
微机原理与系统设计实验报告

20009200077_李傲

一、实验要求

编写程序实现下列 5 项功能，通过从键盘输入 1~5 进行菜单式选择。

(1) 按数字键“1”，完成将字符串中的小写字母变换成大写字母。用户输入由英文大小写字母或数字 0~9 组成的字符串（以回车结束），变换后按下列格式在屏幕上显示：

<原字符串>: abcdgyt0092

<新字符串>: ABCDGYT0092

按任一键重做；按 Esc 键返回主菜单

(2) 按数字键“2”，完成在字符中找最大值。用户输入由英文大小写字母或数字 0~9 组成的字符串（以回车结束），找出最大值后按下列格式在屏幕上显示：

<原字符串> The maximum is <最大值>

按任一键重做；按 Esc 键返回主菜单。

(3) 按数字键“3”，完成输入数据组的排序。用户输入一组十进制数值（小于 255），然后变换成十六进制数，并按递增方式进行排序，按下列格式在屏幕上显示：

<原数值串>

<新数值串>

按任一键重做；按 Esc 键返回主菜单。

(4) 按数字键“4”，完成时间的显示。在屏幕的右上角实时显示出时间：HH:MM:SS。

按任一键重新对时；按 Esc 键返回主菜单。

(5) 按数字键“5”，结束程序的运行，返回操作系统。

二、实验环境

操作系统：Windows10

软件环境：emu8086

CPU：i5-10300H

RAM：16GB

三、解决方法

3.1 实验总体设计

3.1.1 工作流程

流程如下图所示：

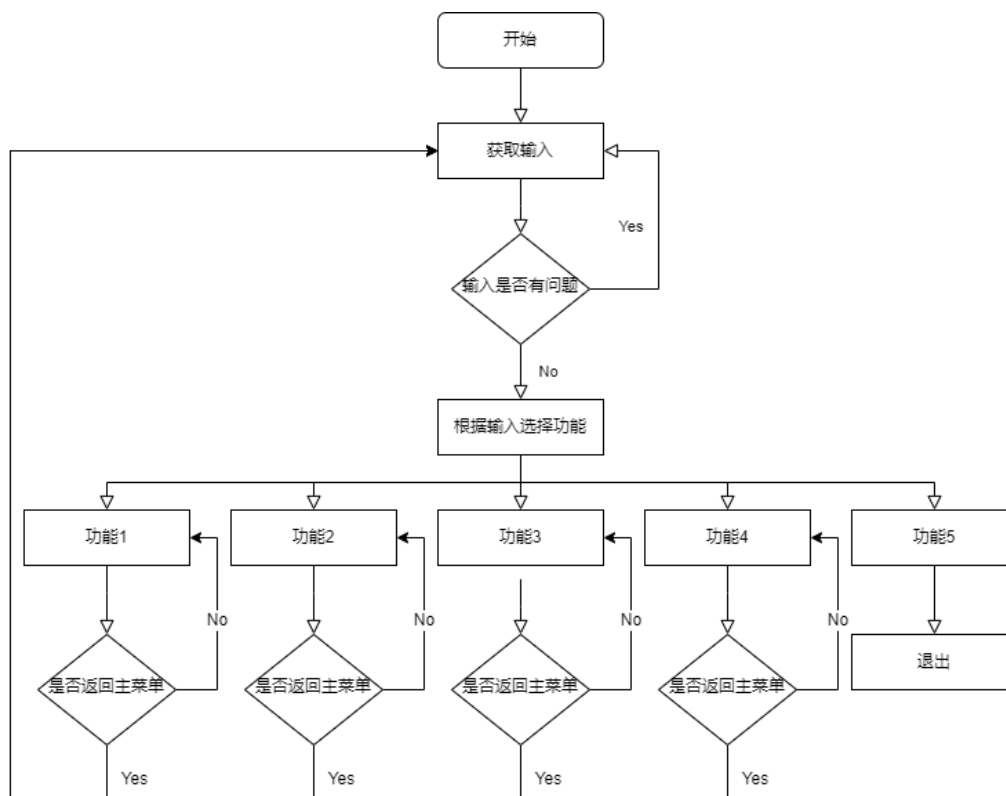


图 1：总体的流程

3.1.2 定义全局函数

有一些功能在各个功能中多次被调用，因此将这些功能封装成宏定义函数，便于调用。

1、获取输入

其功能是获取单个输入字符并存储在 AL 中，与 C 语言中的 `getchar()` 类似，因此将其命名为 `getchar`，相关代码如下：

```
1. getchar MACRO
2.     MOV AH,1
3.     INT 21H
4. ENDM
```

2、输出字符串

该函数的传入参数为 `string`，通过 DOS 的中断调用，可以在屏幕上显示出 DS: DX 地址处的字符串，功能与 C 语言中的 `puts()` 函数类似，因此将其命名为 `puts`，相关代码如下：

```
1. puts MACRO string
2.     LEA DX,string
3.     MOV AH,09H
4.     INT 21H
5. ENDM
```

3、换行显示问题

其功能是显示回车符和换行符的方式实现换行这一效果，功能与 C 语言中的 “\n” 类似，因此将其命名为 `\n`，相关代码如下：

```
1. \n MACRO
2.     MOV AH,2
3.     MOV DL,0DH
4.     INT 21h
5.     MOV AH,2
6.     MOV DL,0AH
7.     INT 21h
8. ENDM
```

3.1.3 定义数据段

某些字符串在运行中多次被调用，因此将其定义在数据段，并在此处定义各功能中所需的变量，相关代码如下：

```
1. DATA SEGMENT
2.     ;定义字符串
3.     welcome DB 'plase input the function number (1~5): $'
4.     scanf DB 'please input character: ','$'
5.     catchError DB 'Error: Invaild Syntax$'
```

```

6.      return0 DB 'What do you want to do next" Main Menu or Redo" [
ESC/any other key]: $'
7.
8.      resultTask2info DB 'The maximum is: $'
9.      scanf3 DB 'please input the decimal numbers: ',0DH,0AH,'$'
10.
11.      Task4info DB 'press anykey to display the time$'
12.
13.      info1 DB 'Now, we are doing function 1: $'
14.      info2 DB 'Now, we are doing function 2: $'
15.      info3 DB 'Now, we are doing function 3: $'
16.
17.      ;初始化变量
18.      inputTask1 DB 100 dup(0)
19.      resultTask1 DB 100 dup(0)
20.
21.      inputTask2 DB 100 dup(0)
22.      resultTask2 DB 10 dup(0)
23.
24.      inputTask3 DB 100 dup(0)
25.      inputTask3init DB 100 dup(0)
26.      countTask3 DB 10 dup(0)
27.      resultTask3 DB 100 dup(0)
28.
29.      hour DB 0
30.      minute DB 0
31.      second DB 0
32.      time DB "00:00:00$"
33.      len equ $-time
34.      num DB 10
35. DATA ENDS

```

3.1.4 捕获异常

在每次获取输入后，都需要对输入数据进行检测，若输入异常值则需要提示错误信息并使重新输入。

例如，在主菜单中，输入只能为 1-5，输入其他值会报错，该部分可由下面的代码实现：

```

1.  main:
2.      puts welcome
3.      getchar
4.      ;转入各 function/非法输入
5.      CMP AL,31H

```

```

6.      JB invalid    ;比 1 小
7.      JE function1
8.
9.      CMP AL,32H
10.     JE function2
11.
12.     CMP AL,33H
13.     JE function3
14.
15.     CMP AL,34H
16.     JE function4
17.
18.     CMP AL,35H
19.     JE function5
20.     JA invalid    ;比 5 大
21.
22.     ;非法输入
23. invalid:
24.     \n
25.     puts catchError
26.     \n
27.     \n
28.     JMP main

```

使用 CMP 指令，将输入字符与目标字符的 ASCII 码进行比较。其中 31H 为数字“1”的 ASCII 码，35h 为数字“5”的 ASCII 码。若输入字符的 ASCII 码小于 31H 或大于 35H，则跳转到错误提示代码块 invalid，否则进入各功能。

运行截图：

The screenshot shows a window titled "emulator screen (80x25 chars)". The text inside the window is as follows:

```

please input the function number (1~5): x
Error: Invalid Input
please input the function number (1~5):

```

At the bottom of the window, there are buttons for "clear screen" and "change font", and a status bar showing "JMP main".

3.2 功能一：将字符串中的小写字母变换成大写字母

考虑到字符在计算机中是以 ASCII 码的方式进行存储，而大小写字母之间的 ASCII 码相差 20H，因此，只需将所有小写字母的 ASCII 码减去 20H，即可转换成大写字母。

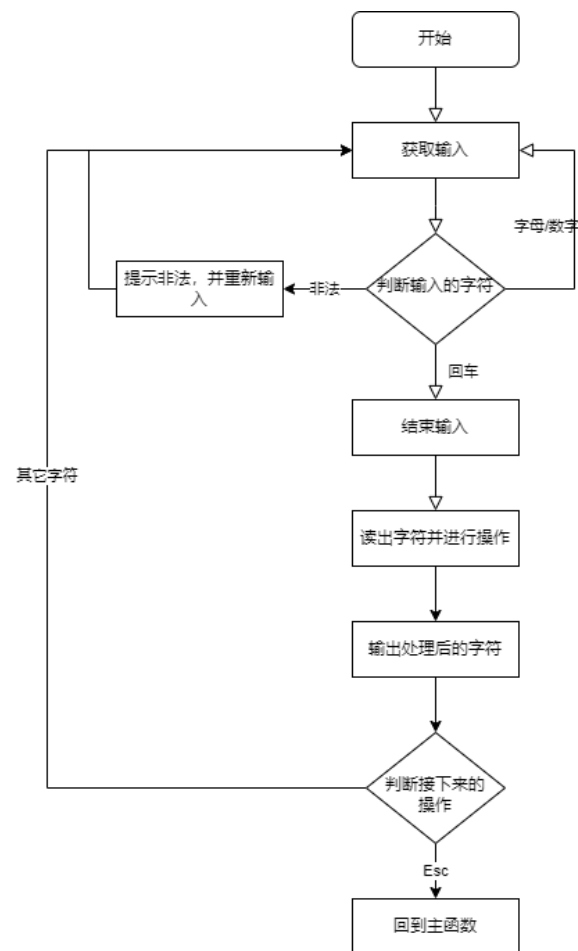


图 2：功能 1 的流程图

```
emulator screen (80x25 chars)
plase input the function number (1~5): 1
Now, we are doing function 1:
please input character: xidian123
xidian123
XIDIAN123
What do you want to do next" Main Menu or Redo" [ESC/any other key]:
Now, we are doing function 1:
please input character: {
Error: Invaild Input
please input character: xidian123
xidian123
XIDIAN123
What do you want to do next" Main Menu or Redo" [ESC/any other key]: +
plase input the function number (1~5): _
```

图 3：功能 1 的运行截图

3.3 功能二：在字符串中找最大值

本程序采用遍历的思路进行比较。在获取每个字符之后，通过源变址寄存器 SI 对原字符串进行遍历，比较每个字符的 ASCII 码，初始化最大值为第一个字符，若后面的存在字符比当前最大值大，则最大值替换为该字符。最后，对源字符串和目标字符串结尾加入字符串结束标记“\$”，调用 puts 函数进行输出。

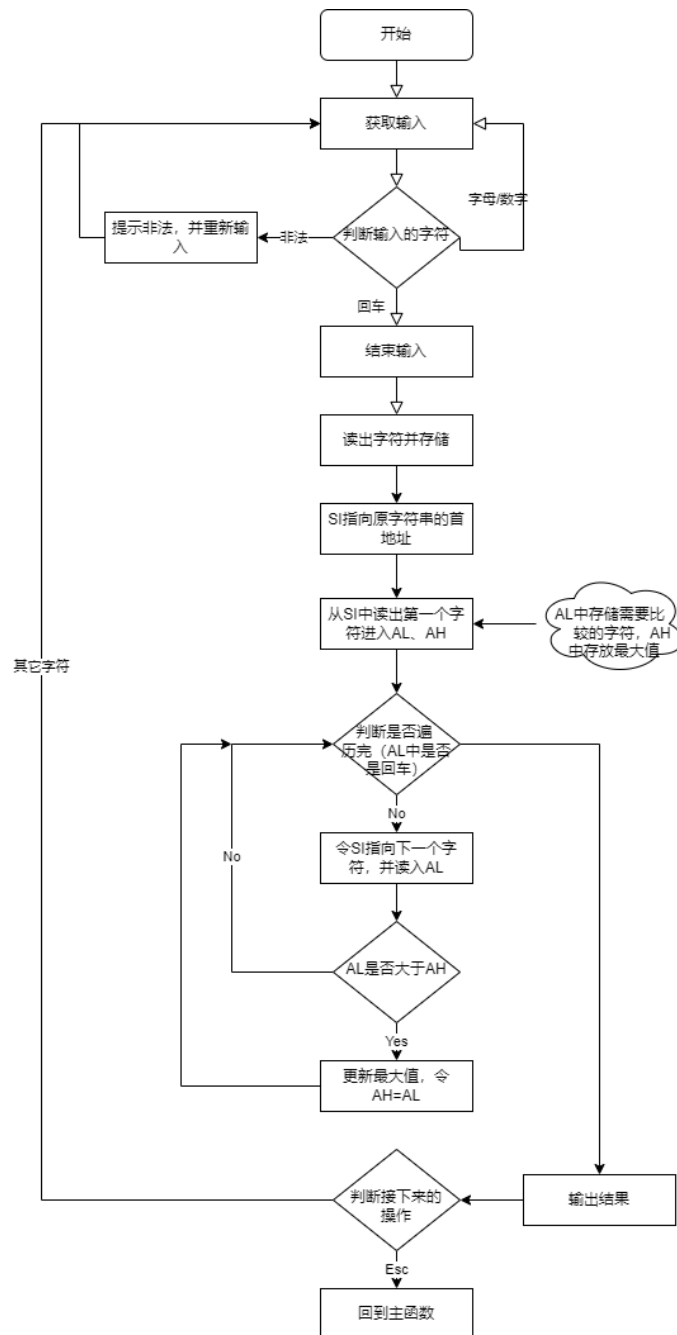


图 4：功能 2 的流程图

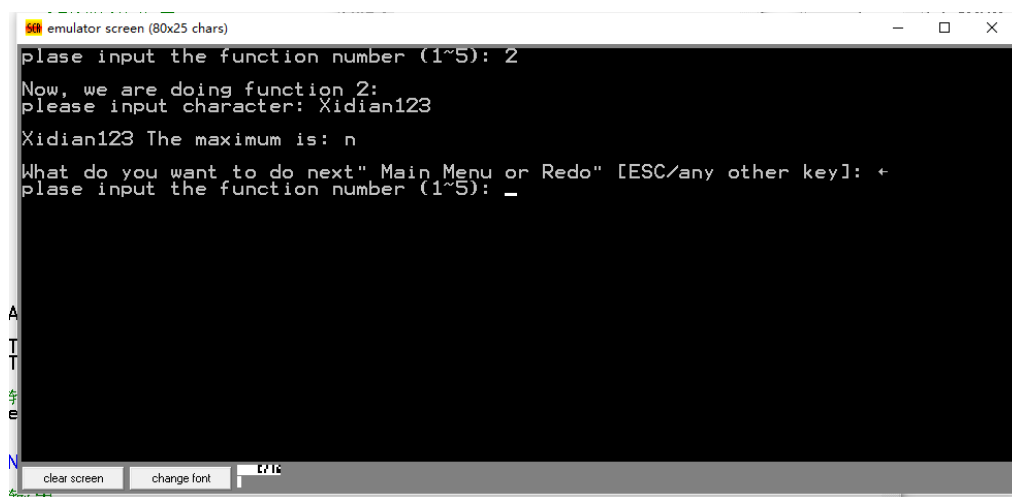


图 5：功能 2 的运行截图

3.4 问题三：输入数据组的排序

3.4.1 输入的 ASCII 码转 16 进制数

输入的数字在计算机内部存储为 ASCII 码，寄存器是以十六进制来进行数据的存储，因此我们要将 ASCII 码转换成 16 进制数。

过程中我选择使用 10 进制数作为转换的媒介，具体操作可以将 0-9 的 ASCII 码的高位清零，高位清零操作可以用当前数据和 0FH 取逻辑与来实现。当数字仅有一位时，十六进制和十进制完全一样，当存在下一位数据时，之间的数据*10 再加上下一位数据，这样就实现了十进制数向十六进制数的转换。

3.4.2 输入数据的递增排序

本程序选择了选择排序算法来对输入数据进行排序，相关代码如下：

```
1. ;选择排序
2. sort:
3.     LEA     SI,countTask3
4.     MOV     CL,[SI]
5.     LEA     SI,inputTask3
6.
7. ;外循环
8. fori:
9.     LEA     DI,countTask3
10.    MOV     CH,CL
11.    MOV     DI,SI
12.    INC     DI
13. ;内循环
14. forj:
```


15.	MOV	BL,[SI]
16.	MOV	BH,[DI]
17.	CMP	BL,BH
18.	JBE	afterswap
19.	MOV	[SI],BH
20.	MOV	[DI],BL

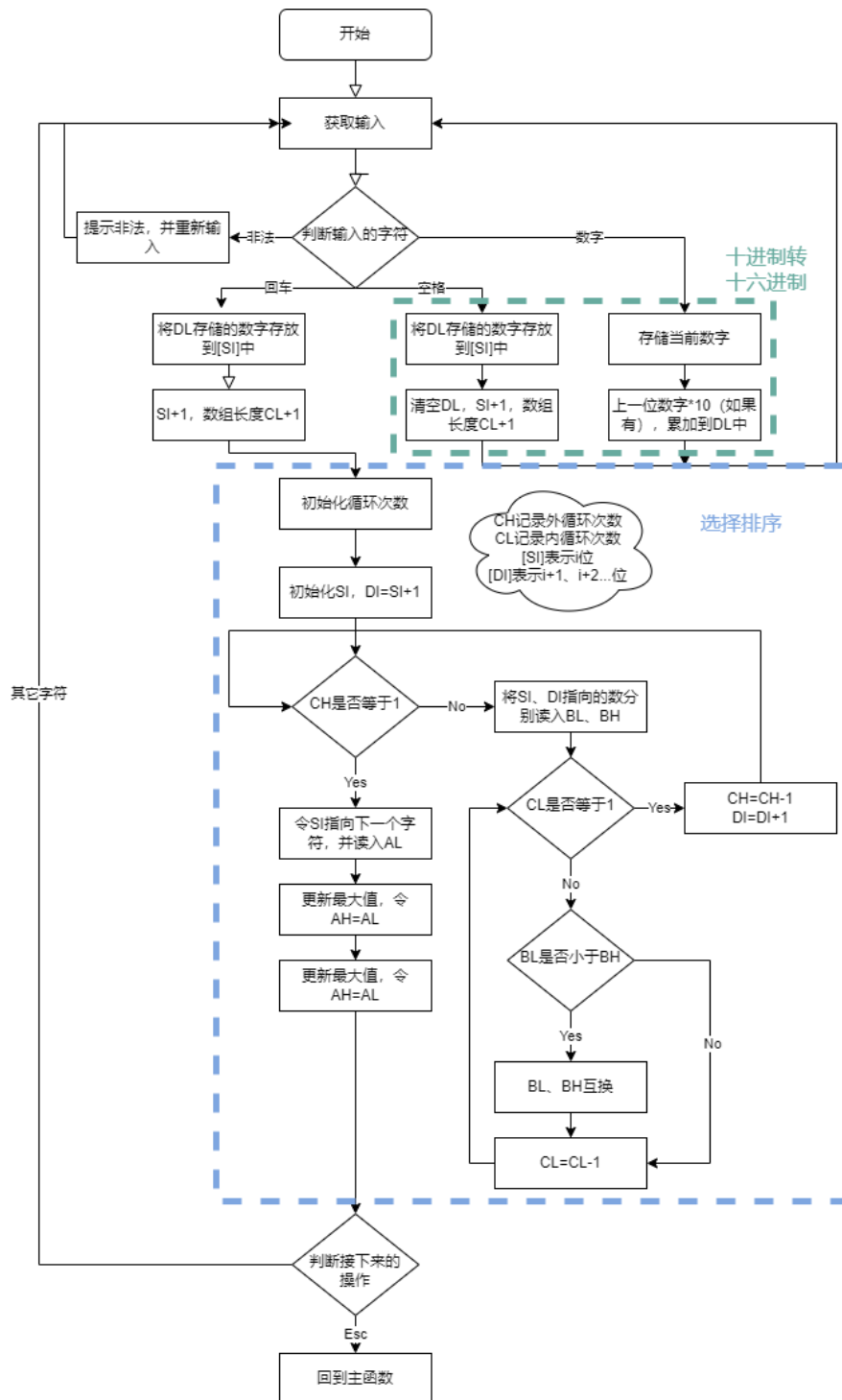
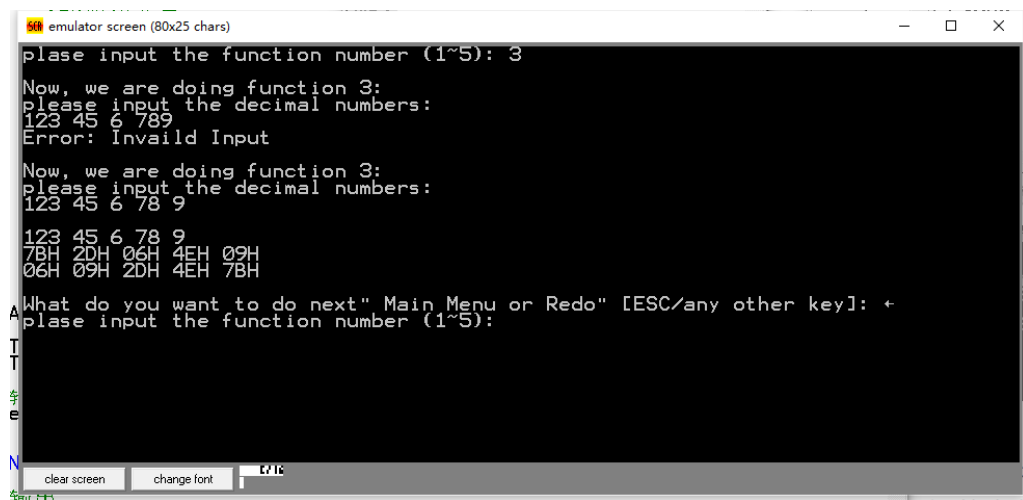


图 6：功能 3 的流程图

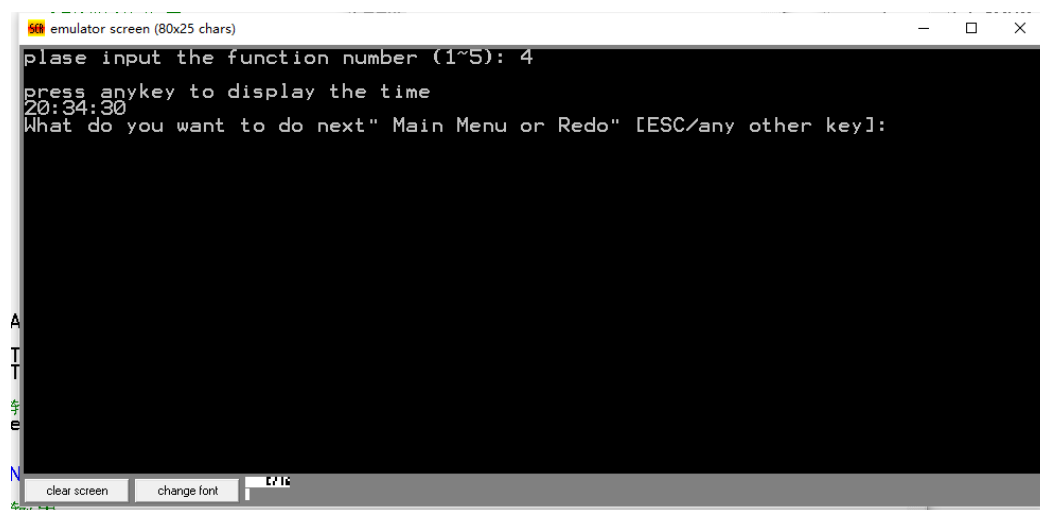


```
emulator screen (80x25 chars)
plase input the function number (1~5): 3
Now, we are doing function 3:
please input the decimal numbers:
123 45 6 789
Error: Invaild Input
Now, we are doing function 3:
please input the decimal numbers:
123 45 6 78 9
123 45 6 78 9
7BH 2DH 06H 4EH 09H
06H 09H 2DH 4EH 7BH
What do you want to do next" Main Menu or Redo" [ESC/any other key]: +
please input the function number (1~5):
```

图 7：功能 3 的运行截图

3.5 功能四：显示系统时间

本程序使用 DOS 中断 INT 21H 的 2CH 功能来获取当前系统时间，时、分、秒的值并存放在 CH、CL、DH 中，之后将其放入字符串中输出即可。



```
emulator screen (80x25 chars)
plase input the function number (1~5): 4
press anykey to display the time
20:34:30
What do you want to do next" Main Menu or Redo" [ESC/any other key]:
```

图 8：功能 4 的运行截图

3.6 功能五：结束程序的运行

本程序使用 DOS 中断 INT 21H 的 4CH 号功能来实现结束程序运行。

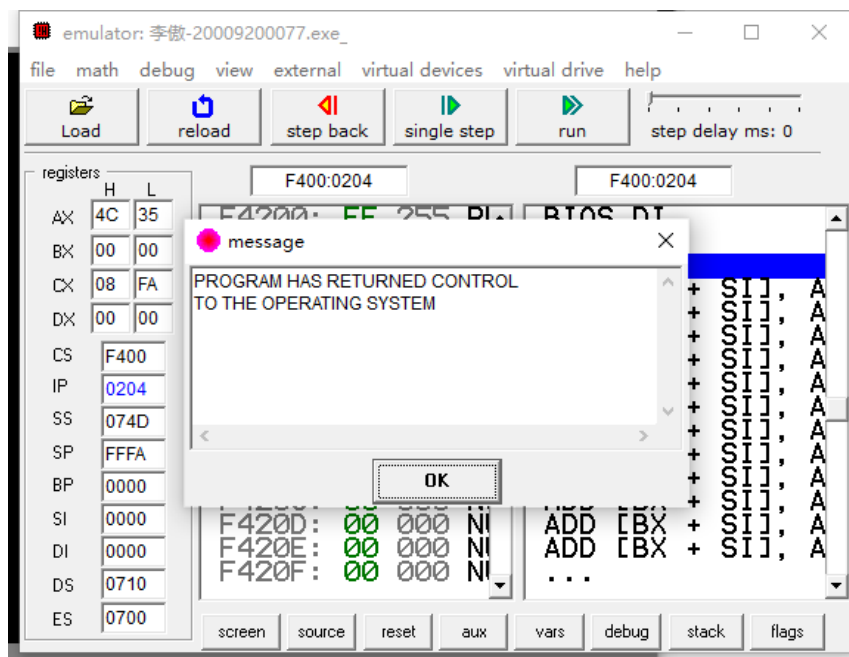


图 9：功能 5 的运行截图

四、总结

4.1 优点

- 1、封装了一些多次复用的功能，使程序整体更加简洁
- 2、函数的命名接近 C 语言，功能明了
- 3、程序段的命名均基于 camelCase 准则

4.2 缺点

- 1、对错误的捕获不够全面，只能捕获输入错误
- 3、程序仍然较长，还有简化的空间
- 2、未进行变量保护操作

4.3 收获

- 1、全面复习了微原课中所学习的汇编知识
- 2、将所学知识应用于实践中，加深了我对汇编语言的理解