**计算机原理与汇编语言**

**课程设计**

**题目名称：查找字符串中的指定字符**

**姓名：谭哲文**

**学号：8202191123**

**专业：计算机科学与技术**

**班级：计科2105**

**指导教师：贺建飚**

**编写日期：2024.6.11**

***目录：***

1. **问题描述----------------------------------------------------------------1**
2. **系统设计----------------------------------------------------------------1**

提示语-------------------------------------------------------------1

子串查找-----------------------------------------------------------1

结果显示-----------------------------------------------------------2

1. **源代码清单--------------------------------------------------------------2**

Main----------------------------------------------------------------2

Print\_str-----------------------------------------------------------3

input---------------------------------------------------------------3

Find\_str------------------------------------------------------------3

Print\_color\_char----------------------------------------------------4

Change\_color--------------------------------------------------------4

Turn\_decimal--------------------------------------------------------5

Show\_result---------------------------------------------------------5

1. **运行结果测试与分析------------------------------------------------------6**
2. **结论与心得--------------------------------------------------------------7**

**6， 附录-------------------------------------------------------------------8**

# 问题描述：

**查找字符串中的指定字符**

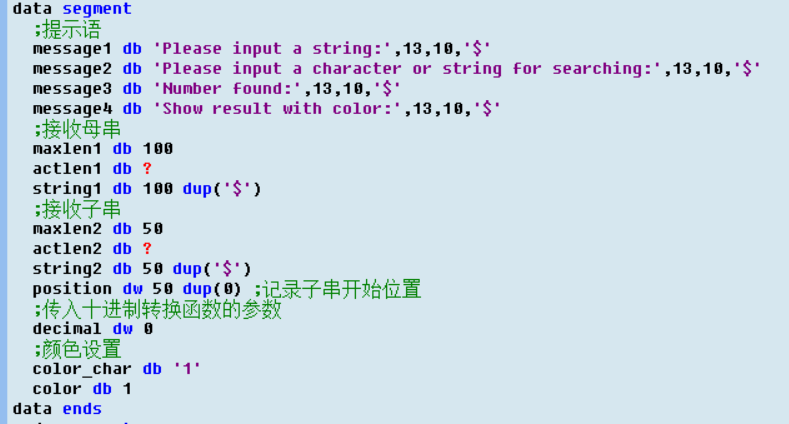
基本功能：

1 做一个操作界面，提示操作：输入一串字符串、输入所查找的字符或字符串等；

2 显示出查找到的数目；

3 用不同颜色或闪烁标示出所找到的字符或字符串；

# 系统设计：



**图1 程序数据段**

* **提示语**

首先，本程序需要接收两个用户输入，一个是母字符串，一个是需要查找的子字符串，这里我们通过字符串打印创建一个简单的带有提示的菜单。事先在数据段中存储好提示字符串（如图1中message1,2,3,4），在需要时通过调用DOS 09h号中断显示相应内容。

* **子串查找**

本问题的核心是字符串的查找，这里采用朴素的字符串查找方法。在数据段中为母串，子串预留位置，通过调用DOS 0Ah号中断接收输入，实际字符串会被存入string1，string2中，并且字符串实际长度被存入actlen1和actlen2中，这在之后也会用到。考虑到之后需要在母串中显示查找到的子串，所以还需要记录子串在母串中的位置，这里用position数组来记录。另外用一个字节decimal记录查找到的子串数目，之后需要输出。

查找依靠两层循环，外层循环枚举查找起点，内层循环比较字符，母串从该起点起，子串从头起，依次向后逐一对比。如果出现不匹配则内层循环退出， 外层循环进入下一轮；如果匹配了完整的子串，则说明在母串中成功找到了一个子串，在position中记录此次查找的起点，子串数目+1。下一次的查找起点需跳过子串长度个字符。直至从所有查找起点均查找过，外层循环退出。

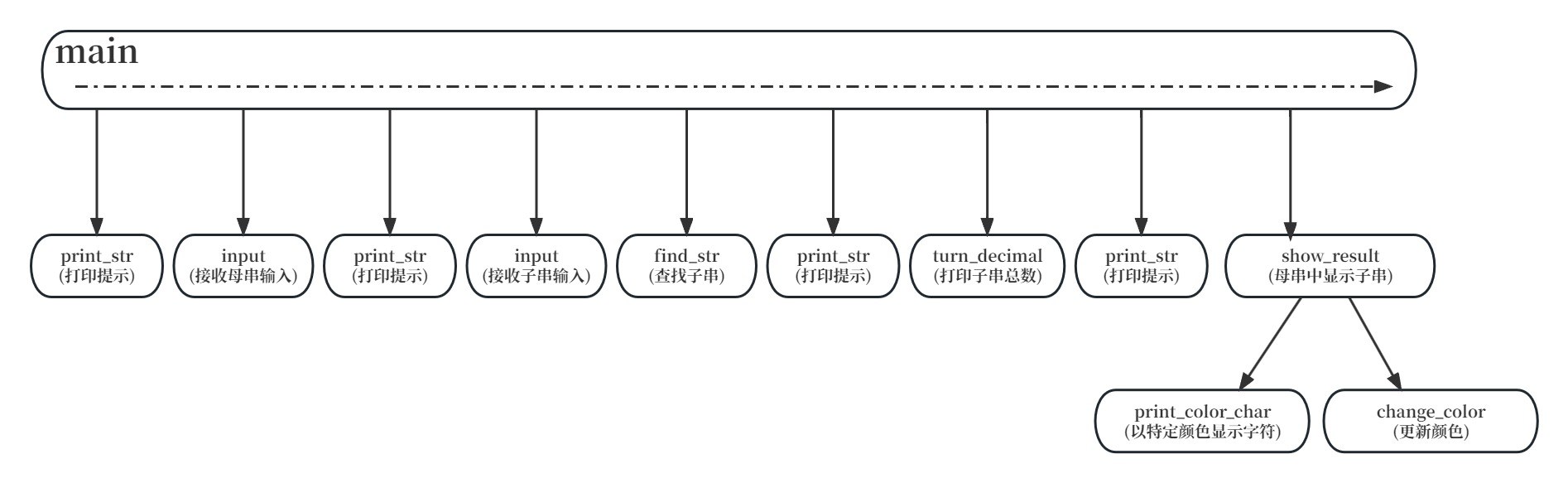
* **结果显示**

查找结束后，position数组中应该记录了所有子串在母串中的起始位置，decimal中应该记录了查找到的字串数目。我们专门编写了一个函数用来设置字符颜色，color\_char用来存放需要变色的字符，即传入该函数的参数。在该函数中首先会调用BIOS 09h号中断以特定样式显示color\_char中字符，之后依次调用BIOS 03h,02h中断将光标向后移动一格。为了达到更好的视觉效果，我令相邻两个子串采用不同的颜色。具体地，共10种颜色，每连续10个子串使用10个不同颜色，之后颜色循环重复。数据段中的color就是用来实现这种模式的，每用颜色标注一个子串，就令color增1，然后将color对10取模的值映射到10个不同的颜色。

为了显示子串个数，需要有一个进制转换过程，这里专门写了一个函数。decimal记录的是需要转换为十进制的数值，作为参数传入该函数。进制转换采用除法转换法。具体地，将需要转换的二进制数作为被除数（ax），要转换的进制（10）作为除数（dl），然后不断地进行除法，每做一次除法后，需将余数压入栈记录，然后用商作为下一轮除法的被除数（cbw）

直至一次除法后商为0，结束除法过程。现在，栈中的每一个值代表着转换出的十进制数的一位（0~9），只要依次弹出栈顶值即可正确显示该十进制数。当然弹出过程中还需要将0~9数值转换为0~9字符。

# 源代码清单：

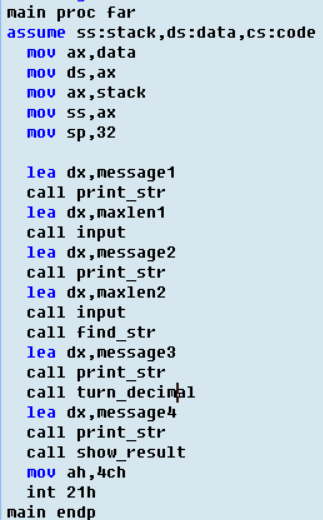


**图2 函数调用关系图**

**本程序包含如下函数:**

* Main

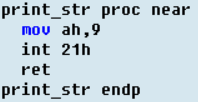
主函数，调用其他函数完成功能。



**图3 main**

* print\_str

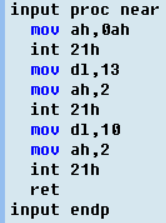
打印字符串函数，用来显示提示语。调用前需将字符串首偏移地址存入dx(数据段message1~4)



**图4 print\_str函数**

* input

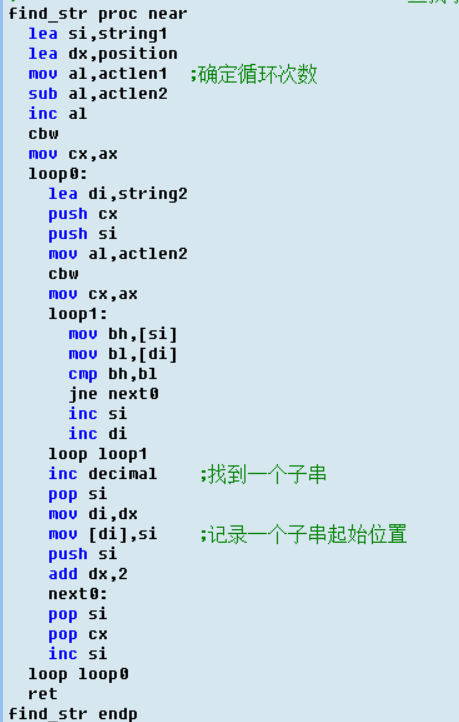
接收用户输入（母串，子串）函数，调用前需将接收输入区域的首偏移地址存入dx（数据段maxlen1,maxlen2）



**图5 input函数**

* find\_str

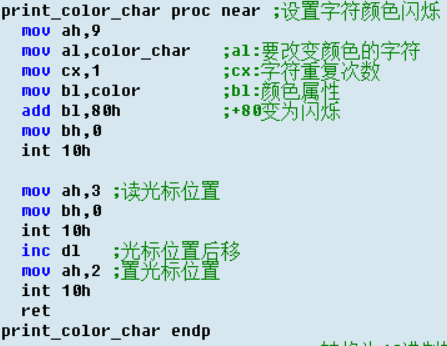
查找子串函数。实现了朴素的子串查找算法。



**图6 find\_str函数**

* print\_color\_char

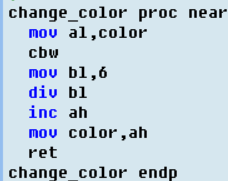
以特定颜色显示字符，调用前需将要变色的字符放入color\_char



**图7 print\_color\_char函数**

* change\_color

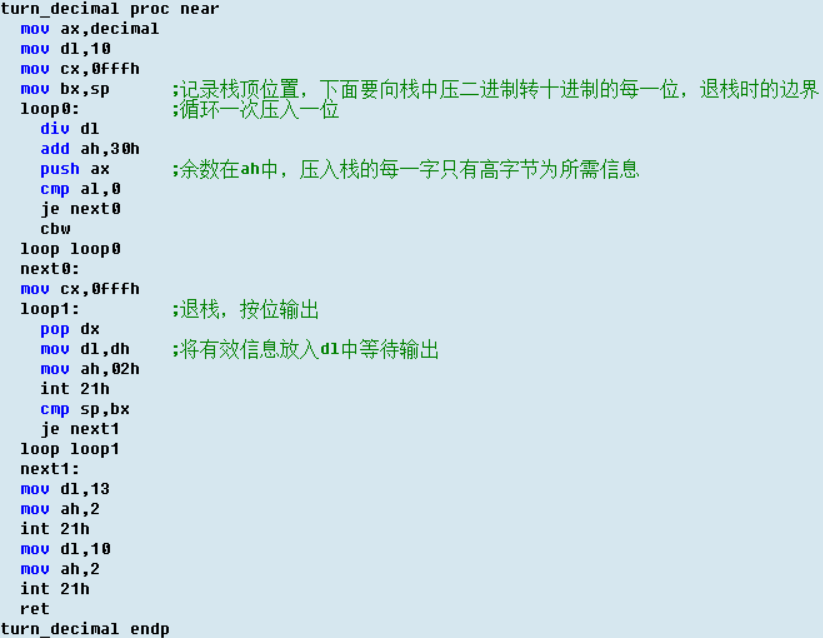
刷新颜色，每显示完一个子串后调用一次



**图8 change\_color函数**

* turn\_decimal

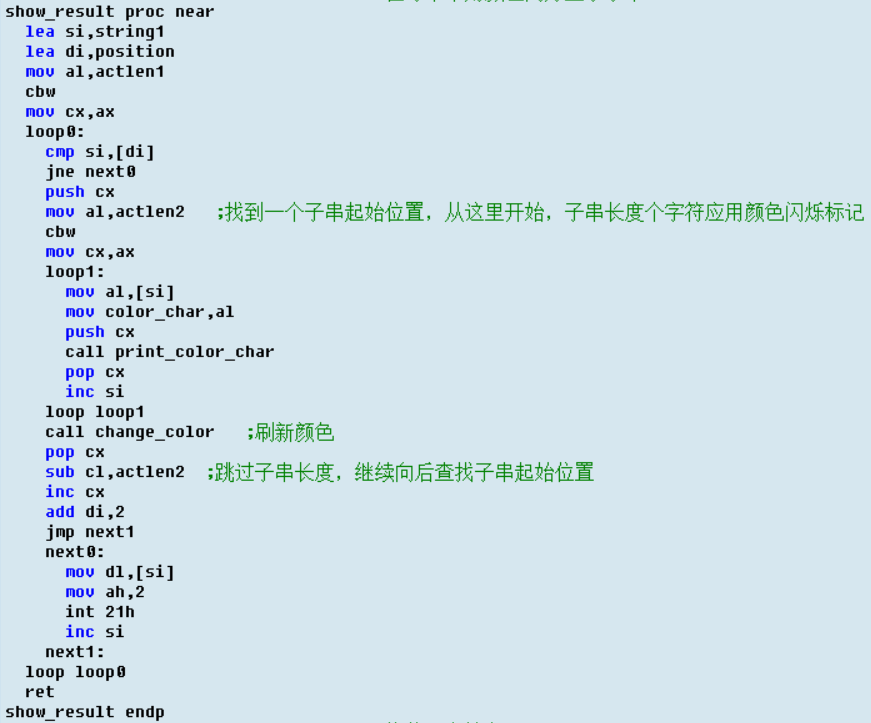
二进制转换为十进制。调用前需将要转换的二进制值存入decimal。



**图9 turn\_decimal函数**

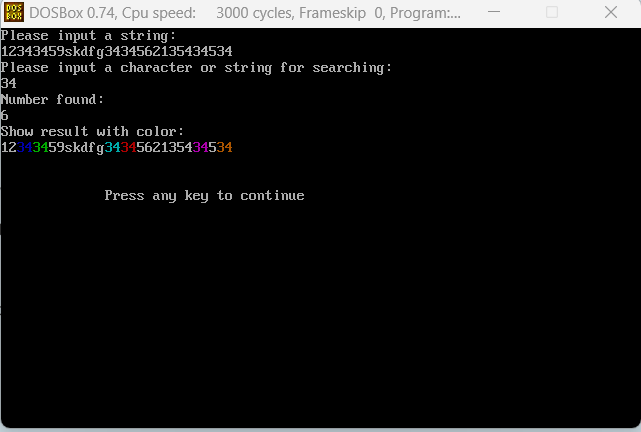
* show\_result

显示结果。同时遍历母串（string1）和position数组，每遍历到一个子串的起始，开始用颜色标注接下来的子串长度（actlen2）个字符。将每一个子串字符存入color\_char然后调用print\_color\_char，每显示完一个子串调用change\_color以刷新颜色。

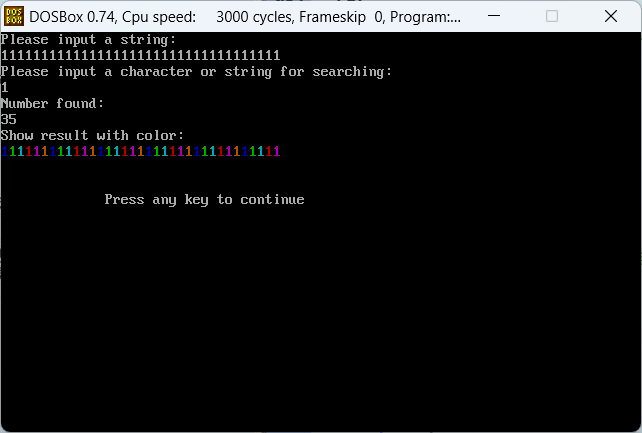


**图10 show\_result函数**

# 运行结果测试与分析：



**图11 运行结果a**



**图12 运行结果b**

可以看到正确查找出了子串，以十进制形式显示字串个数，用颜色在母串中标记出子串，并且相邻子串用不同颜色凸显，每6个颜色一重复。上面的子串是闪烁的！

# 结论与心得：

本题目在所有题目中相对简单，所以该设计100%独立完成，遇到的问题也相对较少。

调试中遇到的一个问题是调用BIOS 09h号中断以特定颜色显示字符时发现显示完第一个字符后要显示第二个字符时发现光标并未后移，而是将上一个字符覆盖。通过查阅资料发现这一BIOS中断并没有自动后移光标的功能，所以每次调用完这一中断需要手动后移光标。于是我通过BIOS 02h,03h号中断进行手动移动光标，解决了这一问题。

由于时间关系，本实验中的字符串查找采用最为朴素的方法，其实我更想尝试实现一些更为灵活高效的字符串查找算法，例如KMP算法（奈何要赶着期末复习…）。

总的来说，这次实验是对上一学期汇编课程的复习，实验中用到了一些之前没用过的新的有趣的中断，让我对汇编语言有了更牢固的掌握。

# 附录：

**源码如下：**

stack segment

dw 16 dup(?)

stack ends

data segment

;提示语

message1 db 'Please input a string:',13,10,'$'

message2 db 'Please input a character or string for searching:',13,10,'$'

message3 db 'Number found:',13,10,'$'

message4 db 'Show result with color:',13,10,'$'

;接收母串

maxlen1 db 100

actlen1 db ?

string1 db 100 dup('$')

;接收子串

maxlen2 db 50

actlen2 db ?

string2 db 50 dup('$')

position dw 50 dup(0) ;记录子串开始位置

;传入十进制转换函数的参数

decimal dw 0

;颜色设置

color\_char db '1'

color db 1

data ends

code segment

main proc far

assume ss:stack,ds:data,cs:code

mov ax,data

mov ds,ax

mov ax,stack

mov ss,ax

mov sp,32

lea dx,message1

call print\_str

lea dx,maxlen1

call input

lea dx,message2

call print\_str

lea dx,maxlen2

call input

call find\_str

lea dx,message3

call print\_str

call turn\_decimal

lea dx,message4

call print\_str

call show\_result

mov ah,4ch

int 21h

main endp

;-------------------------------------打印字符串

print\_str proc near

mov ah,9

int 21h

ret

print\_str endp

;-------------------------------------- 查找子串

find\_str proc near

lea si,string1

lea dx,position

mov al,actlen1 ;确定循环次数

sub al,actlen2

inc al

cbw

mov cx,ax

loop0:

lea di,string2

push cx

push si

mov al,actlen2

cbw

mov cx,ax

loop1:

mov bh,[si]

mov bl,[di]

cmp bh,bl

jne next0

inc si

inc di

loop loop1

inc decimal ;找到一个子串

pop si

mov di,dx

mov [di],si ;记录一个子串起始位置

push si

add dx,2

next0:

pop si

pop cx

inc si

loop loop0

ret

find\_str endp

;------------------------------------ 在母串中用颜色闪烁显示子串

show\_result proc near

lea si,string1

lea di,position

mov al,actlen1

cbw

mov cx,ax

loop0:

cmp si,[di]

jne next0

push cx

mov al,actlen2 ;找到一个子串起始位置，从这里开始，子串长度个字符应用颜色闪烁标记

cbw

mov cx,ax

loop1:

mov al,[si]

mov color\_char,al

push cx

call print\_color\_char

pop cx

inc si

loop loop1

call change\_color ;刷新颜色

pop cx

sub cl,actlen2 ;跳过子串长度，继续向后查找子串起始位置

inc cx

add di,2

jmp next1

next0:

mov dl,[si]

mov ah,2

int 21h

inc si

next1:

loop loop0

ret

show\_result endp

;----------------------------------- 接收用户输入

input proc near

mov ah,0ah

int 21h

mov dl,13

mov ah,2

int 21h

mov dl,10

mov ah,2

int 21h

ret

input endp

;---------------------------- 子串颜色改变，相邻两子串颜色不同，6个一循环

change\_color proc near

mov al,color

cbw

mov bl,6

div bl

inc ah

mov color,ah

ret

change\_color endp

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

print\_color\_char proc near ;设置字符颜色闪烁

mov ah,9

mov al,color\_char ;al:要改变颜色的字符

mov cx,1 ;cx:字符重复次数

mov bl,color ;bl:颜色属性

add bl,80h ;+80变为闪烁

mov bh,0

int 10h

mov ah,3 ;读光标位置

mov bh,0

int 10h

inc dl ;光标位置后移

mov ah,2 ;置光标位置

int 10h

ret

print\_color\_char endp

;-------------------------------转换为10进制输出

turn\_decimal proc near

mov ax,decimal

mov dl,10

mov cx,0fffh

mov bx,sp ;记录栈顶位置，下面要向栈中压二进制转十进制的每一位，退栈时的边界

loop0: ;循环一次压入一位

div dl

add ah,30h

push ax ;余数在ah中，压入栈的每一字只有高字节为所需信息

cmp al,0

je next0

cbw

loop loop0

next0:

mov cx,0fffh

loop1: ;退栈，按位输出

pop dx

mov dl,dh ;将有效信息放入dl中等待输出

mov ah,02h

int 21h

cmp sp,bx

je next1

loop loop1

next1:

mov dl,13

mov ah,2

int 21h

mov dl,10

mov ah,2

int 21h

ret

turn\_decimal endp

code ends

end main