

# 微机原理与接口技术

## 汇编语言上机大作业



姓名： 付凯文

学号： 19170100004

学院： 人工智能学院

日期： 2021/11/25

### 目录

1. 实验要求.....	3
2. 实验环境.....	3
3. 程序设计.....	3
3.1 程序总体结构设计.....	3
3.2 主程序 main 设计.....	4
3.3 子程序 menu 设计.....	5
3.4 子程序 changestr 设计.....	6
3.5 子程序 maxnumber 设计.....	7
3.6 子程序 sortnumber 设计.....	8
3.7 子程序 showtime 设计.....	12
4. 实验结果.....	14
5. 实验总结.....	17
6. 附录.....	17

### 3 微机原理实验 付凯文

## 1. 实验要求

编写程序实现下列五项功能，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  
按任意键重做；按 Esc 键返回主菜单
2. 按数字键“2”，完成在字符串中找最大值。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>.  
按任意键重做；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>  
按任意键重做；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。  
按任意键重做；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

## 2. 实验环境

1. 系统环境：windows10
2. 软件环境：emu8086, Vscode

## 3. 程序设计

### 3.1 程序总体结构设计

根据题目要求，程序共有五个功能，我们可以考虑采用跳转的方式来实现多分支结构程序设计。对问题进行分析，共需要设计五个子程序，分别是菜单子程序 menu，大小写转换子程序 changestr，寻找最大值子程序 maxnumber，数字排序程序子程序 sortnumber 和时间显示子程序 showtime

主程序 main 首先调用子程序 menu，设置显示器并打印提示信息，然后读入‘1’-‘5’之间的 ASCII 表示数，之后将读入的 ASCII 码表示数分别与‘1’、‘2’、‘3’、‘4’、‘5’进行比较，若相等则跳转到对应数字表示的功能。

当输入‘1’时，跳转到标号 opt1，然后调用子程序 changestr，完成将输入字符串中小写字母变为大写字母的功能。之后检测用户输入，若输入 ESC 则跳转到主程序首部调用 menu，若输入其他字符则跳转到 opt1。

## 4 微机原理实验 付凯文

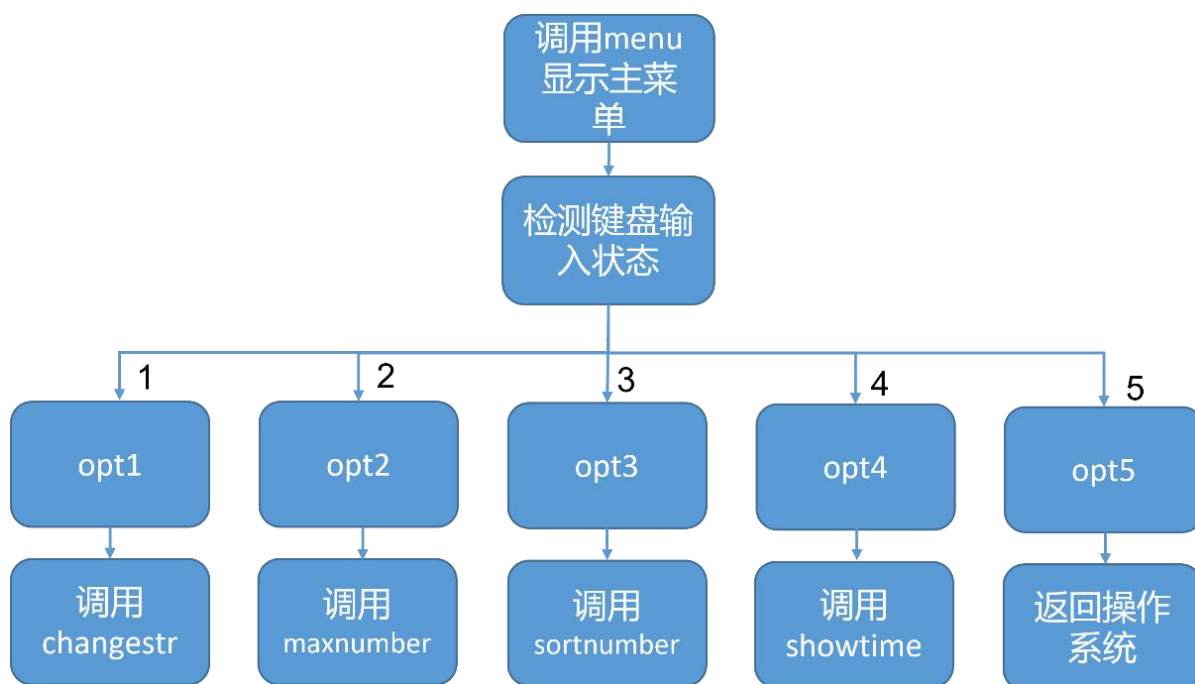
当输入'2'时, 跳转到标号 opt2, 然后调用子程序 maxnumber, 找出输入字符串中 ASCII 码值最大的字符。之后检测用户输入, 若输入 ESC 则跳转到主程序首部调用 menu, 若输入其他字符则跳转到 opt2.

当输入'3'时, 跳转到标号 opt3, 然后调用子程序 sortnumber, 将输入的数字进行排序后以 16 进制形式输出。之后检测用户输入, 若输入 ESC 则跳转到主程序首部调用 menu, 若输入其他字符则跳转到 opt3.

当输入'4'时, 跳转到标号 opt4, 然后调用子程序 showtime, 将系统时间显示在屏幕右上角并实时刷新所显示的时间。之后检测用户输入, 若输入 ESC 则跳转到主程序首部调用 menu, 若输入其他字符则跳转到 opt4.

当输入'5'时, 跳转到标号 opt5, 通过调用系统指令 int 21h((ah=4ch))结束程序的运行, 返回参作系统。

程序总体设计思路如图所示:



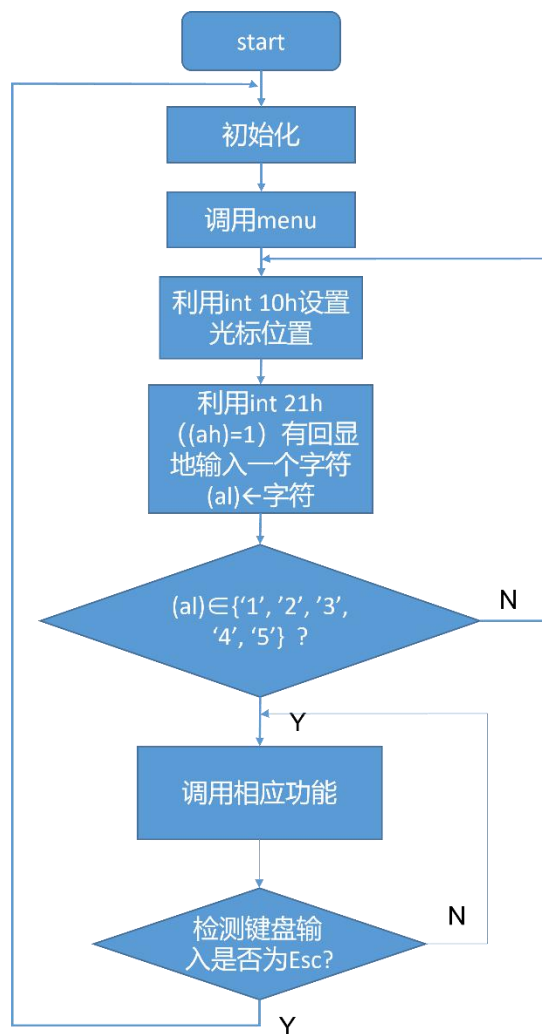
:

## 3.2 主程序 main 设计

对于主函数 main, 具体设计如下。

1. 对系统进行初始化
2. 调用子程序 menu 打印提示信息
3. 利用 int10h 指令设置字符输入时的光标位置
4. 利用 int 21h ((ah)=1) 指令有回显地输入一个字符
5. 对字符进行判断, 若字符的 ASCII 码小于'1'的 ASCII 码或大于'5'的 ASCII 码, 则证明输入有错误, 程序跳转到 menu 之后, 重新输入字符。若输入没有错误, 则将输入的字符分别与'1','2','3','4','5'进行比对, 若相等, 则跳转到相应的标号处并调用对应的子程序。

程序流程库框图如下:



### 3.3 子程序 menu 设计

#### 3.3.1 子程序 menu 设计思路

子程序 menu 的功能较为简单，只需显示打印出提示信息即可。设计的大致思路为：首先调用系统程序设置显示器，之后依次设置输出提示信息时的光标位置并打印输出信息。我们调用 int 21h((ah)=9)打印输出信息，需要注意的是，使用该指令打印字符串时，字符串末尾需要有结束标志 '\$'，否则打印会出错。

#### 3.3.2 子程序 menu 具体设计

对于菜单程序 menu，具体设计如下：

入口参数：无

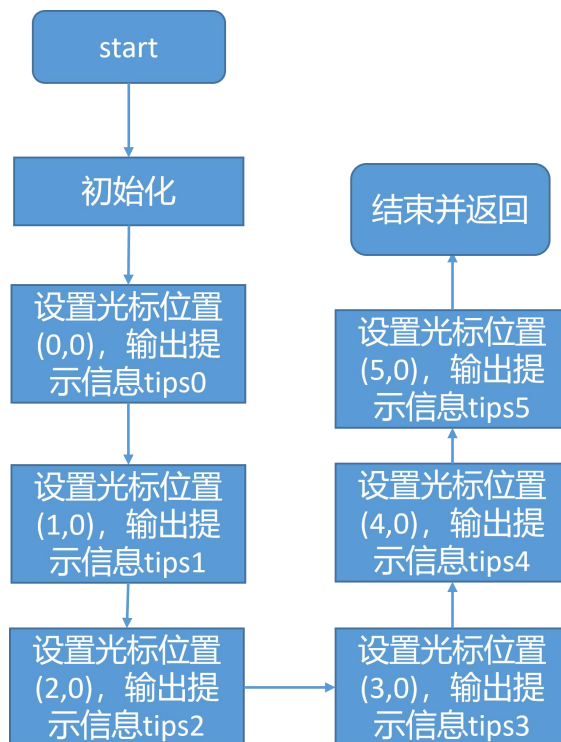
出口参数：无

1. 利用 int 10h((ah=0)) 调用显示器，取(al)=3，设置显示方式为 80\*25，16 色文本方式

## 6 微机原理实验 付凯文

2. 依次设置光标位置，输出提示信息 tips0, tips1, tips2, tips3, tips4, tips5

子程序 menu 的流程框图如下：



## 3.4 子程序 changestr 设计

### 3.4.1 子程序 changestr 设计思路

对于该程序，需要能够将输入字符串中的小写字母转化为大写字母，其他字符保持不变，设计的大致思路为：首先调用系统程序将字符串输入缓冲区，并对缓存区的字符进行存储。在这里我们定义存储区 `keybuf db129 db0 db 61 dup(0)`，第一位 129 表示存储器的容量为 129，第二位用于存储字符串的长度，其余空间用于存储字符串。之后遍历整个字符串，若字符为小写字母，则将其转化为大写字母，否则，字符保持不变。

小写字母转化为大写字母，我们可以使小写字母对应的 ASCII 码值减去 32，从而实现大小写的转化，通过查阅相关资料得知，可以通过按位与 `0dfh` 的方式将小写字母转化为大写字母，此种方法有着更快的运算速度。

### 3.4.2 子程序 changestr 具体设计

对于大小写转换程序，具体设计如下：

入口参数：无

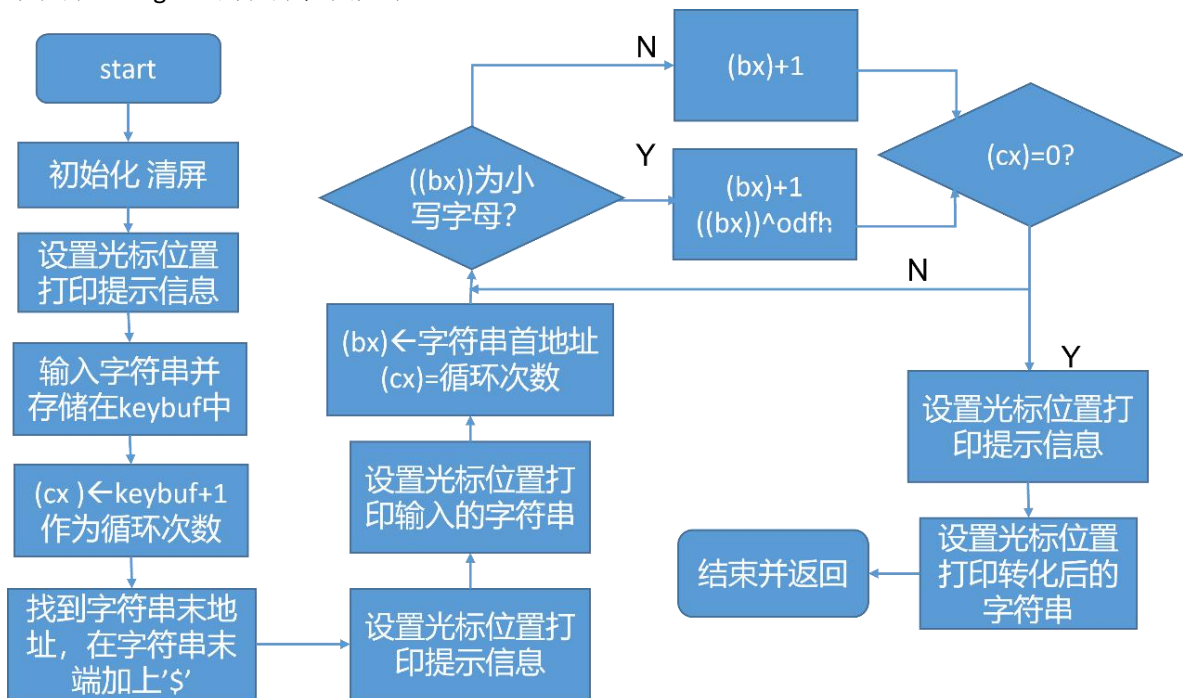
出口参数：无

1. 利用 `int 10h((ah=0))` 调用显示器，取 `(al)=3`，设置显示方式为 80\*25，16 色文本方式，同时完成清屏

## 7 微机原理实验 付凯文

2. 利用 `int 10h((ah)=2)` 设置光标位置并利用 `int 21h((ah)=9)` 打印提示信息
3. 调用 `int 21h((ah)=0ah)` 通过键盘将字符串有回显地输入到缓存区
4. 将 `keybuf+1` 的值传递给 `cx`, 作为后续循环的次数
5. 找到字符串的末地址, 在原字符串末尾加上终止标志 '\$'
6. 设置光标位置, 打印提示信息和输入的字符串。
7. 依次将字符串中的字符转移到 `bx` 中, 对 `bx` 中的字符进行判断, 若不是小写字母不做处理, 若是小写字母则与上 `0dfh` 变为大写字母
8. 设置光标位置, 打印提示信息
9. 设置光标位置, 打印转换后的字符串
10. 子程序返回

子程序 `changestr` 的程序框图如下:



## 3.5 子程序 maxnumber 设计

### 3.5.1 子程序 maxnumber 设计思路

对于子程序 `maxnumber`, 需要能够找出输入字符串中 ASCII 码值最大的字符。设计的大致思路为: 首先调用系统程序将字符串输入缓冲并在字符串末尾加上终止符 '\$', 将最大值存储在 `dl` 中, 初始化(`dl`)=0, 之后用 `bx` 存储字符串的首地址, 遍历字符串, 若 `((bx)) > ((dl))`, 则 `((dl)) = ((bx))`, 遍历结束后, 打印 `dl` 中存储的值即可。

### 3.5.2 子程序 maxnumber 具体设计

对于 `maxnumber`, 具体设计如下:

入口参数: 无

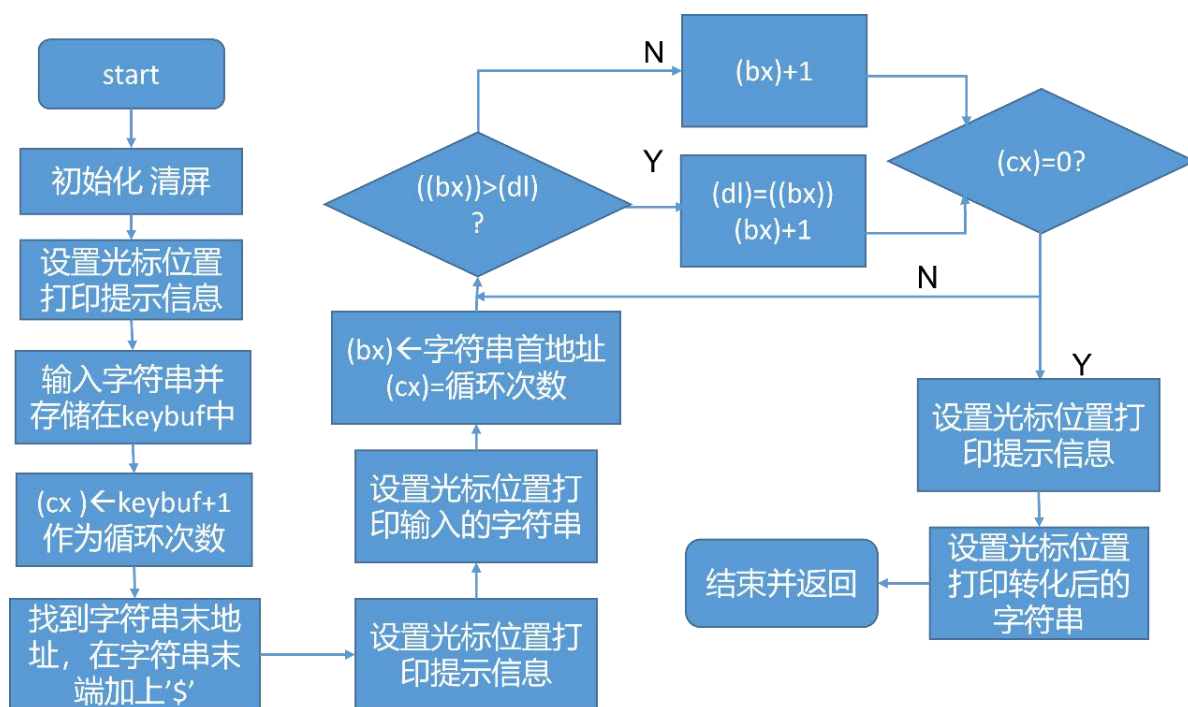


## 8 微机原理实验 付凯文

出口参数：无

1. 调用显示器，设置显示方式，清屏（用到的指令前文已提到，这里不再赘述）
2. 设置光标位置并打印提示信息
3. 输入字符串到缓冲区
4. 将 `keybuf+1` 的值传递给 `cx`，作为后续循环的次数
5. 找到字符串的末地址，在原字符串末尾加上终止标志 '\$'
6. 设置光标位置打印提示信息并输出原字符串
7. 设置 `(dl)=0`，字符串首地址存储于 `bx` 中，遍历数组，若 `((bx))>(dl)`，则 `(dl)=((bx))`
8. 设置光标位置打印提示信息并通过 `int 21h((ah)=2)` 输出 `dl` 中存储的内容
9. 设置光标位置打印输出信息
10. 子程序返回

子程序 `maxnumber` 的程序框图如下：



## 3.6 子程序 `sortnumber` 设计

### 3.6.1 子程序 `sortnumber` 设计思路

对于子程序 `sortnumber`，需要完成输入数据组的排序。然后转换成十六进制数按递增方式进行排序。设计的大致思路为：首先设置显示器并初始化打印提示信息，之后输入字符串存储在 `keybuf` 中，然后将字符串转化为数字。接着对数据进行排序，按十六进制形式输出数组，最后打印提示文档。



### 3.6.2 子程序 sortnumber 具体设计

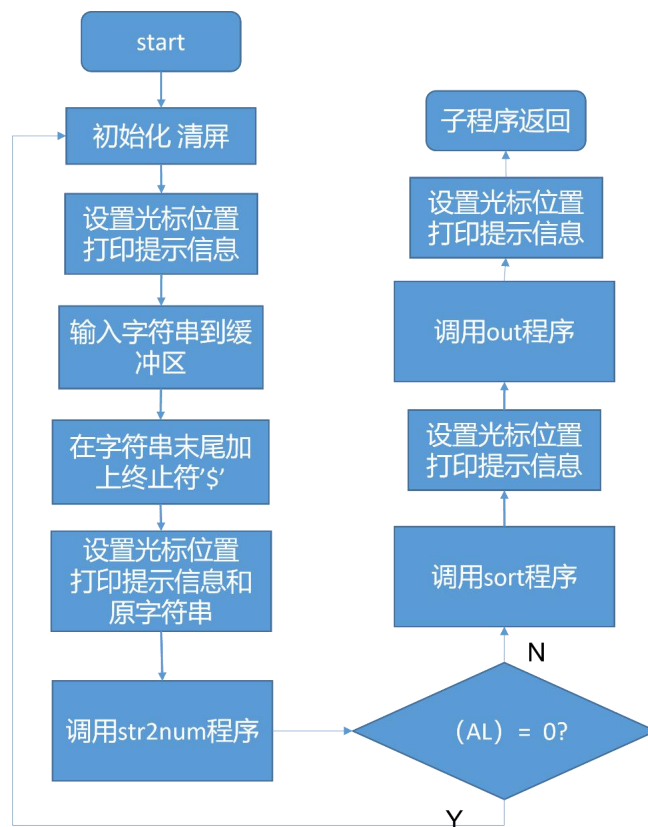
对于 sortnumber 程序，具体设计如下：

入口参数：无

出口参数：无

1. 调用显示器，设置显示方式，清屏
2. 设置光标位置并打印提示信息
3. 输入字符串到缓冲区
4. 找到字符串的末地址，在原字符串末尾加上终止标志'\$'
5. 输出提示信息和原字符串
6. 调用子程序 str2num 将字符串转为数字
7. 调用 sort 程序将数字按从小到大排序
8. 设置光标位置打印提示信息
9. 调用 out 程序将排列好的数字以十六进制形式输出
10. 子程序返回

子程序 sortnumber 的程序框图如下：



#### 3.6.2.1 子程序 str2num 具体设计

对于 str2num 程序，具体设计如下：

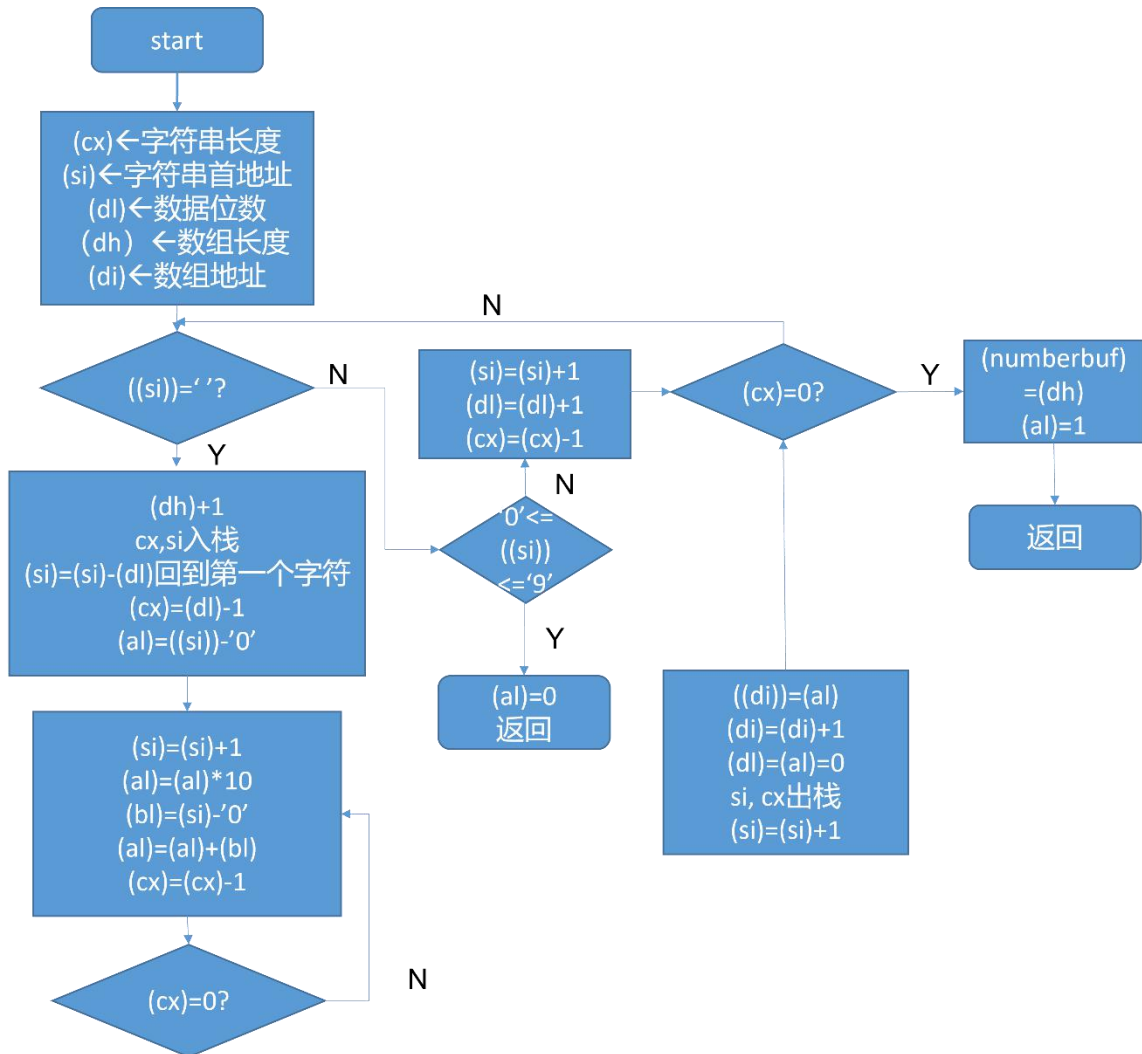
入口参数：无

出口参数：al

## 10 微机原理实验 付凯文

函数主要通过遍历的方式查找到分隔符',找到分隔符就开始处理分隔符前的数据,从高位开始,加上当前位,移到下一位,乘 10 后加上该位,直到该数据结束。遍历整个字符串,把所有数据放入 numberbuf 中。

程序的流程框图如下:



### 3.6.2.2 子程序 sort 具体设计

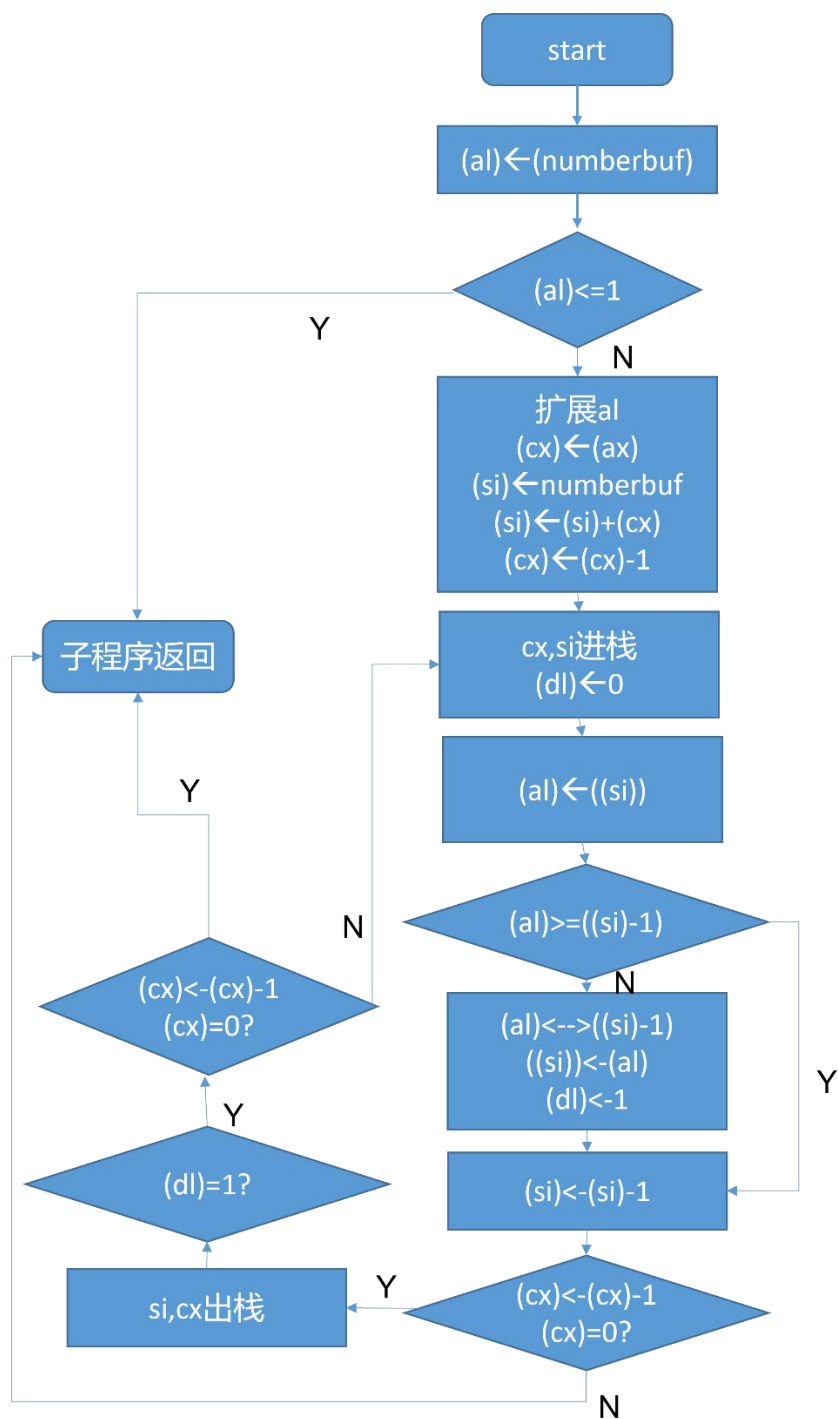
对于 sort 程序,具体设计如下:

入口参数: 无

出口参数: 无

对 N 个数据进行从小到大排序,采用冒泡法:从前向后,每两个数据进行比较,当前者大于后者时,交换两者的次序,否则不变,经过 N-1 次比较,可以将最大值交换到第 N 个单元,接着对前 N-1 个数据,重复上述过程。

程序的流程框图如下:



### 3.6.2.3 子程序 out 具体设计

对于 out 程序，具体设计如下：

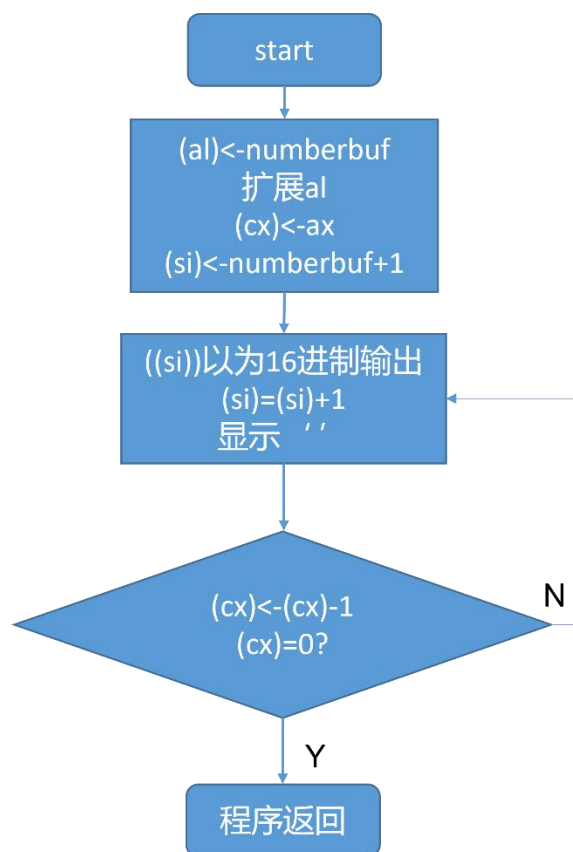
入口参数：无

出口参数：无

程序 out 较为简单，只需将数组长度放入 cx 中，数组首地址放入 si 中，之后 loop 循环打印十六进制数即可。

## 12 微机原理实验 付凯文

程序的流程框图如下：



### 3.7 子程序 showtime 设计

#### 3.7.1 子程序 showtime 设计思路

对于子程序 `showtime`，需要在屏幕右上角显示实时时间。设计的大致思路为：首先完成初始化以及清屏，之后调用系统程序得到当前系统时间。接着把得到的十六进制时间转化为十进制，最后将得到的十进制时间在屏幕右上角打印。之后检测键盘输入，若没有输入，则重新显示时间，输入 `esc` 退出系统，输入其他值重新执行功能四。

#### 3.7.2 子程序 showtime 具体设计

对于 `showtim`，具体设计如下：

入口参数：无

出口参数：无

1. 调用显示器，设置显示方式，清屏
2. 设置光标位置并打印提示信息
3. 调用 `int 21h ((ah)=2ch)` 获取系统时间，时，分，秒分别存储在 `ch,cl,dh` 寄存器中
4. 调用 `shownum` 函数依次将时分秒从十六进制转化为十进制并打印在右上角
5. 设置光标位置打印输出信息

13 微机原理实验 付凯文

6. 检测键盘输入，若无输入则重新显示时间，输入 `esc` 退出功能四，输入其他按键重新执行功能四。

3.7.2.1 子程序 `shownum` 具体设计

对于子程序 `shownum`，需要将时间从十六进制转化为十进制并在屏幕右上角打印。我们可以采用十六进制除法的方式得到对应的十进制数，将十六进制数除以十六进制的 10，即将要转换的十六进制数除以 `a`，则得到的商为对应十进制时间的十位数，余数为对应十进制时间的个位数。需要注意的是，子程序 `shownum` 在调用过程中会改变寄存器 `cx`, `dx` 中的数值造成寄存器使用冲突，因此，在调用 `shownum` 时需要进行现场保护，将 `cx`, `dx` 压入堆栈中，在 `shownum` 执行结束后将 `cx`, `dx` 从堆栈通弹出。

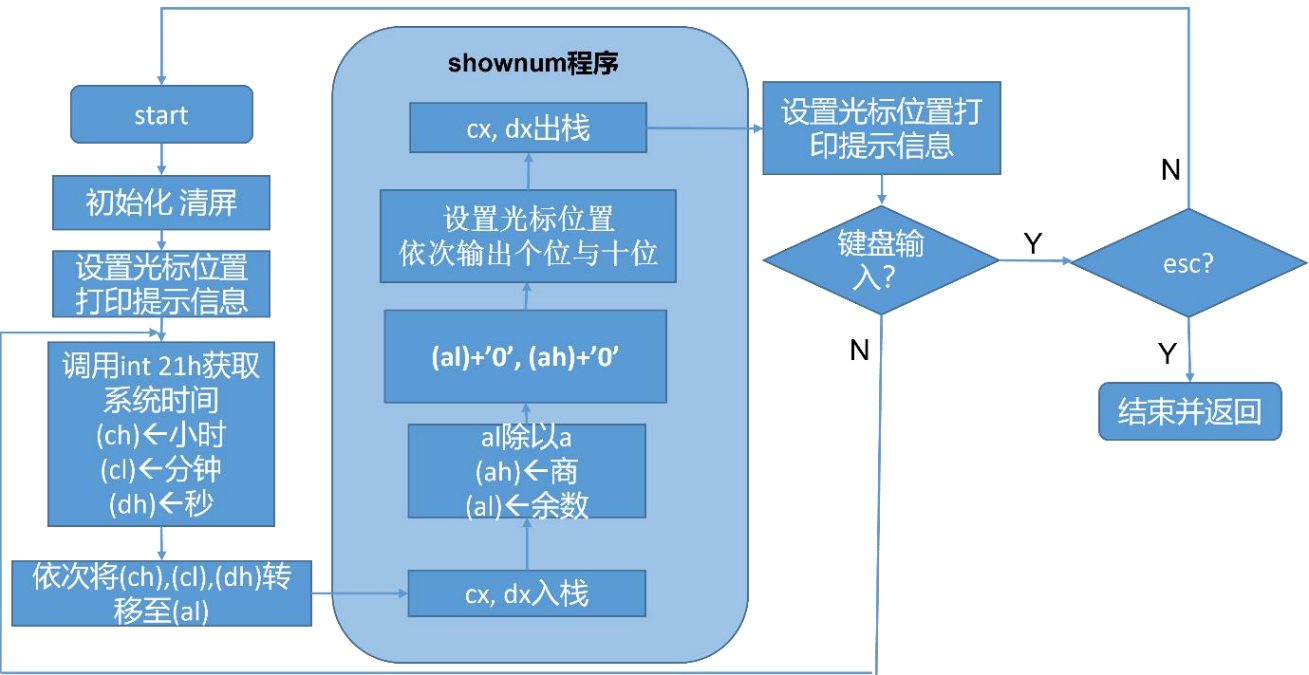
对于 `showtime`，具体设计如下：

入口参数：`al`

出口参数：无

- 1. 保护现场，将 `cx`, `dx` 压入栈
- 2. 将`(al)`除 `a`，将商存储在 `ah`，余数存储在 `al` 中
- 3. , 转化为十进制字符
- 4. 依次输出个位数和十位数
- 5. 恢复现场。将 `cx`, `dx` 弹出
- 6. 子程序返回

子程序 `showtime` 的程序框图如下：



### 4. 实验结果

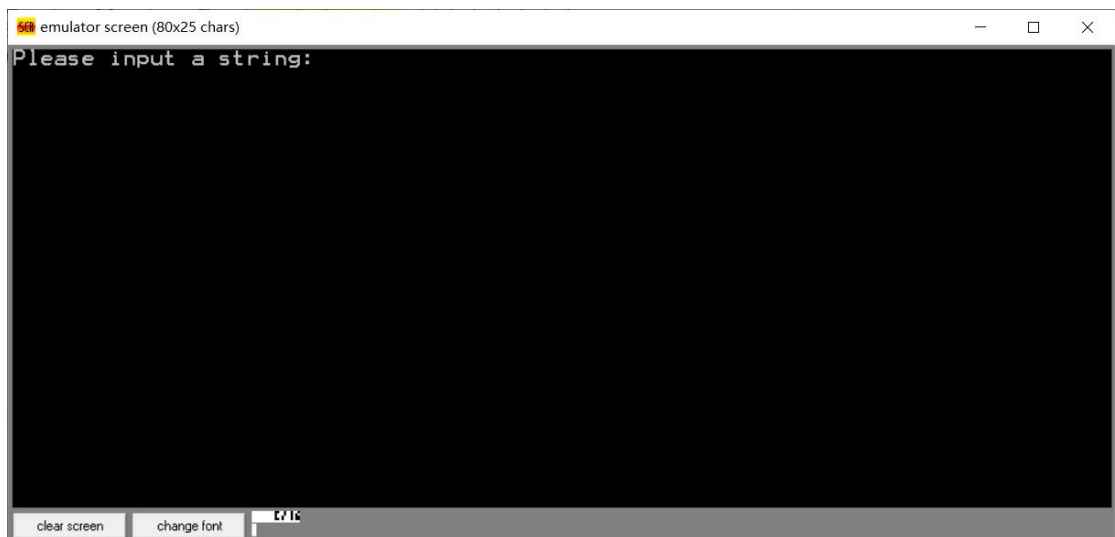
将编写好的各个程序模块进行拼接，组成最终的系统程序。在 emu8086 中进行编译，调试，最终生成了 version\_4.exe, version\_4exe.debug, version\_4exe.list, version\_4exe.symbol 四个文件，执行生成的 exe 文件，得到以下结果。

进入程序，首先输出提示信息，如图所示



```
emu8086 emulator screen (80x25 chars)
Welcome! Please press the corresponding number(1-5)
1.Change small letter into capital letter of string
2.Find the maximum ASCII value in the string
3.Sort strings from large to small
4.Show time
5.Exit
_
```

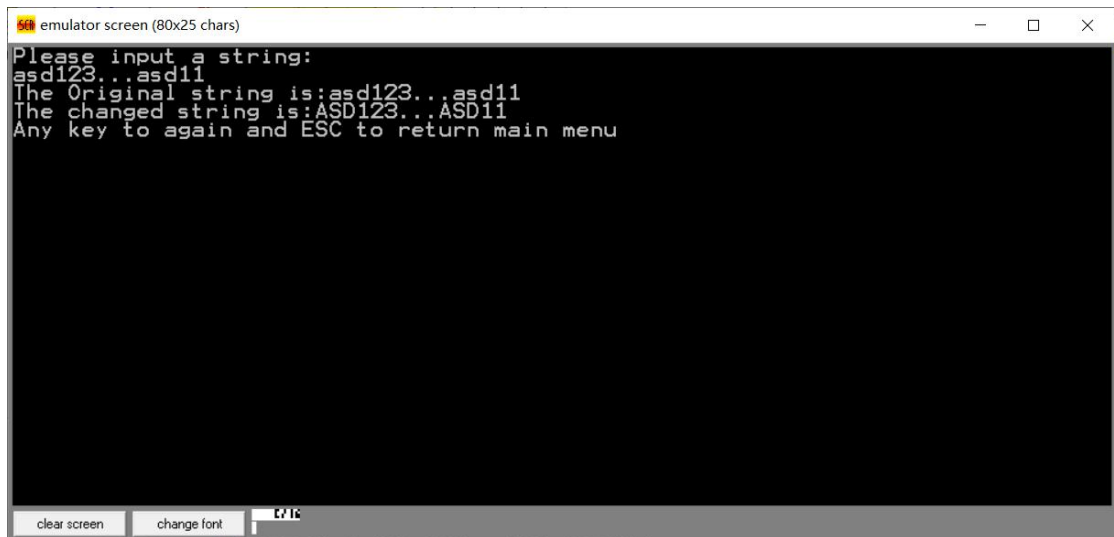
键入数字 1，进入第一个功能，系统首先输出提示信息



```
emu8086 emulator screen (80x25 chars)
Please input a string:
```

键入一个有小写字母，数字，特殊字符组成的字符串，系统输出如下

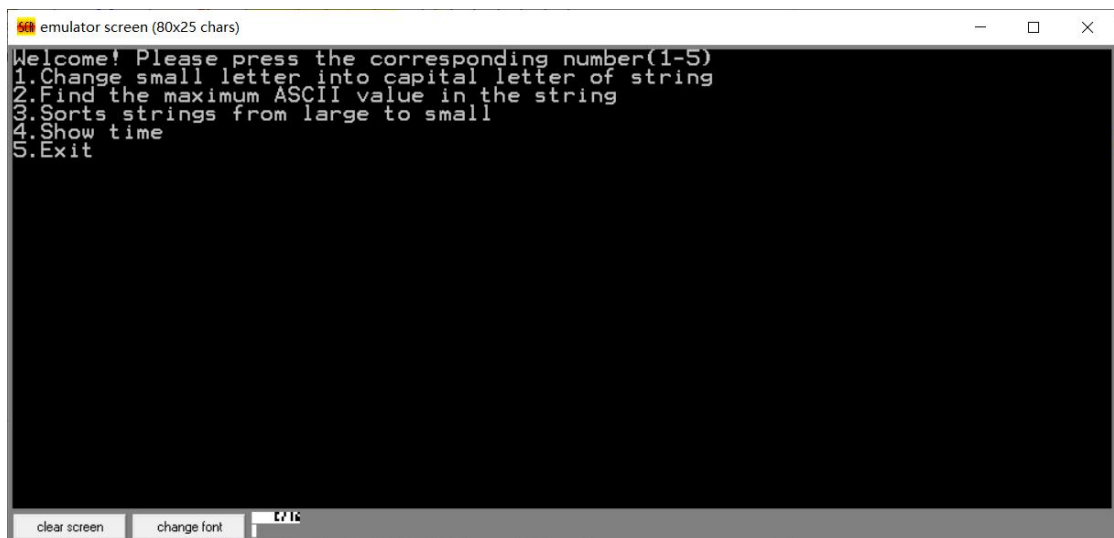
## 15 微机原理实验 付凯文



输入除 esc 外的任意字符，得到以下结果



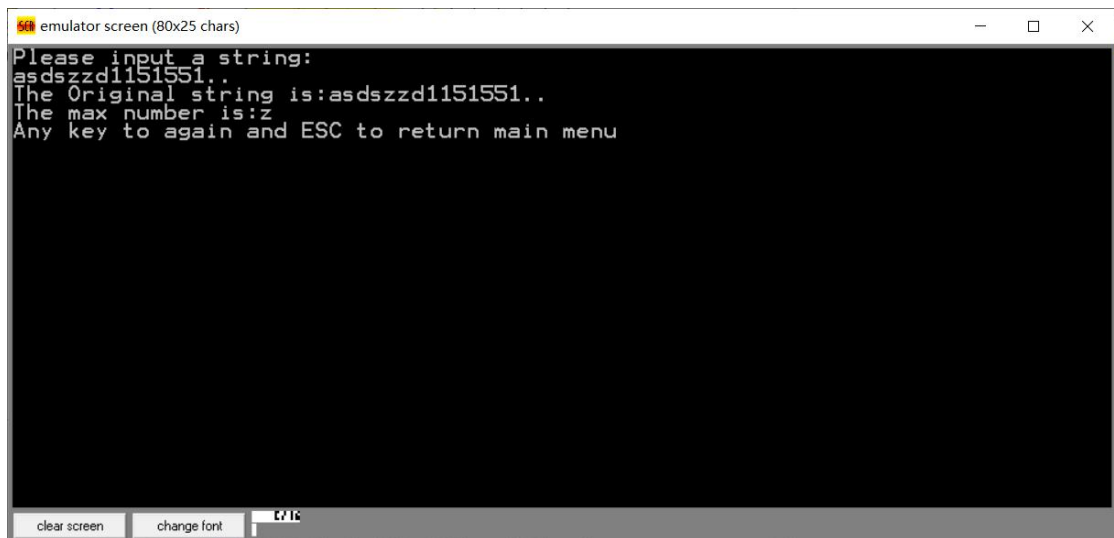
esc 返回主菜单





## 16 微机原理实验 付凯文

功能二如图所示:

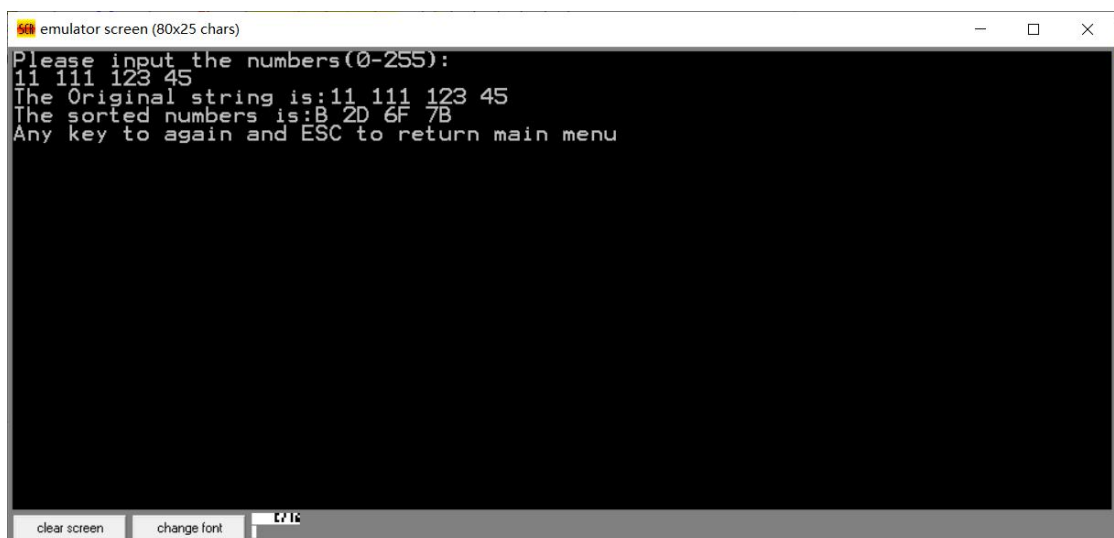


The screenshot shows a window titled "emulator screen (80x25 chars)". The text displayed is as follows:

```
Please input a string:
asdszzd1151551..
The Original string is:asdszzd1151551..
The max number is:z
Any key to again and ESC to return main menu
```

At the bottom of the window, there are two buttons: "clear screen" and "change font". A status bar at the very bottom shows "C/16".

功能三如图所示:



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

```
Please input the numbers(0-255):
11 111 123 45
The Original string is:11 111 123 45
The sorted numbers is:B 2D 6F 7B
Any key to again and ESC to return main menu
```

At the bottom of the window, there are two buttons: "clear screen" and "change font". A status bar at the very bottom shows "C/16".

功能四如图所示:



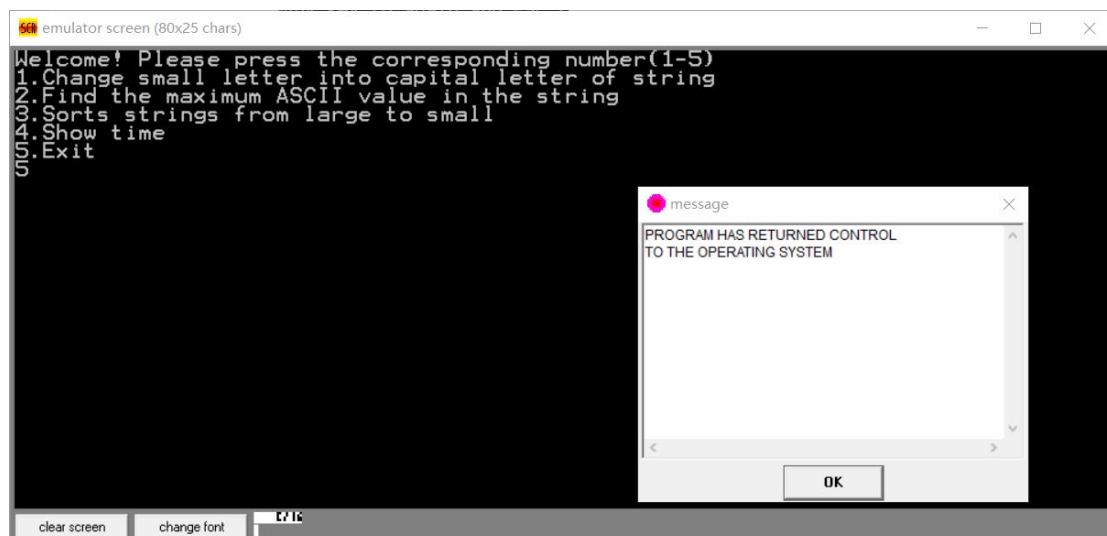
The screenshot shows a window titled "emulator screen (80x25 chars)". The text displayed is as follows:

```
Now time is: 21:16:12
Any key to again and ESC to return main menu
```

At the bottom of the window, there are two buttons: "clear screen" and "change font". A status bar at the very bottom shows "C/16".

## 17 微机原理实验 付凯文

键入数字 5，退出系统



经过实验测试，系统完成的实现所有功能，全部模块无 bug

## 5. 实验总结

通过本次实验，我学习到了汇编语言的程序设计方法，切实地掌握了编程知识。同时，我掌握了汇编程序的调试方法，更加熟练的掌握了各种汇编指令以及一些相似指令之间的区别。另外，在本次实验中，我学习到了 DOS 中断 int 21h 与 int 10h 的一些常用功能，掌握了 emu8086 的使用。

本次实验中，同样也遇到了一些问题，比如 emu8086 不能利用 '?' 开辟存储空间，同时，emu 调用远程函数会出现错误等。最终在查阅相关资料后顺利解决了这些问题。

总的来说，本次实验让我收益良多，对汇编语言有了更深层次的理解。

## 6. 附录

```
1. ;author: fkw
2. ;creat time: 2021/11/20
3. ;coding: utf-8
4. ;version: 3.0
5. ;finish time: 2021/11/24
6.
7. ;定义堆栈段
8. stack segment stack
9.     db 256 dup(0)
10. top label word ; 插入一个变量，名称为 top，类型为 word，即栈顶
11. stack ends
12.
13. ;定义数据段
```

## 18 微机原理实验 付凯文

```
14. data segment
15. ;定义输入提示
16.     tips0 db 'Welcome! Please press the corresponding number(1-5)',0dh,0ah,'$'
17.     tips1 db '1.Change small letter into capital letter of string$',0dh,0ah,'$'

18.     tips2 db '2.Find the maximum ASCII value in the string',0dh,0ah,'$'
19.     tips3 db '3.Sort strings from large to small',0dh,0ah,'$'
20.     tips4 db '4.Show time',0dh,0ah,'$'
21.     tips5 db '5.Exit',0dh,0ah,'$'
22.     tipsn db 'Any key to again and ESC to return main menu$',0dh,0ah,'$'
23.     in_str db 'Please input a string: $'
24.     out_str db 'The changed string is:$'
25.     in_num db 'Please input the numbers(0-255):$'
26.     out_num db 'The sorted numbers is:$'
27.     max_num db 'The max number is:$'
28.     out_time db 'Now time is:$'
29.     str_pre db 'The Original string is:$'
30.     ;字符缓冲区
31.     keybuf  db 129 ;容量 129
32.             db 0    ;存放 string 中字符个数
33.             db 61 dup(0) ;存放 string
34.     numbuf db 0
35.             db 20 dup(0)
36. data ends
37.
38. ;code 段定义
39. code segment
40.     assume cs:code ds:data,ss:stack ;声明数据段, 代码段, 堆栈段
41. start:
42.     mov ax, data ;data->ax    立即数不能直接到段寄存器, 通常段寄存器->通用寄存器->段寄存器
43.     mov ds, ax   ;ax->ds      ds: 数据段寄存器,用于存放当前数据段的段地址
44.     mov ax, stack ;stack->ax
45.     mov ss, ax   ;ax->ss      ss: 堆栈段寄存器,用于存放当前堆栈段的段地址
46.     mov sp, offset top ;sp: 堆栈指针, 用于保存堆栈段的段内偏移地址
47.     ; offset top: 栈顶的偏移地址
48. main:
49.     call menu ;调用 menu
50.     again:
51.     mov ah,2
52.     mov bh,0 ;页号
53.     mov dl,0 ;列号
54.     mov dh,6 ;行号
55.     int 10h ;光标位置设置
```

## 19 微机原理实验 付凯文

```
55.    mov ah,1
56.    int 21h    ;带返回码结束用户程序
57.    cmp al,'1'
58.    jb again
59.    cmp al,'5'
60.    ja again
61.    ;功能选择
62.    cmp al,'1'
63.    jz opt1
64.    cmp al,'2'
65.    jz opt2
66.    cmp al,'3'
67.    jz opt3
68.    cmp al,'4'
69.    jz opt4
70.    cmp al,'5'
71.    jz opt5
72.
73. opt1:
74.    call changestr
75.    ;带回显的字符输入
76.    mov ah,8
77.    int 21h
78.    ;输入 esc 则调到 main
79.    cmp al,1bh
80.    jz main
81.    ;输入其他的跳到子程序 opt1
82.    jmp opt1
83. opt2:
84.    call maxnumber
85.    mov ah,8
86.    int 21h
87.    cmp al,1bh
88.    jz main
89.    jmp opt2
90. opt3:
91.    call sortnumber
92.    mov ah,8
93.    int 21h
94.    cmp al,1bh
95.    jz main
96.    jmp opt3
97. opt4:
98.    call showtime
```

## 20 微机原理实验 付凯文

```
99.      mov ah,8
100.     int 21h
101.     cmp al,1bh
102.     jz  main
103.     jmp opt4
104. opt5:
105.     mov ah,4ch
106.     int 21h
107.
108. menu:
109.     mov ah,0    ;入口参数 ah=0
110.     mov al,3    ;显示模式 80*25 16 色
111.     mov bl,0
112.     int 10h
113.     ;设置光标位置
114.     mov ah,2    ;入口参数 ah=2,设置光标位置
115.     mov bh,0    ;bh=页号
116.     mov dl,0    ;dl=列号
117.     mov dh,0    ;dh=行号
118.     int 10h    ;光标位置设置
119.     ;输出 tips0
120.     mov ah,9    ;入口参数 ah=9,在光标处显示字符串
121.     lea dx,tips0 ;取 tips0 的地址
122.     int 21h    ;显示字符串
123.     ;设置光标位置
124.     mov ah,2
125.     mov dl,0
126.     mov dh,1
127.     int 10h
128.     ;输出 tips1
129.     mov ah,9
130.     lea dx,tips1
131.     int 21h
132.     ;设置光标位置
133.     mov ah,2
134.     mov dl,0
135.     mov dh,2
136.     int 10h
137.     ;输出 tips2
138.     mov ah,9
139.     lea dx,tips2
140.     int 21h
141.     ;设置光标位置
142.     mov ah,2
```

## 21 微机原理实验 付凯文

```
143.    mov dl,0
144.    mov dh,3
145.    int 10h
146.    ;输出 tips3
147.    mov ah,9
148.    lea dx,tips3
149.    int 21h
150.    ;设置光标位置
151.    mov ah,2
152.    mov dl,0
153.    mov dh,4
154.    int 10h
155.    ;输出 tips4
156.    mov ah,9
157.    lea dx,tips4
158.    int 21h
159.    ;设置光标位置
160.    mov ah,2
161.    mov dl,0
162.    mov dh,5
163.    int 10h
164.    ;输出 tips5
165.    mov ah,9
166.    lea dx,tips5
167.    int 21h
168.    ret
169.
170. ;小写转大写
171. changestr:
172. rechg:
173.    ;清屏
174.    mov ah,0
175.    mov al,3    ;设置显示方式
176.    mov bl,0
177.    int 10h
178.    ;输入提示光标位置设置
179.    mov ah,2
180.    mov bh,0    ;页号
181.    mov dl,0    ;列号
182.    mov dh,0    ;行号
183.    int 10h
184.    ;输出 tips
185.    mov ah,9
186.    lea dx,in_str
```

## 22 微机原理实验 付凯文

```
187.    int 21h
188.    ;输入 sting
189.
190.    mov ah,2
191.    mov dl,0
192.    mov dh,1
193.    int 10h
194.    ;键盘输入到缓存区
195.    mov ah,0ah
196.    lea dx,keybuf
197.    int 21h
198.    ;判定输入字符串是否为空
199.    cmp keybuf+1,0
200.    jz rechg ;若为空重新开始
201.    ;字符串末尾加 $
202.    lea bx,keybuf+2 ;bx 为输入字符串的首地址
203.    mov al,keybuf+1 ;keybuff+1 为输入的字符个数
204.    cbw             ;al->ax 扩展
205.    mov cx,ax        ;字符串长度->cx 用作 loop 的传入参数
206.    add bx,ax        ;ax->bx bx 为字符串末尾的下一位地址
207.    mov byte ptr [bx], '$' ;在输入字符串的末尾加上结束标识符
208.    lea bx,keybuf+2
209.
210.    ;设置光标位置
211.    mov ah,2
212.    mov bh,0
213.    mov dl,0
214.    mov dh,2
215.    int 10h
216.
217.    mov ah,9
218.    lea dx ,str_pre
219.    int 21h
220.    mov ah,9
221.    lea dx ,keybuf+2
222.    int 21h
223.    lea bx,keybuf+2
224.
225.lchg:
226.    ;判断是否需要转换
227.    cmp byte ptr [bx],61h
228.    jb nochg ;不需要转换跳转
229.    and byte ptr [bx],0dfh ;小写转大写
230.nochg:
```



## 23 微机原理实验 付凯文

```
231.    inc bx    ;bx++
232.    loop lchg    ;loop 循环，传入参数（循环次数为 cx）
233.
234.    ;设置光标位置
235.    mov ah,2
236.    mov bh,0
237.    mov dl,0
238.    mov dh,3
239.    int 10h
240.    ;输出 tips
241.    mov ah,9
242.    lea dx,out_str
243.    int 21h
244.    ;输出改变后的字符串
245.    mov ah,9
246.    lea dx,keybuf+2
247.    int 21h
248.    ;清屏，设置光标位置
249.    mov ah,2
250.    mov bh,0
251.    mov dl,0
252.    mov dh,4
253.    int 10h
254.    ;输出 tips
255.    mov ah,9
256.    lea dx,tipsn
257.    int 21h
258.    ret
259.
260. ;找最大数
261. maxnumber:
262. remax:
263.    ;设置显示器方式
264.    mov ah,0
265.    mov al,3
266.    mov bl,0
267.    int 10h ;清屏
268.    ;光标位置
269.    mov ah,2
270.    mov bh,0
271.    mov dl,0
272.    mov dh,0
273.    int 10h
274.    ;输出 tips
```

## 24 微机原理实验 付凯文

```
275.    mov ah,9
276.    lea dx,in_str
277.    int 21h
278.    ;设置光标位置
279.    mov ah,2
280.    mov dl,0
281.    mov dh,1
282.    int 10h
283.    ;输入字符串到缓存区 keybuf+1 为字符个数, keybuf+2 开始为字符串
284.    mov ah,0ah
285.    lea dx,keybuf
286.    int 21h
287.    ;判断字符串是否为空
288.    cmp keybuf+1,0
289.    jz remax ;字符串为空重新输入
290.
291.    lea bx,keybuf+2 ;bx 为字符串首地址
292.    mov al,keybuf+1 ;al 为输入的字符串长度
293.    cbw ;al->ax 扩展 al
294.    mov cx,ax ;cx 用作 loop 的传入参数
295.    add bx,ax ;bx 为字符串末尾的下一个地址
296.    mov byte ptr [bx],'$' ;末尾加 $
297.
298.    ;设置光标位置
299.    mov ah,2
300.    mov bh,0
301.    mov dl,0
302.    mov dh,2
303.    int 10h
304.
305.    mov ah,9
306.    lea dx ,str_pre
307.    int 21h
308.    mov ah,9
309.    lea dx ,keybuf+2
310.    int 21h
311.
312.    ;设置光标位置
313.    mov ah,2
314.    mov bh,0
315.    mov dl,0
316.    mov dh,3
317.    int 10h
318.    ;输出 tips
```

## 25 微机原理实验 付凯文

```
319.    mov ah,9
320.    lea dx,max_num
321.    int 21h
322.
323.
324.    mov dl,0 ;dl 存储最大值
325.    lea bx,keybuf+2
326.
327. ;选出最大值
328.lcmp:
329.    cmp [bx],dl ;判断是否小于 dl
330.    jb nolchg
331.    mov dl,[bx]
332.nolchg:
333.    inc bx
334.    loop lcmp
335.
336.
337.    mov ah,2
338.    int 21h
339.    ;设置光标位置
340.    mov ah,2
341.    mov bh,0
342.    mov dl,0
343.    mov dh,4
344.    int 10h
345.    ;输出 tips
346.    mov ah,9
347.    lea dx,tipsn
348.    int 21h
349.    ret
350. ;转 16 进制并 sort
351.sortnumber:
352.resort:
353.    mov ah,0
354.    mov al,3
355.    mov bl,0
356.    int 10h
357.
358.    mov ah,2
359.    mov bh,0
360.    mov dl,0
361.    mov dh,0
362.    int 10h
```

## 26 微机原理实验 付凯文

```
363.
364.     mov ah,9
365.     lea dx,in_num
366.     int 21h
367.
368.     mov ah,2
369.     mov dl,0
370.     mov dh,1
371.     int 10h
372.
373.     mov ah,0ah
374.     lea dx,keybuf
375.     int 21h
376.
377.     lea bx,keybuf+2 ;bx 为字符串首地址
378.     mov al,keybuf+1 ;al 为输入的字符串长度
379.     cbw ;al->ax 扩展 al
380.     mov cx,ax ;cx 用作 loop 的传入参数
381.     add bx,ax ;bx 为字符串末尾的下一个地址
382.     mov byte ptr [bx],'$' ;末尾加 $
383.
384.     ;设置光标位置
385.     mov ah,2
386.     mov bh,0
387.     mov dl,0
388.     mov dh,2
389.     int 10h
390.
391.     mov ah,9
392.     lea dx ,str_pre
393.     int 21h
394.     mov ah,9
395.     lea dx ,keybuf+2
396.     int 21h
397.     call cin_int
398.     cmp al,0
399.     jz resort
400.     cmp numbuf,0
401.     jz resort
402.
403.     mov ah,2
404.     mov bh,0
405.     mov dl,0
406.     mov dh,3
```

## 27 微机原理实验 付凯文

```
407.     int 10h
408.
409.     mov ah,9
410.     lea dx,out_num
411.     int 21h
412.
413.
414.
415.     call mpsort
416.     call int_out
417.     mov ah,2
418.     mov bh,0
419.     mov dl,0
420.     mov dh,4
421.     int 10h
422.     mov ah,9
423.     lea dx,tipsn
424.     int 21h
425.     ret
426.
427. cin_int:
428.     mov cl,keybuf+1
429.     lea si,keybuf+2
430.     mov ch,0
431.     mov dh,10
432.     mov al,0
433.     mov dl,0
434. fndnum:
435.     cmp byte ptr[si],' '
436.     jz addnum
437.     cmp byte ptr[si],'0'
438.     jb errnum
439.     cmp byte ptr[si],'9'
440.     ja errnum
441.     mov dl,1
442.     mul dh
443.
444.     xor bh,bh
445.     mov bl,[si]
446.     add ax,bx
447.     sub ax,'0'
448.     cmp ah,0
449.     ja errnum
450.     jmp next
```

## 28 微机原理实验 付凯文

```
451. addnum:
452.     cmp dl,1
453.     jnz next
454.     inc ch
455.     call addnew
456.     mov dl,0
457.     mov al,0
458. next:
459.     inc si
460.     dec cl
461.     cmp cl,0
462.     jnz fndnum
463.     cmp dl,1
464.     jnz total
465.     inc ch
466.     call addnew
467. total:
468.     mov numbuf,ch
469.     mov al,1
470.     jmp crtnum
471.
472. errnum:
473.     mov al,0
474.
475. crtnum:
476.     ret
477.
478.
479. addnew:
480.
481.     push ax
482.     lea bx,numbuf
483.     mov al,ch
484.     cbw
485.     add bx,ax
486.     pop ax
487.     mov [bx],al
488.     ret
489.
490. mpsort:
491.     mov al,numbuf
492.     cmp al,1
493.     jbe nosort
494.     cbw
```

## 29 微机原理实验 付凯文

```
495.      mov cx,ax
496.      lea si,numbuf
497.      add si,cx
498.      dec cx
499.lp1:
500.      push cx
501.      push si
502.      mov dl,0
503.lp2:
504.      mov al,[si]
505.      cmp al, [si-1]
506.      jae noxchg
507.      xchg al, [si-1]
508.      mov [si],al
509.      mov dl,1
510.noxchg:
511.      dec si
512.      loop lp2
513.      pop si
514.      pop cx
515.      cmp dl,1
516.      jnz nosort
517.      loop lp1
518.
519.nosort:ret
520.
521.
522.
523.int_out:
524.      mov al,numbuf
525.      cbw
526.      mov cx,ax
527.      mov bl,10h
528.      lea si,numbuf + 1
529.print:
530.      mov al,[si]
531.      call outnum
532.      inc si
533.      mov ah,2
534.      mov dl,' '
535.      int 21h
536.      loop print
537.      ret
538.
```



## 30 微机原理实验 付凯文

```
539.
540.
541. outnum:
542.     mov ah,0
543.     div bl
544.     push ax
545.     cmp ah,10
546.     jb pnum
547.     add ah,7
548. pnum: add ah,30h
549.     mov dl,ah
550.     pop ax
551.     push dx
552.     cmp al,0
553.     jz outn
554.     call outnum
555. outn:
556.     pop dx
557.     mov ah,2
558.     int 21h
559.     ret
560.
561.
562.
563. ;时间展示
564. showtime:
565.     ;清屏
566.     mov ah,0
567.     mov al,3
568.     mov bl,0
569.     int 10h
570.     ;光标位置
571.     mov ah,2
572.     mov bh,0
573.     mov dl,0
574.     mov dh,0
575.     int 10h
576.     ;输出 tips
577.     mov ah,9
578.     lea dx,out_time
579.     int 21h
580. dis:
581.     ;光标位置
582.     mov ah,2
```

## 31 微机原理实验 付凯文

```
583.    mov bh,0
584.    mov dl,72
585.    mov dh,0
586.    int 10h
587.    ;取系统时间
588.    mov ah,2ch
589.    int 21h
590.    ;显示小时
591.    mov al,ch
592.    call shownum
593.    ;显示:
594.    mov ah,2
595.    mov dl,':'
596.    int 21h
597.    ;显示分钟
598.    mov al,cl
599.    call shownum
600.    ;显示:
601.    mov ah,2
602.    mov dl,':'
603.    int 21h
604.    ;显示秒
605.    mov al,dh
606.    call shownum
607.    ;清屏,设置光标位置
608.    mov ah,2
609.    mov bh,0
610.    mov dl,0
611.    mov dh,1
612.    int 10h
613.    ;输出 tips
614.    mov ah,9
615.    lea dx,tipsn
616.    int 21h
617.    ;检测键盘输入
618.    mov ah,0bh
619.    int 21h
620.    cmp al,0
621.    jz dis
622.
623. ret
624.
625. shownum:
626. ;把 al 中的数字以十进制输出
```

## 32 微机原理实验 付凯文

```
627. ;入口参数 al
628. ;出口参数: 无
629.     cbw
630.     push cx
631.     push dx
632.
633.     mov cl,10
634.     div cl
635.
636.     add ah,'0'
637.     mov bh,ah
638.     add al,'0'
639.
640.     mov ah,2
641.
642.     mov dl,al
643.     int 21h
644.
645.     mov dl,bh
646.     int 21h
647.
648.     pop dx
649.     pop cx
650.     ret
651.
652. code ends
653. end start
```