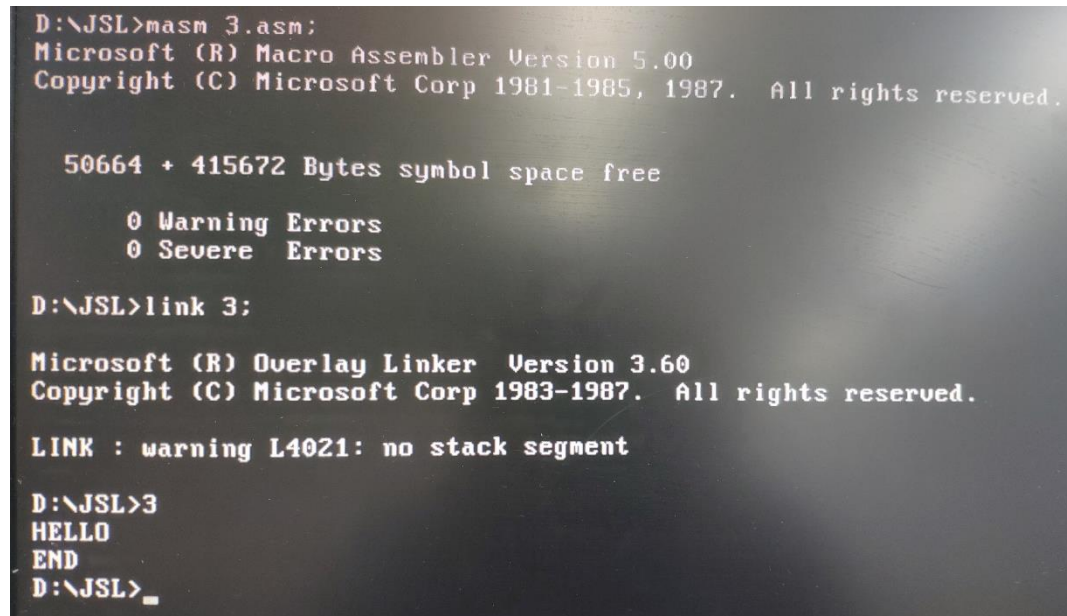
```
code segment
assume cs:code
G1:
MOV AH,1
INT 21H
CMP AL,'$'
JZ EXIT
MOV BX,2000H
MOV SS,BX
MOV SP,10H
ADD SP,6
MOV BH,1
MOV BL,0AH
PUSH BX
MOV BL,0DH
PUSH BX
PUSH AX
POP AX
MOV DL,AL
MOV AH,2
INT 21H
POP AX
MOV DL,AL
MOV AH,2
INT 21H
POP AX
MOV DL,AL
```

```
MOV AH,2
INT 21H
JMP G1
EXIT:
MOV AH,4CH
INT 21H
code ends
end G1
```

该程序通过入栈和出栈操作，在源程序的基础上，首先让换行符（0A）和回车符（0D）依次入栈，然后让输入得到的字符入栈，随后再次让换行符（0A）和回车符（0D）依次入栈，此时执行出栈操作，将其依次赋值给 DL，调用 DOS 进行输出，就实现了如上功能。

2. 任务 2

将实验材料中所示代码敲入计算机，保存为 3.asm，使用 masm 和 link 进行编译链接，然后直接运行程序，得到以下运行结果：



```
D:\JSL>masm 3.asm;
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

50664 + 415672 Bytes symbol space free

0 Warning Errors
0 Severe Errors

D:\JSL>link 3;

Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

LINK : warning L4021: no stack segment

D:\JSL>3
HELLO
END
D:\JSL>_
```

使用 debug 中的 t 指令和 p 指令逐步调试：


```

AX=13E5 BX=0000 CX=00A8 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=13D5 ES=13D5 SS=13E5 CS=13ED IP=0003 MV UP EI PL NZ NA PO NC
13ED:0003 8ED8 MOV DS,AX
-t

AX=13E5 BX=0000 CX=00A8 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=13E5 ES=13D5 SS=13E5 CS=13ED IP=0005 NV UP EI PL NZ NA PO NC
13ED:0005 8D1E0000 LEA BX,[0000] DS:0000=6548
-t

AX=13E5 BX=0000 CX=00A8 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=13E5 ES=13D5 SS=13E5 CS=13ED IP=0009 NV UP EI PL NZ NA PO NC
13ED:0009 8A17 MOV DL,[BX1] DS:0000=48
-t

AX=13E5 BX=0000 CX=00A8 DX=0048 SP=0000 BP=0000 SI=0000 DI=0000
DS=13E5 ES=13D5 SS=13E5 CS=13ED IP=000B NV UP EI PL NZ NA PO NC
13ED:000B 80FA24 CMP DL,24
-t

AX=13E5 BX=0000 CX=00A8 DX=0048 SP=0000 BP=0000 SI=0000 DI=0000
DS=13E5 ES=13D5 SS=13E5 CS=13ED IP=000E NV UP EI PL NZ NA PE NC
13ED:000E 7414 JZ 0024
-

```

```

AX=13E5 BX=0000 CX=00A8 DX=0048 SP=0000 BP=0000 SI=0000 DI=0000
DS=13E5 ES=13D5 SS=13E5 CS=13ED IP=0010 NV UP EI PL NZ NA PE NC
13ED:0010 80FA61 CMP DL,61
-t

AX=13E5 BX=0000 CX=00A8 DX=0048 SP=0000 BP=0000 SI=0000 DI=0000
DS=13E5 ES=13D5 SS=13E5 CS=13ED IP=0013 NV UP EI NG NZ NA PE CY
13ED:0013 7208 JB 001D
-t

AX=13E5 BX=0000 CX=00A8 DX=0048 SP=0000 BP=0000 SI=0000 DI=0000
DS=13E5 ES=13D5 SS=13E5 CS=13ED IP=001D NV UP EI NG NZ NA PE CY
13ED:001D B402 MOV AH,02
-t

AX=02E5 BX=0000 CX=00A8 DX=0048 SP=0000 BP=0000 SI=0000 DI=0000
DS=13E5 ES=13D5 SS=13E5 CS=13ED IP=001F NV UP EI NG NZ NA PE CY
13ED:001F CD21 INT 21
-p
H
AX=0248 BX=0000 CX=00A8 DX=0048 SP=0000 BP=0000 SI=0000 DI=0000
DS=13E5 ES=13D5 SS=13E5 CS=13ED IP=0021 NV UP EI NG NZ NA PE CY
13ED:0021 43 INC BX
-

```

```

AX=0248 BX=0001 CX=00A8 DX=004B SP=0000 BP=0000 SI=0000 DI=0000
DS=13E5 ES=13D5 SS=13E5 CS=13ED IP=0022  NU UP EI PL NZ NA PO CY
13ED:0022 EBE5          JMP     0009
-t

AX=0248 BX=0001 CX=00A8 DX=004B SP=0000 BP=0000 SI=0000 DI=0000
DS=13E5 ES=13D5 SS=13E5 CS=13ED IP=0009  NU UP EI PL NZ NA PO CY
13ED:0009 8A17          MOV     DL,[BX]          DS:0001=65
-t

AX=0248 BX=0001 CX=00A8 DX=0065 SP=0000 BP=0000 SI=0000 DI=0000
DS=13E5 ES=13D5 SS=13E5 CS=13ED IP=000B  NU UP EI PL NZ NA PO CY
13ED:000B 80FA24       CMP     DL,24
-t

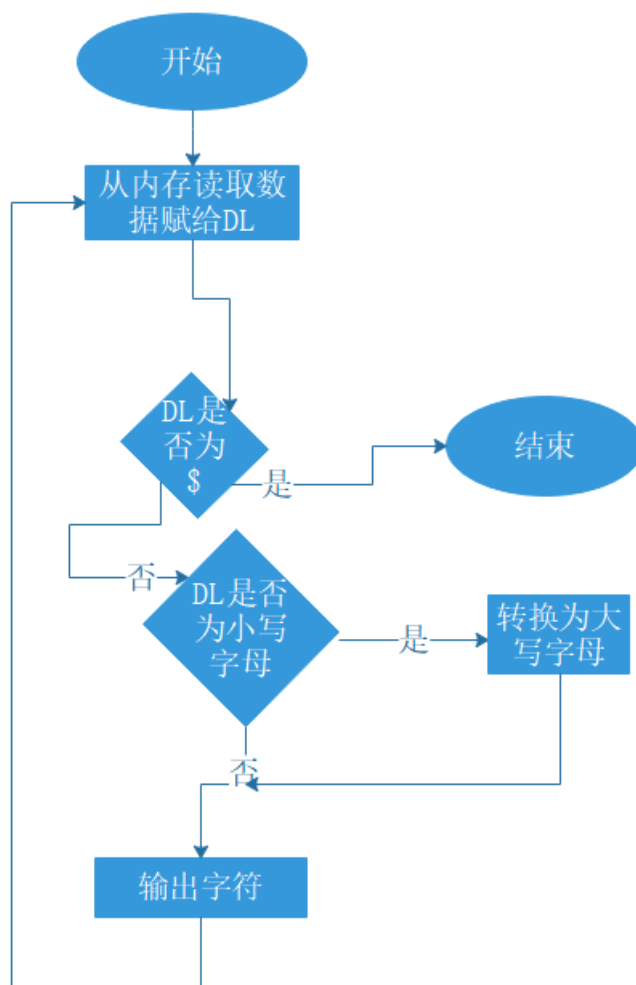
AX=0248 BX=0001 CX=00A8 DX=0065 SP=0000 BP=0000 SI=0000 DI=0000
DS=13E5 ES=13D5 SS=13E5 CS=13ED IP=000E  NU UP EI PL NZ NA PE NC
13ED:000E 7414          JZ      0024
-t

AX=0248 BX=0001 CX=00A8 DX=0065 SP=0000 BP=0000 SI=0000 DI=0000
DS=13E5 ES=13D5 SS=13E5 CS=13ED IP=0010  NU UP EI PL NZ NA PE NC
13ED:0010 80FA61       CMP     DL,61

```

由图示运行结果可以看到，每次执行完指令后，CS:IP 自动指向了下一个命令地址，并且每次调用 INT 21 后输出一个大写字母直到结束。

查看源代码，画出流程图：



首先，从内存中读取已经写入内存的字符赋值给 DL，然后判断 DL 是否为 '\$'，如果是

则退出程序，不是则判断是否为小写字母，是则转换为大写字母输出，不是则直接输出，然后继续读取已经写入内存的下一个字符

四、实验总结

通过这次实验课，我了解到了 DOS 调用输入输出的编程特点，能够熟练地做到编写类似的程序，收获颇丰。