

东南大学

《微机系统与接口实验》

实验报告

实验九 时钟实验

姓 名：薛宇飞

学 号：04020235

同 组：

学 号：

专 业：信息工程

实 验 室：金智楼硬件实验室

实验时间：2022 年 4 月 27 日

报告时间：2022 年 4 月 27 日

评定成绩：

评阅教师：裴文江

目录

1 实验目的与内容	3
2 预习报告	3
1 实验任务	3
2 实验原理	3
3 流程框图	4
4 实验可能用到的代码	5
5 实验代码	5
3 实验效果图	11
4 思考题	11
5 附加任务描述及完成	12
1 附加任务 1	12
2 附加任务 2	13
3 附加任务 3	15
6 实验总结	16
参考文献	16

一. 实验目的与内容

1. 结合实验教材^[1-2]，熟悉系统功能调用 INT 21H 的有关功能。
2. 编写时钟程序。

二. 预习报告

(一) 实验任务

1. 执行时钟程序时，屏幕上显示提示符“:”，由键盘输入当前时、分和秒值，即 XX:XX:XX√，随即显示时间并不停地计时。
2. 当有键按下时，立即停止计时，返回 DOS。

(二) 实验原理

首先利用系统调用 INT 21H 中 02H 功能，在 CRT 上显示一个提示符“:”，要求用户从键盘输入时钟初值 (即当前时间)，其输入格式为 XX(时):XX(分):XX(秒)√。然后利用 0AH 功能调用接收从键盘输入的字符串，并将接收的字符串存入到缓冲区。

在利用 0AH 功能调用前要设置一个缓冲区，在调用时，用 DX 作为输入缓冲区的指针，由键盘输入的字符存入该缓冲区，直至遇到回车键为止。程序中把输入的‘时’、‘分’、‘秒’初值分别从输入缓冲区中取出，各自放在一个寄存器中，然后调用一个延时 1 秒钟的子程序，每过 1 秒使秒值增 1，然后检查是否已为 60 秒，若不是则转显示；若是，则使秒值为 0，分值增 1，检查是否已为 60 分，若不是则转显示，若是，则使分值为 0，时值增 1，接着检查时值是否为 24 小时，若不是则转显示，若是，则使时值为 0，接着也是转显示。若使程序运行停止，只要有键按下，即可返回 DOS。下面列出两种判别是否有键按下的方法：

1. 读键扫描码：

	code		
1	IN	AL,60H	;读键扫描码
2	TEST	AL,80H	
3	JZ	AAA	;有键按下,就转AAA
4		:	
5		:	
6	AAA:	MOV AH,4CH	
7	INT	21H	

2. 调用 INT 21H 中 06 功能：

code

```

1      MOV    AH,06
2      MOV    DL,0FFH      ;判断是否有键按下,有键按下则转AAA
3      INT     21H
4      JNZ    AAA
5      :
6      :
7  AAA: MOV    AH,4CH
8      INT     21H
    
```

(三) 流程框图

实验流程图如图 1所示。

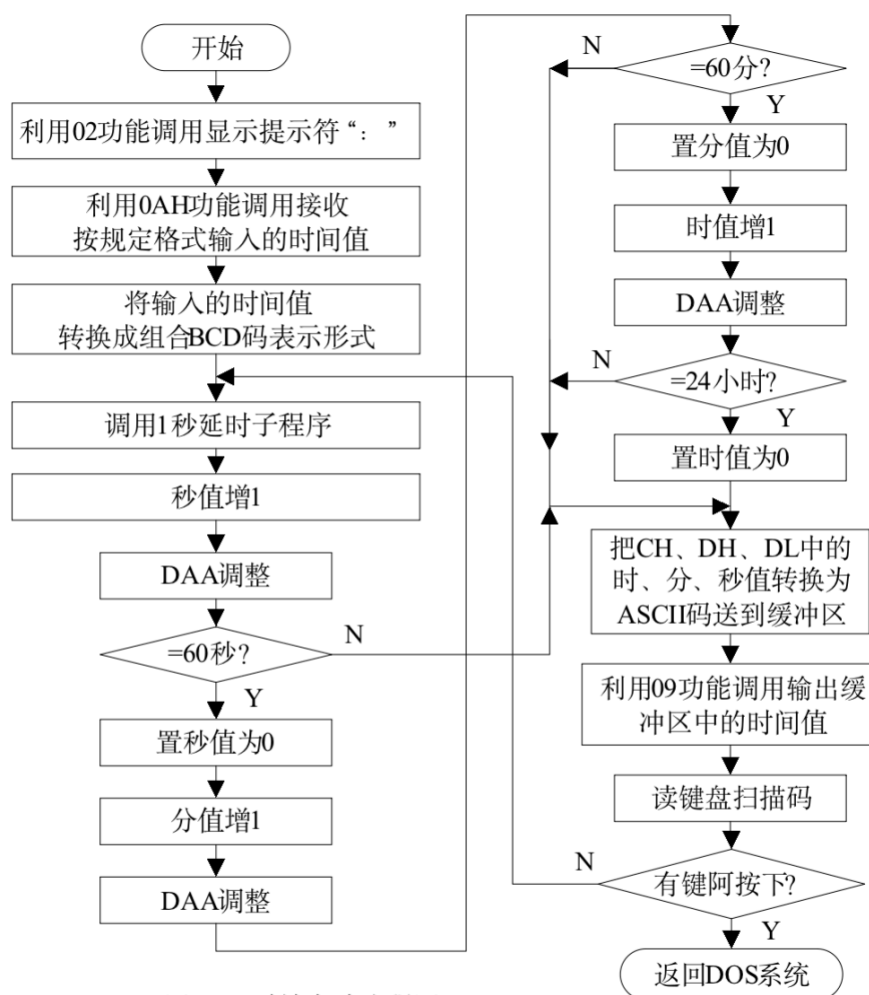


图1-9-1 时钟实验流程图

图 1

(四) 实验可能用到的代码

延时 1 秒子程序 DELAY:

delay

```
1  DELAY PROC
2      PUSH    CX
3      PUSH    AX
4      MOV     CX,0FFFFH
5  GOON: DEC     CX
6      JNE     GOON
7      POP     AX
8      POP     CX
9      RET
10 DELAY ENDP
```

(五) 实验代码

时钟实验代码如下（包括三个附加程序）:

CLK

```
1  DATA SEGMENT
2  TIME_IN      DB 9,?,9 DUP(?),'$'    ;缓冲区格式要求:
3                                          ;允许最大字符数（带回车）、
4                                          ;实际字符数（系统自动写入回车）、
5                                          ;缓冲区、ODH
6  PRINT_ETR    DB 0DH,0AH,'$'         ;设置一回车输出串,
7                                          ;在“:”显示后输出,换行缩进
8                                          ;汇编里回车是回到本行首位,
9                                          ;换行是到下一行同样位置
10 TIME_CHG     DB ?,?,?,?            ;设置一时间存储空间,
11                                          ;便于在执行时间的ASCII码与压缩BCD码
12 WRONG_VALUE  DB 'Wrong value! Try again','$'
13                                          ;错误类型1
14 WRONG_NUMBER DB 'The input must be colons or number! Try again','$'
15                                          ;错误类型2
16 DATA ENDS
17
18 STACK SEGMENT
19 STACK_SPACE  DB 8 DUP(?)            ;开辟栈段,
20                                          ;dos调用delay以及‘分’、‘秒’
```

```
19                                     ;保护会用到栈段
20 STACK ENDS
21
22 CODE SEGMENT
23 ASSUME CS:CODE,DS:DATA,ES:DATA
24 START:
25     MOV AX,DATA
26     MOV DS,AX
27     MOV ES,AX                       ;设置DS段和ES段的段地址
28 INPUT:
29     MOV DL,':'                     ;显示时间输入提示符号“:”
30     MOV AH,02H
31     INT 21H
32     LEA DX,PRINT_ETR               ;附加任务1:
33                                     ;显示“:”后输出一个回车,
34                                     ;便于时间的正确输出
35     MOV AH,09H
36     INT 21H
37     LEA DX,TIME_IN                 ;接收输入的时间
38     MOV AH,0AH
39     INT 21H
40 LENTH_TEST:                         ;长度测试
41     MOV AL,TIME_IN+1               ;实际键入字数
42     CMP AL,08H                     ;和8比较,不相等跳转到 WRONG_INPUT
43     JNE WRONG_INPUT_2
44 FORMAT_TEST:                       ;格式测试
45     MOV AL,TIME_IN+4               ;格式:XX:XX:XX
46     CMP AL,':'
47     JNE WRONG_INPUT_2
48     MOV AL,TIME_IN+7
49     CMP AL,':'
50     JNE WRONG_INPUT_2
51 INIT_LOW_LIMIT_TEST:
52     LEA SI,TIME_IN+2
53     MOV CL,03H                     ;循环3次
54 LOW_LIMIT_TEST:
55     MOV AL,[SI]
56     MOV AH,[SI+1]
57     CMP AL,30H                     ;字符‘0’的ASCII码是30,不能小于0
58     JB WRONG_INPUT_2               ;无符号小于则跳转
59     CMP AH,30H                     ;字符‘0’的ASCII码是30,不能小于0
```

```
60         JB WRONG_INPUT_2                ;无符号小于则跳转
61         ADD SI,0003H
62         LOOP LOW_LIMIT_TEST
63     INIT_HIGH_LIMIT_TEST:
64         LEA SI,TIME_IN+2
65         MOV CL,03H
66     HIGH_LIMIT_TEST:
67         MOV AL,[SI]
68         MOV AH,[SI+1]
69         CMP AL,39H                        ;字符‘9’的ASCII码是39,
70                                           ;不能超过9,先得是个数字
71         JA WRONG_INPUT_2                ;无符号大于则跳转
72         CMP AH,39H
73         JA WRONG_INPUT_2
74         ADD SI,0003H
75         LOOP HIGH_LIMIT_TEST
76     HOUR_HIGH_TEST:
77         MOV AL,TIME_IN+2
78         CMP AL,32H                        ;字符‘2’的ASCII码是32,
79                                           ;‘时’第一位不能超过2,
80                                           ;‘分’和‘秒’同理
81         JA WRONG_INPUT_1
82     HOUR_FULL_HIGH_TEST:
83         MOV AL,TIME_IN+3
84         MOV AH,TIME_IN+2
85         CMP AX,3233H                     ;字符‘0203’的ASCII码是3233
86         JA WRONG_INPUT_1
87     MINUTE_TEST:
88         MOV AL,TIME_IN+5
89         CMP AL,35H
90         JA WRONG_INPUT_1
91     SECOND_TEST:
92         MOV AL,TIME_IN+8
93         CMP AL,35H
94         JA WRONG_INPUT_1
95         JMP INIT_IN                      ;检查完进行一个时间数值的调整
96     WRONG_INPUT_1:
97         LEA DX,WRONG_VALUE
98         MOV AH,09H
99         INT 21H
100        JMP INPUT                        ;有错误需要重新输入
```

```
101  WRONG_INPUT_2:
102      LEA DX,WRONG_NUMBER
103      MOV AH,09H
104      INT 21H
105      JMP INPUT                ;有错误需要重新输入
106  INIT_IN:
107      LEA SI,TIME_IN+2        ;时间ASCII码转化为压缩BCD码的初始化
108      LEA DI,TIME_CHG
109      MOV BL,03H              ;时、分、秒各一次
110      MOV CL,04H
111  TRANS_IN:                    ;结果是TIME_CHG中分别放着
112                                ;时、分、秒的BCD码
113      MOV AL,[SI]             ;移入相应时间量度的数值
114      MOV AH,[SI+1]
115      SHL AH,CL                ;左移4位，把ASCII的30挪掉
116                                ;让时间的两位挨在一起
117      ROL AX,CL                ;右移4位，把挨在一起的高低位挪进AL
118      MOV [DI],AL
119      DEC BL                    ;计算换算次数，判断是否换算完毕
120      CMP BL,00
121      JE GETTIME               ;若换算完毕，进入下一部分操作
122      ADD SI,0003H             ;指向下一时间单位的缓冲区存储空间
123      INC DI                    ;时间储存区加1
124      JMP TRANS_IN             ;循环进行换算操作
125  GETTIME:
126      LEA DI,TIME_CHG        ;将时间的压缩BCD码赋给相应的寄存器
127      MOV CH,[DI]
128      MOV DH,[DI+1]
129      MOV DL,[DI+2]
130      JMP PRINT
131  SECPLUS:
132      CALL DELAY              ;执行延时1秒子程序
133      MOV AL,DL                ;秒+1，并进行压缩BCD码加法调整，
134                                ;计算完毕放回相应寄存器
135      ADD AL,01
136      DAA
137      MOV DL,AL
138  SECOND:
139      CMP DL,60H              ;判断秒值与60的大小关系，
140                                ;小于则跳转执行输出初始化
141      JB INIT_OUT              ;进行秒的进制调整，无符号小于则跳转
```



```
142      MOV DL,00          ;秒值置0
143      MOV AL,DH          ;分+1，并进行压缩BCD码加法调整，
144                          ;计算完毕放回相应寄存器
145      ADD AL,01
146      DAA
147      MOV DH,AL
148  MINUTE:
149      CMP DH,60H         ;判断分值与60的大小关系，
150                          ;小于则跳转执行输出初始化
151      JB INIT_OUT        ;进行分的进制调整
152      MOV DH,00          ;分值置0
153      MOV AL,CH          ;时+1，并进行压缩BCD码加法调整，
154                          ;计算完毕放回相应寄存器
155      ADD AL,01
156      DAA
157      MOV CH,AL
158  HOUR:
159      CMP CH,24H         ;判断时值与24的大小关系，
160                          ;小于则跳转执行输出初始化
161      JB INIT_OUT        ;进行时的进制调整
162      MOV CH,00          ;时值置0
163  INIT_OUT:
164      LEA DI,TIME_CHG    ;时间输出的初始化
165      LEA SI,TIME_IN+2
166      MOV [DI],CH        ;将各时间单位数值对应存放在
167                          ;TIME_CHG中，以备码制转换
168      MOV [DI+1],DH
169      MOV [DI+2],DL
170      MOV BL,03H
171      MOV CL,04H
172  TRANS_OUT:
173      MOV AL,[DI]        ;移入相应时间量度的压缩BCD码
174      MOV AH,33H         ;通过移位操作，可以得到对应的ASCII码
175      ROR AX,CL
176      ROR AH,CL
177      MOV [SI],AL        ;将转换的ASCII码存入缓冲区
178      MOV [SI+1],AH
179      DEC BL             ;计算换算次数，判断是否换算完毕
180      CMP BL,00
181      JE PRINT           ;若换算完毕，进入下一部分操作
182      ADD SI,0003H       ;指向下一时间单位的缓冲区存储空间
```

```
183         INC DI
184         JMP TRANS_OUT           ;循环进行换算操作
185 PRINT:
186         MOV BX,DX               ;保护计算得到的分、秒信息
187         LEA DX,TIME_IN+2        ;输出时间
188         MOV AH,09H
189         INT 21H
190 IS_PUSH:
191         MOV AH,06               ;判断是否有按键按下
192         MOV DL,0FFH
193         INT 21H
194         JNZ LAST                ;若有按键按下，执行LAST
195         MOV DX,BX               ;若无按键按下，则恢复分、秒信息，
196                                 ;继续进行计时
197         JMP SECPLUS
198 LAST:
199         MOV AH,4CH              ;返回dos，结束程序
200         INT 21H
201         HLT                     ;处理器进入暂停状态
202 DELAY PROC NEAR                ;延时1秒子程序
203         PUSH CX
204         PUSH AX
205         PUSH DX
206         PUSH BX
207         MOV AH,2CH              ;读取当前时间，
208                                 ;CH：小时；CL：分；DH：秒；
209                                 ;DL：百分之一秒
210         INT 21H
211         MOV BH,DH               ;将当前秒数存在BH中
212         MOV AH,2DH              ;设置时间
213         INT 21H
214 READ:
215         MOV AH,2CH
216         INT 21H
217         CMP DH,BH               ;比较两个时间的秒数，若相等则继续循环
218         JE READ                 ;否则则经过了一秒
219         POP BX
220         POP DX
221         POP AX
222         POP CX
223         RET
```

224

225

226

DELAY ENDP
CODE ENDS
END START

三. 实验效果图

实验中错误提示如图 2，正确显示结果如图 3。

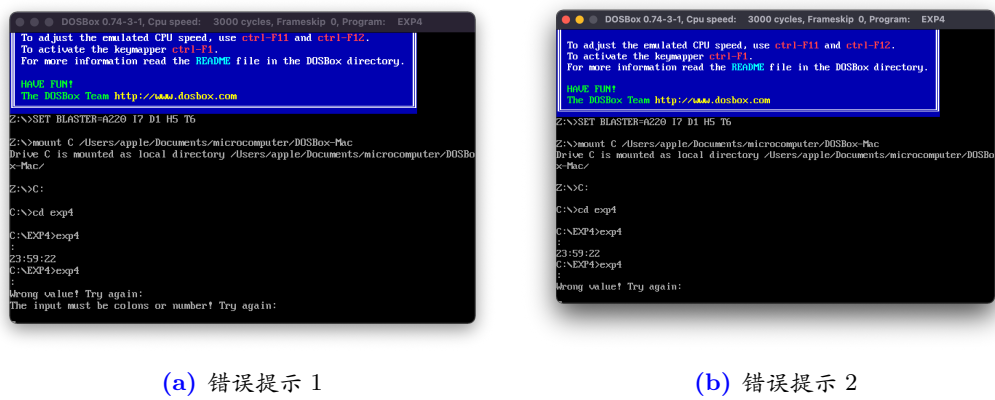


图 2

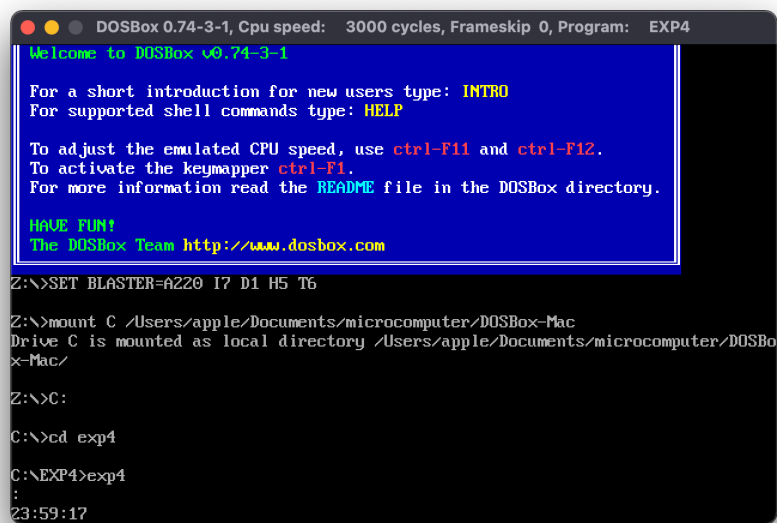


图 3: 正确显示结果

四. 思考题

表 1: 思考

实验思考题

时钟程序中存在时间误差吗？若有误差，其来源在何处？如何进行误差校正？

- 来源：**
1. 计算机设备自身的运行速度；
 2. 第一次计时存在一些初始化和检测操作，这些操作在之后的计时中不需要，因此第一次计时和之后的计时之间有一定误差；
 3. 延时 1 秒子程序无法十分精确地确定 1 秒钟，造成误差。

- 校正：**
1. 在延时 1 秒子程序中，调整循环程序代码执行的次数和循环程序中循环的次数，实现控制延时 1 秒的功能；
 2. 可以通过调用系统时间进行校正。

五. 附加任务描述及完成

(一) 附加任务 1

在同一行的相同位置显示更新的计时时间，不换行。

delay

```
1  DATA SEGMENT
2  :
3  :
4  PRINT_ETR      DB 0DH,0AH,'$'          ;设置一回车输出串，
5                                          ;在“:”显示后输出，换行缩进
6  DATA ENDS
7  :
8  :
9  MOV DL,':'      ;显示时间输入提示符号“:”
10 MOV AH,02H
11 INT 21H
12 LEA DX,PRINT_ETR
13 MOV AH,09H
14 INT 21H
15 :
16 :
```

分析 1: 换行显示问题

汇编里回车是回到本行首位，换行是到下一行同样位置。回车与换行的不同之处在于，换行会移到下一行，但‘回车 + 换行’却可以回到本行重新输出，这样，在待输出的字符串后面加上回车、换行符，即可实现同一行更新时间。

(二) 附加任务 2

输入时间初值时，会检查是否有错、提示错误信息，并可重新输入时间初值。错误提示信息可以分两种：

1. 输入的时间初值是错误的字符，即不是数字和冒号；
2. 输入的时间值是错误的，即“时”大于等于 24，“分”和“秒”大于等于 60。

delay

```
1  DATA SEGMENT
2  :
3  :
4  WRONG_VALUE DB 'Wrong value! Try again','$'
                    ;错误类型1
5  WRONG_NUMBER DB 'The input must be colons or number! Try again','$'
                    ;错误类型2
6  DATA ENDS
7  :
8  :
9  LENTH_TEST:                ;长度测试
10     MOV AL,TIME_IN+1        ;实际键入字数
11     CMP AL,08H              ;和8比较，不相等跳转到 WRONG_INPUT
12     JNE WRONG_INPUT_2
13  FORMAT_TEST:              ;格式测试
14     MOV AL,TIME_IN+4        ;格式: XX:XX:XX
15     CMP AL,':'
16     JNE WRONG_INPUT_2
17     MOV AL,TIME_IN+7
18     CMP AL,':'
19     JNE WRONG_INPUT_2
20  INIT_LOW_LIMIT_TEST:
21     LEA SI,TIME_IN+2
22     MOV CL,03H              ;循环3次
23  LOW_LIMIT_TEST:
24     MOV AL,[SI]
25     MOV AH,[SI+1]
```

```
26      CMP AL,30H                ;字符‘0’的ASCII码是30，不能小于0
27      JB WRONG_INPUT_2          ;无符号小于则跳转
28      CMP AH,30H                ;字符‘0’的ASCII码是30，不能小于0
29      JB WRONG_INPUT_2          ;无符号小于则跳转
30      ADD SI,0003H
31      LOOP LOW_LIMIT_TEST
32  INIT_HIGH_LIMIT_TEST:
33      LEA SI,TIME_IN+2
34      MOV CL,03H
35  HIGH_LIMIT_TEST:
36      MOV AL,[SI]
37      MOV AH,[SI+1]
38      CMP AL,39H                ;字符‘9’的ASCII码是39，
39                                ;不能超过9，先得是个数字
40      JA WRONG_INPUT_2          ;无符号大于则跳转
41      CMP AH,39H
42      JA WRONG_INPUT_2
43      ADD SI,0003H
44      LOOP HIGH_LIMIT_TEST
45  HOUR_HIGH_TEST:
46      MOV AL,TIME_IN+2
47      CMP AL,32H                ;字符‘2’的ASCII码是32，
48                                ;‘时’第一位不能超过2，
49                                ;‘分’和‘秒’同理
50      JA WRONG_INPUT_1
51  HOUR_FULL_HIGH_TEST:
52      MOV AL,TIME_IN+3
53      MOV AH,TIME_IN+2
54      CMP AX,3233H              ;字符‘0204’的ASCII码是3233
55      JA WRONG_INPUT_1
56  MINUTE_TEST:
57      MOV AL,TIME_IN+5
58      CMP AL,35H
59      JA WRONG_INPUT_1
60  SECOND_TEST:
61      MOV AL,TIME_IN+8
62      CMP AL,35H
63      JA WRONG_INPUT_1
64      JMP INIT_IN               ;检查完进行一个时间数值的调整
65  WRONG_INPUT_1:
66      LEA DX,WRONG_VALUE
```

```
67      MOV AH,09H
68      INT 21H
69      JMP INPUT          ;有错误需要重新输入
70  WRONG_INPUT_2:
71      LEA DX,WRONG_NUMBER
72      MOV AH,09H
73      INT 21H
74      JMP INPUT          ;有错误需要重新输入
75      :
76      :
```

分析 2: 输入检查

输入检查原理是将相应字符 ASCII 码与数字比较，确保格式以及数值正确。

注意 1: 误区

检查数值过程需要逐位比较，例如在判断时候原先误将代码写为两位一起判断，如下

code

```
1      MOV AL,[SI]
2      MOV AH,[SI+1]
3      CMP AX,3030H
4      JB  WRONG_INPUT_2
```

这种情况下，如果非法字符的组合 ASCII 码大于 3030H 也可以通过检查，错误！

(三) 附加任务 3

延时一秒用 DOS 系统功能调用实现。

实验代码如下：

code

```
1  DELAY PROC NEAR          ;延时1秒子程序
2      PUSH CX
3      PUSH AX
4      PUSH DX
5      PUSH BX
6      MOV AH,2CH          ;读取当前时间，
7                          ;CH：小时；CL：分；DH：秒；
8                          ;DL：百分之一秒
9      INT 21H
```

```
10      MOV BH,DH          ;将当前秒数存在BH中
11      MOV AH,2DH        ;设置时间
12      INT 21H
13  READ:
14      MOV AH,2CH
15      INT 21H
16      CMP DH,BH          ;比较两个时间的秒数,
17                          ;若相等则继续循环
18      JE READ            ;否则则经过了一秒
19      POP BX
20      POP DX
21      POP AX
22      POP CX
23      RET
24  DELAY ENDP
```

分析 3: Delay 程序说明

起初按照教材的方法设置延时过程，但是程序运行后时间变化非常快，更正后：判断循环时仅需与一秒进行比较即可，经过改正，成功实现了 DOS 调用。

六. 实验总结

实验总结已随文附在“注意”、“思考”、“分析”中。

参考文献

- [1] 李继灿. 新编 16/32 位微型计算机原理及应用（第五版）[M]. 5 版. 北京: 清华大学出版社, 2013.
- [2] 微机教学组. 《微计算机实验讲义》[A]. 南京: 东南大学, 2015.