\_07\_更灵活的定位内存地址

**1.AND、OR、XOR、NOT指令**

（1）逻辑运算指令介绍

①AND指令（按位与）

语法：AND dest, src

功能：将dest与src中的数据进行按位与运算，结果存储在dest中。

按位与：将dest和src对应的每一位分别相与（全1才取1）。

②OR指令（按位或）

语法：OR dest, src

功能：将dest与src中的数据进行按位或运算，结果存储在dest中。

按位或：将dest和src对应的每一位分别相或（有1就取1）。

③XOR指令（按位异或）

语法：XOR dest, src

功能：将dest与src中的数据进行按位异或运算，结果存储在dest中。

按位异或：将dest和src对应的每一位分别相异或（同0异1）。

④NOT指令（按位取反）

语法：NOT dest

功能：将dest中的数据进行按位取反。

按位取反：把dest的每一位都进行取反操作（1变0，0变1）。

（2）逻辑运算指令的用途

①AND指令用途：将操作对象的指定位设为0。

e.g. 将al的低三位设为0：and al, 11111000b

②OR指令用途：将操作对象的指定位设为1。

e.g. 将ah的高三位设为1：OR ah, 11100000b

③XOR用途：将操作对象的值置为0。

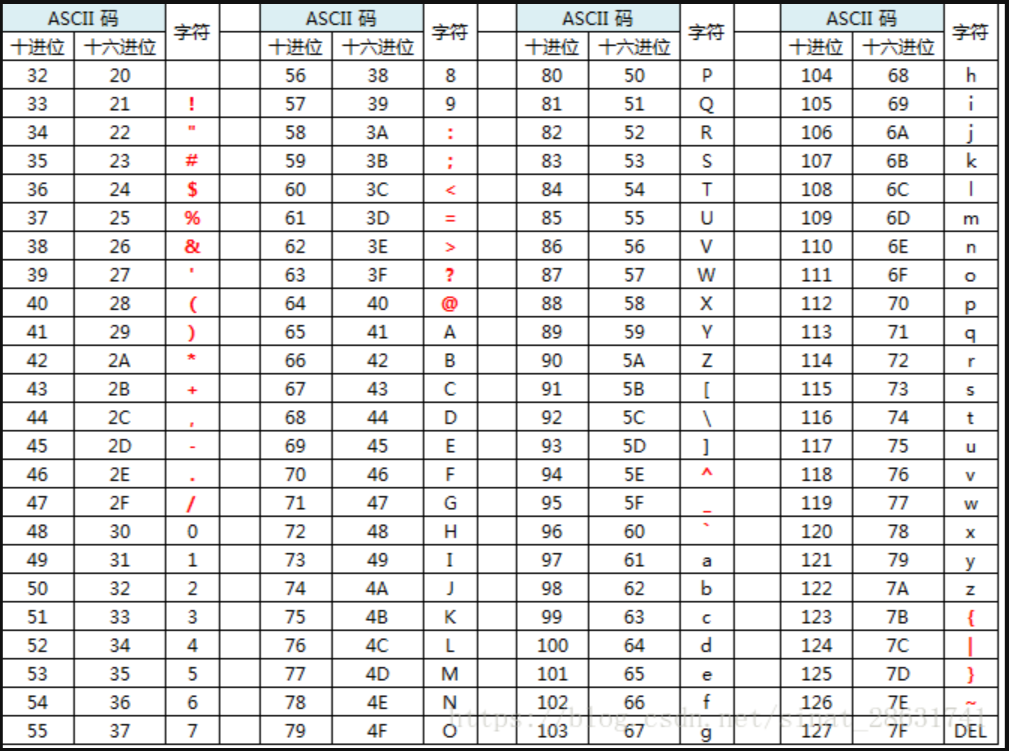
e.g. 将ax置为0：XOR ax, ax

**2.ASCII码**

（1）ASCII码介绍

**在汇编语言中，字符或字符串均以ASCII码的形式存储在内存中。**

（每个ascii码占用1字节）



**（2）字符、字符串在内存中的存储**

①定义字符

**语法：charName DB ‘a’ （等价于：charName DB 97）**

**功能：开辟1字节空间，用于存储字符的ASCII码。**

注意：①汇编语言中，用单引号表示字符，如’a’、’A’。

②字符在内存中以ASCII码的方式存储。

②定义字符串

**语法：stringName1 DB ‘abcdef$’**

**或：stringName2 DB ‘abcdef’, ‘$’**

**或：stringName3 DB ‘a’,’b’,’c’,’d’,’e’,’f’,’$’**

**功能：开辟一块连续的空间，用于存储字符串（字符数组）。其中，每1个字节单元存储一个字符的ASCII码。**

注意：①汇编语言中，用单引号表示字符串，如’abcdef$’。

②**汇编语言中，字符串的结束标志是’$’。利用中断输出字符串时，会从指定偏移地址的字符开始连续输出，直到遇到’$’为止（不输出’$’）。**

对于C语言，编译器会在字符串结尾隐式的自动添加’\0’作为结束标志；但是汇编语言并没有这样的自动机制，用户需要手动在字符串结尾添加’$’。

**（3）利用中断输出字符、字符串**

**①输出1个字符（2号功能）**

**语法： mov ah, 02h; DOS功能号：输出单个字符**

**mov dl, 'A'; 将要输出的字符(ASCII码)放入DL寄存器**

**int 21h; 调用DOS中断**

**功能：输出DL寄存器中存放的ASCII码对应的字符。**

**②输出字符串（9号功能）**

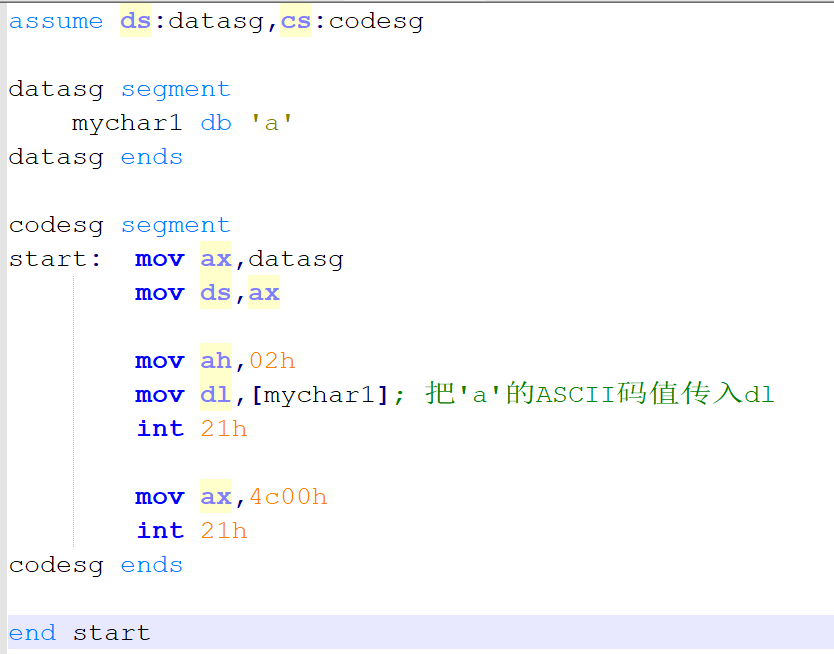
**语法： mov ah, 09h; DOS功能号：输出字符串**

**mov dx, offset 变量名; 将字符串的首地址放入DX寄存器**

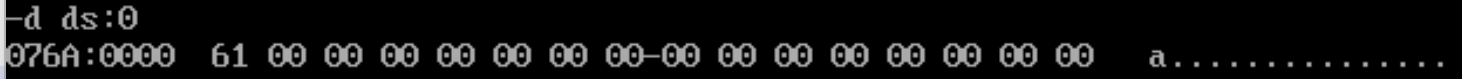
**int 21h; 调用DOS中断**

**功能：以DS:DX为字符串的起始地址，持续输出字符串，直到遇到’$’时停止（不输出’$’）。**

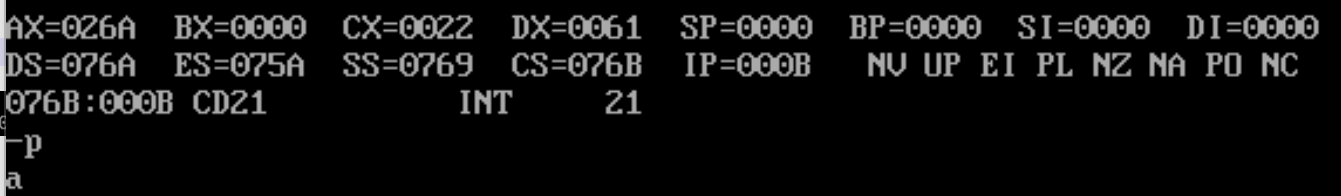
案例：输出内存中的字符



如下，字符在内存中以ascii码的形式存储：



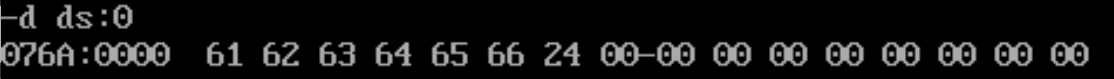
如下，利用中断的2号功能输出字符：



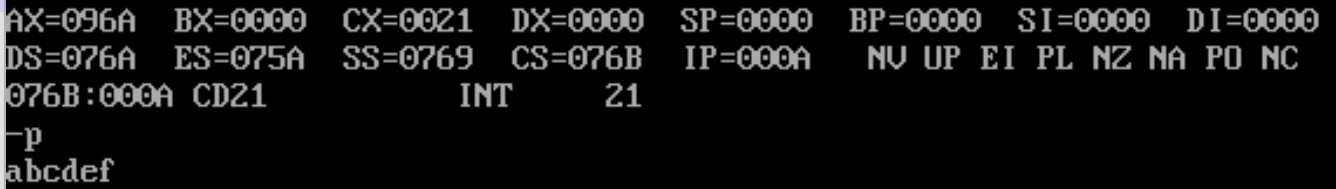
案例：输出内存中的字符串



如下，字符串在内存中以ascii码的形式存储：



如下，利用中断的9号功能输出字符串：



**（4）利用中断从键盘输入字符、字符串**

**①从键盘读取1个字符（1号功能）**

**语法： mov ah, 01h; DOS功能号：从键盘读取单个字符**

**int 21h; 调用DOS中断**

**; 此时，al存储了该字符的ascii码**

**功能：从键盘读取1个字符，将其ascii码存到al中。**

**②从键盘读取字符串（10号功能）**

**语法： ; 假设字符串名为buffer**

**mov [buffer], 最大字符数; 字符串的第一个字节存放缓冲区可容纳的最大字符数（不包括前两字节，包括末尾的回车）**

**mov ah, 10; DOS功能号：从键盘读取字符串**

**mov dx, offset buffer; 将字符串地址传入dx**

**int 21h; 调用DOS中断**

**; 此时，buffer中写入了用户输入的字符串**

**; 第一个字节是字符串可容纳的最大字符数（不含前两字节）**

**; 第二个字节是实际输入的字符数（不包括末尾的回车）**

**; 后续的字节是用户输入的字符串（包含回车）**

**功能：从键盘输入字符串，传入dx中的地址指向的空间。**

**注意事项：**

**若想输入包含100个字符的字符串，应该定义102个字节的空间，前两个字节用于存放最大字符数和实际字符数。**

**调用中断前，字符串的第一个字节应写入该串可容纳的最大字符数；调用中断后，第二个字节被写入实际输入的字符数（不包括回车），从第三个字节开始是用户输入的字符串。**

（5）字母大小写转换的问题

①通过加减法转换

由于‘A’的ASCII码是65（41H），’a’的ASCII码是97（61H），二者之间相差32（20H）。

因此，可以通过加减法转换字母的大小写：

大写字母 + 20H = 小写字母

小写字母 – 20H = 大写字母

②**通过逻辑运算转换**

大写字母的ASCII码为65~90（0100 0001~ 0101 1010）

小写字母的ASCII码为97~122（0110 0001~ 0111 1010）

可以发现，同一个英文字母的大小写形式对应的ASCII码，除了第6位不同，其它位的值均相同（小写字母第六位为1，大写字母第六位为0）。

因此，可以通过逻辑运算转换字母的大小写：

**大写字母 | 00100000B = 小写字母**

**小写字母 & 11011111B = 大写字母**