实验 9 编写显示子程序

实验报告提交要求：提供代码正确性验证分析、实验结果截图、源代码。

1. 显示字符串

问题

显示字符串是现实工作中经常要用到的功能，应该编写一个通用的子程序来实现这个功能。我们应该提供灵活的调用接口，使调用者可以决定显示的位置(行、列)、内容和颜色。

子程序描述

名称：show_str
功能：在指定的位置，用指定的颜色，显示一个用 0 结束的字符串。
参数：(dh)=行号(取值范围 0~24)，(dl)=列号(取值范围 0~79)，
(cl)=颜色，ds:si指向字符串的首地址
返回：无
应用举例：在屏幕的 8 行 3 列，用绿色显示 data 段中的字符串。

```
assume cs:code
data segment
  db 'Welcome to masm!',0
data ends

code segment
start:  mov dh,8
        mov dl,3
        mov cl,2
        mov ax,data
        mov ds,ax
        mov si,0
        call show_str

        mov ax,4c00h
        int 21h
show_str:   :
            :
            :
code ends
end start
```

DOSbox有0行（不显示），列是从第1列开始

提示

(1) 子程序的入口参数是屏幕上的行号和列号，注意在子程序内部要将它们转化为显存中的地址，首先要分析一下屏幕上的行列位置和显存地址的对应关系；

(2) 注意保存子程序中用到的相关寄存器；

(3) 这个子程序的内部处理和显存的结构密切相关，但是向外提供了与显存结构无关的接口。通过调用这个子程序，进行字符串的显示时可以不必了解显存的结构，为编程提供了方便。在实验中，注意体会这种设计思想。

注：显存的结构请见实验 8。

3. 数值显示

问题

编程，将 data 段中的数据以十进制的形式显示出来。

```
data segment
  dw 123,12666,1,8,3,38
data ends
```

这些数据在内存中都是二进制信息，标记了数值的大小。要把它们显示到屏幕上，成为我们能够读懂的信息，需要进行信息的转化。比如，数值 12666，在机器中存储为二进制信息：0011000101111010B(317AH)，计算机可以理解它。而要在显示器上读到可以理解的数值 12666，我们看到的应该是一串字符：“12666”。由于显卡遵循的是 ASCII 编码，为了让我们能在显示器上看到这串字符，它在机器中应以 ASCII 码的形式存储为：31H、32H、36H、36H、36H(字符“0”～“9”对应的 ASCII 码为 30H~39H)。

通过上面的分析可以看到，在概念世界中，有一个抽象的数据 12666，它表示了一个数值的大小。在现实世界中它可以有多种表示形式，可以在电子机器中以高低电平(二进制)的形式存储，也可以在纸上、黑板上、屏幕上以人类的语言“12666”来书写。现在，我们面临的问题就是，要将同一抽象的数据，从一种表示形式转化为另一种表示形式。

可见，要将数据用十进制形式显示到屏幕上，要进行两步工作：

(1) 将用二进制信息存储的数据转变为十进制形式的字符串；
(2) 显示十进制形式的字符串。

第二步我们在本次实验的第一个子程序中已经实现，在这里只要调用一下 show_str 即可。我们来讨论第一步，因为将二进制信息转变为十进制形式的字符串也是经常要用到的功能，我们应该为它编写一个通用的子程序。

子程序描述

名称：dtoc
功能：将 word 型数据转变为表示十进制数的字符串，字符串以 0 为结尾符。
参数：(ax)=word 型数据
　　ds:si指向字符串的首地址
返回：无
应用举例：编程，将数据 12666 以十进制的形式在屏幕的 8 行 3 列，用绿色显示出来。在显示时我们调用本次实验中的第一个子程序 show_str。

```
assume cs:code

data segment
  db 10 dup (0)
data ends
```

```
code segment
  start:mov ax,12666
        mov bx,data
        mov ds,bx
        mov si,0
        call dtoc

        mov dh,8
        mov dl,3
        mov cl,2
        call show_str
        :
        :
        :
code ends
end start
```

提示

下面我们对这个问题进行一下简单地分析。

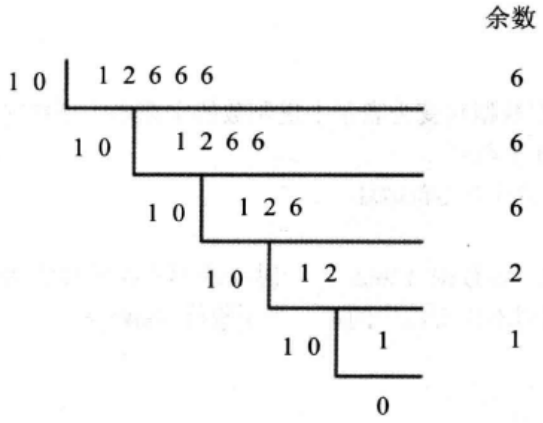
(1) 要得到字符串“12666”，就是要得到一列表示该字符串的 ASCII 码：31H、32H、36H、36H、36H。

十进制数码字符对应的 ASCII 码 = 十进制数码值 +30H。

要得到表示十进制数的字符串，先求十进制数每位的值。

例：对于 12666，先求得每位的值：1、2、6、6、6。再将这些数分别加上 30H，便得到了表示 12666 的 ASCII 码串：31H、32H、36H、36H、36H。

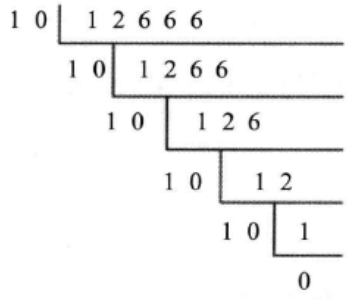
(2) 那么，怎样得到每位的值呢？采用下面的方法：



可见，用 10 除 12666，共除 5 次，记下每次的余数，就得到了每位的值。

(3) 综合以上分析，可得出处理过程如下。

用 12666 除以 10，循环 5 次，记下每次的余数；将每次的余数分别加 30H，便得到了表示十进制数的 ASCII 码串。如下：



余数	+30H	ASCII 码串	字符串
6		36H	‘6’
6		36H	‘6’
6		36H	‘6’
2		32H	‘2’
1		31H	‘1’

(4) 对(3)的质疑。

在已知数据是 12666 的情况下，知道进行 5 次循环。可在实际问题中，数据的值是多少程序员并不知道，也就是说，程序员不能事先确定循环次数。

那么，如何确定数据各位的值已经全部求出了呢？我们可以看出，只要是除到商为 0，各位的值就已经全部求出。可以使用 jcxz 指令来实现相关的功能。