中南大学

计算机原理与汇编语言

课程设计报告

题 目

学生姓名

指导教师

学 院

专业班级

一、问题描述

设计一个数字钟,在屏幕上显示分：秒（mm:ss）。按下非空格键开始计时，并显示 00：00，每过 1 秒，ss 增 1，到 60 秒 mm 增 1，到 60 分就是 1 小时。经过 1 小时后又回到 00：00 重新计数。当按下空格键时，程序返回 DOS，数字钟消失。

二、程序分析与设计

2.1 难点

①在屏幕上显示分：秒

②按下非空格开始计时

③每过 1 秒，ss 增 1，到 60 秒 mm 增 1

④按下空格返回DOS

⑤延时一秒

2.2设计组成框图

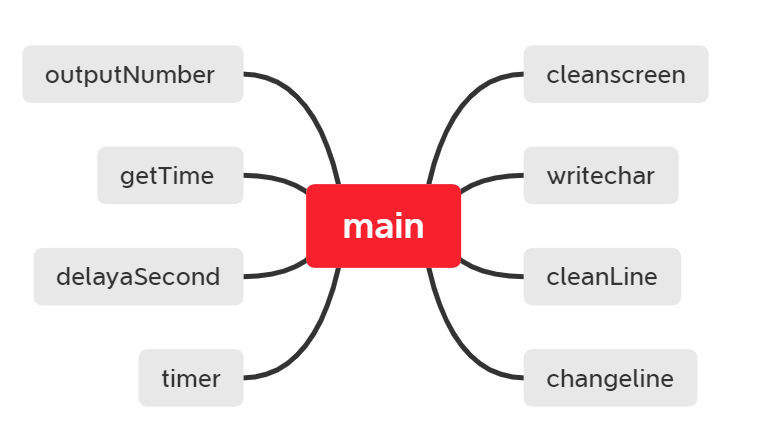


图2.2 程序设计组成框图

三、程序模块功能与流程

3.1 模块功能

1. main主程序模块

主程序，调用各个子程序实现完整地功能

2.getTime模块

刷新当前时间

3. cleanscreen模块

清屏

4.cleanLine模块

清除光标所在行

5. writechar模块

选定屏幕颜色

6. outputnumber模块

把ax中的数字输出出来。

7. delayaSecond模块

延时一秒

8. timer模块

显示当前时间

3.2 流程图

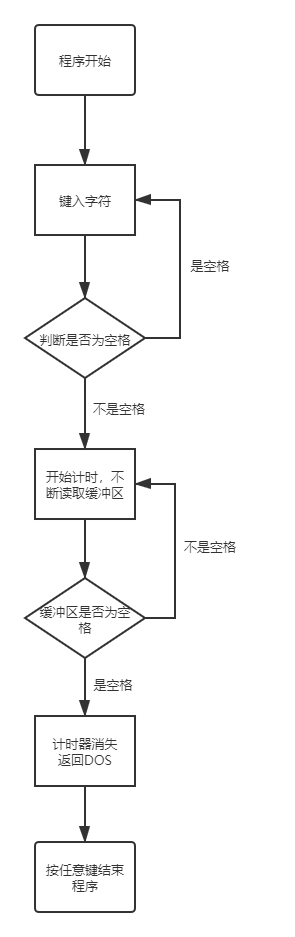


图3.2.1 程序流程图

四、调试与运行

4.1 调试方法

开始时按下空格观察是否会进行计时

程序运行时观察是否会满60进1

结束程序时按下非空格观察是否会结束运行

4.2 遇到的问题及解决措施

问题：①在结束程序时需要按下空格让程序结束，同时又在打印当前计时情况，不清楚如何加入中断或者是如何编写汇编多进程

②如何清除缓冲区，让后续输入缓冲区的空格可以被扫描到

③如何判断是否是非空格输入从而开始程序

④如何延时一秒后让second加1

⑤如何记录之前的时间

解决措施：①使用int 16h的11h功能，查看缓冲区有无字符且不等待，无字符，跳转，有字符，顺序执行

②使用int 21h的0ch功能，并给ah赋06h功能，删除缓冲区

③使用int 16h的10h功能，比较输入字符和空格的ascll码

④查找公式进行计算，计算机主频：x （Hz） 2.3GHz，一条LOOP语句执行始终周期数：y ，所需要延时的时间：z （s） 1s，需要执行的语句数：a 1条，z=y\*(1/x)\*a =2.3\*10^7

⑤使用变量second和minute，保存每一次的运算结果

4.3 运行结果

4.3.1 测试数据

①调快时间，观察是否有进位

②是否按非空格会结束

4.3.2 测试结果

①second = 59时，下一秒second == 0，minute+1

②不会结束

五、心得体会

经过这次实验，对汇编的掌握更深了，在整个实验中，将《汇编语言程序设计》重新学习了一次，从中学到了很多之前不曾掌握的命令，收获颇丰，关于数字计时转换为电子钟的问题，我也有了一些思考，使用七条语句来表示七段数码管，根据数字的不同而打印不同的语句，从而呈现数字钟的形式。

虽然之后可能汇编的使用次数会大大降低，但汇编知识的掌握可以让我更加清晰明了的去理解其他的编程语言的核心思想，站在这个角度来看，本次汇编实验对我的编程能力有着巨大的提升和促进作用。

六、源程序

cleanscreen macro clor ;相当于清屏 ，指定颜色

push cx

push si

push di

mov cx,1

lea si, blank ;将si中存入blank

writechar 0,0,clor,2000 ;打印blank，blank为空，相当于用空格代替原有字符

;设置字符属性，光标位置，从上至下清屏

pop di

pop si

pop cx

endm

writechar macro arra,row,color,cc ;x

local LP ;定义局部变量LP

mov dl,arra ;第几列

LP: mov dh ,row ;第几行

mov ah,02 ;设置位置

mov bh, 0

int 10h

mov al,[si]

mov ah,09h

mov bl,color ;颜色

push cx

mov cx,cc

int 10h

add si,1

add dl,1

pop cx

loop LP

endm

.model small

.stack

.data

; ----------------------------------------------------------打印学生个人信息

MESS1 DB 0DH,0AH,0DH,0AH,0DH,0AH

DB 9,9,9,'-------------------------------',0DH,0AH ;回车换行

DB 9,9,9,'| Welcome to Clock System |',0DH,0AH

DB 9,9,9,'| |',0DH,0AH

DB 9,9,9,'| name: South\_yang |',0DH,0AH

DB 9,9,9,'| class: CS1903 |',0DH,0AH

DB 9,9,9,'| Number: 8208190213 |',0DH,0AH

DB 9,9,9,'------------------------------- ',0DH,0AH,'$'

MSG1 DB 0DH,0AH,9,9,9,'press any key to begin.....',0DH,0AH,'$' ;提示信息

; ----------------------------------------------------------打印菜单

MESS2 DB 0DH,0AH,0DH,0AH,0DH,0AH

DB 9,9,9,'-------------------------------',0DH,0AH

DB 9,9,9,'| System function |',0DH,0AH

DB 9,9,9,'| 1- Start the clock |',0DH,0AH

DB 9,9,9,'------------------------------- ',0DH,0AH,'$'

MSG2 DB 0DH,0AH,9,9,' PRESS any key except the space to use function 1 ',0DH,0AH,'$' ;提示信息

; ----------------------------------------------------------变量数据

MSG DB 0DH,0AH,9,9,9,'press any key to continue.....',0DH,0AH,'$' ;提示信息

setTime DB 'Clock has set!',0DH,0AH

minute DB 0 ;分钟数

second DB 0 ;秒钟数

blank DB 25 dup(' ') ;空格,用于清屏

s1 DB 'The time is: $'

s2 DB 'The timing is over! $'

.code

.startup

; ----------------------------------------------------------程序开始

MOV AH,09H

LEA DX, MESS1

INT 21H

MOV AH,09H

LEA DX, MSG1

INT 21H

mov ah,07h

int 21H

;第一个界面，显示提示信息MESS1和press any key to begin.....

START0:

cleanscreen 3fh;调用宏

mov ah,09h

lea DX,MESS2

INT 21H

LEA DX, MSG2

INT 21H

mov ah,10h

int 16H ;将输入字符的ascll码存在al中

cmp al,32 ;判断输入是否是空格，不是，返回start0，是，继续运行

jz START0

;第二个界面，显示提示信息MESS2和PRESS any key except the space to use function 1

;----------------------------------------------- 开始计时

START1:

cleanscreen 3fh ;清屏并选择界面颜色

push dx

push bx

mov dl,10 ;选定光标位置

mov dh ,5

mov ah,02

mov bh, 0

int 10h

;10h ah=2

;dl=列 dh=行

;置光标到指定位置

lea dx, setTime ;打印setTime

mov ah,09h

int 21h

pop bx

pop dx

call timer

MOV AX,4C00H ;返回DOS

int 21h

; -------------------------------

;用于清除一行的字符

cleanLine proc

push ax

push bx

push cx

push dx

mov cx,75

mov ah,2h

mov dl,0dh

int 21h

clean:

mov dl,' '

int 21h

loop clean

mov dl,0dh ;回车，回到行首，等待下一步输出

int 21h

pop dx

pop cx

pop bx

pop ax

ret

cleanLine endp

;用于显示一个回车换行

changeline proc

push ax

push dx

mov ah,2h

mov dl,0dh

int 21h

mov dl,0ah

int 21h

pop dx

pop ax

ret

changeline endp

outputNumber proc;输出ax中的一个数字

push ax

push bx

push cx

push dx

mov cx,0

mov bx,10

p0:

mov dx,0

div bx

push dx ;将时间从低到高依次入栈

inc cx

cmp ax,0

jnz p0

mov ah,2

cmp cx,2

jae p1

mov dx,0

push dx

inc cx

p1:

pop dx

add dx,30h ;按高低位输出时间

int 21h

loop p1

pop dx

pop cx

pop bx

pop ax

ret

outputNumber endp

getTime proc;用于刷新时间

push ax

mov al,minute ;获取当前分钟数并打印

mov ah,0

g1: call outputNumber

mov ah,2h

mov dl,':'

int 21h

mov ah,2h

mov dl,' '

int 21h

mov ah,0

mov al,second ;获取当前秒钟数并打印

cbw ;位扩展

g2: call outputNumber

inc second ;时间加，若为60则进位

cmp second,60

jne c3

c1: mov second,0

inc minute

cmp minute,60

jne c3

c2: mov second,0

mov minute,0

c3:

pop ax

ret

getTime endp

delayaSecond proc;用于延时一秒

push bx

push cx

mov bx,100 ;做22700000次循环来达到延时1s的目的

back1:

mov cx,227000

back2:

loop back2

dec bx

jnz back1

pop cx

pop bx

ret

;计算机主频：x （Hz） 2.3GHz

;一条LOOP语句执行始终周期数：y

;所需要延时的时间：z （s） 1s

;需要执行的语句数：a 1条

;z=y\*(1/x)\*a 2.3\*10^7

delayaSecond endp

timer proc

time: ;用于每过一秒种刷新一次显示的时间

call cleanLine;清除屏幕上光标所在这一行上的字符串

mov bx,0 ;并把光标移动到行首

mov dl,10

mov dh ,7

mov ah,02

mov bh, 0

int 10h

mov ah,9h

lea dx,s1

int 21h

call getTime ;打印当前时间

mov ah,11h ;查看缓冲区有无字符等待

int 16h

jz nokeywaiting ;无字符，跳转；有字符，将ascll码赋给bl

mov bh,0

mov bl,al

mov ah,0ch ;ascll码保存后清空缓存区，确保下次输入缓存区的字符可以被读取

mov al,06h ;01,06,07,08，只有06可以用

int 21h

nokeywaiting:

cmp bx,32 ;如果按下空格，则跳转next

je next

call delayaSecond

mov ah,2h

mov dl,0dh

int 21h

jmp time

next: ;结束

call cleanLine;清除屏幕上光标所在这一行上的字符串，并把光标移动到行首

mov bx,0 ;移动光标位置

mov dl,10

mov dh ,7

mov ah,02

mov bh, 0

int 10h

mov ah,9h

lea dx,s2 ;打印s2

int 21h

mov ah,07h ;输入任何字符结束程序

int 21H

finish:

ret ;time子程序运行结束并返回

timer endp

end