

# 汇编基础知识之输入输出

原创 Eastmount 2013-04-16 00:55:06 5588 收藏 4

展开



## Python+TensorFlow人工智能

该专栏为人工智能入门专栏，采用Python3和TensorFlow实现人工智能相关算法。前期介绍安装流程、基础语法、



Eastmount

¥9.90

订阅

最近在学习汇编程序,但输入输出是我遇到的第一个难题,这里就简单讲述一些汇编的基本输出输出字符串、字符、输出数字等知识,仅供大家学习。这只是一些基础知识,汇编高手亦可以温习一些基础的输入输出知识。

## 一.输出字符串

下面是调用9号功能输出字符串"Hello World"的代码。

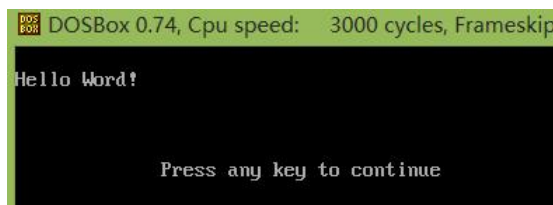
```
DATAS SEGMENT
    ;输出的字符串 13回车 10换行 $结束标志(\0)
    Input      DB      13,10,'Hello Word!',$
DATAS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS
START:
    MOV AX,DATAS
    MOV DS,AX      ;数据放入DS

    LEA DX,Input    ;字符串偏移起始DX,同MOV DX,OFFSET ASC指令
    MOV AH,9        ;09功能 打印字符串
    INT 21H         ;INT 21H是DOS中断功能调用的中断号,调用功能

    MOV AH,4CH      ;返回DOS系统
    INT 21H
CODES ENDS
END START
```

输出结果为:



## 二.输入输出单个字符

下面是调用2号功能输出单个字符"a"和调用1号功能输入单个字符的代码。

```
DATAS SEGMENT
    ;此处输入数据段代码
    Input      DB      13,10,'Please input:',$
DATAS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS
```

```

START:      MOV AX,DATAS
            MOV DS,AX      ;数据放入DS

            MOV DL,'a'      ;字符存入寄存器
            MOV AH,02H      ;调用2号功能输出单个字符
            INT 21H         ;INT 21H调用功能

            ;调用9号功能输出字符串"请输入"
            LEA DX,Input
            MOV AH,9
            INT 21H

            MOV AH,01H      ;调用1号功能输入字符(只能输入一个字符)
            INT 21H

            MOV AH,4CH      ;返回DOS系统
            INT 21H
CODES ENDS
END START

```

输出结果为:



### 三.输出4位数字

在做汇编实验时,总是遇到一个问题,不会输出一串数字如:2030.(8位最大5位数)

下面就是输出一串数字的代码,这里采用的方法是(必须是4位数字)

- 1.先除以1000.2030/1000=2(商)..30(余)判断商是否为0,不为0输出;
- 2.输出千位2后,商赋值给寄存器接着跳转到判断百位;
- 3.这里重点是设置了一个判断变量Nozero,如果这里是3/1000=0(商)..3(余),为0就判断该变量是否为0,为0表示它小于1000,不输出0,否则会输出0003.而当数字第一位不为0时(如2030第一位是2,不为0)就令变量Nozero=1,后面要输出的0都是在数字中显示输出.(如2000,当输出2时,Nozero=1,余数为0,后面的0均输出.而30只会到十位Nozero才赋值为1,输出个位的0).
- 4.依次判断十位,个位.

```

DATAS SEGMENT
;变量A和B为0-9999的无符号整数
        A                DW                2030 ;变量A
        Num              DW                ?
        Nozero           DW                ?    ;添加一个判断标志变量
DATAS ENDS

CODES SEGMENT
        ASSUME CS:CODES,DS:DATAS
START:
;-----
        MOV AX,DATAS
        MOV DS,AX

        ;16位出发在DX:AX一对寄存器中

```

```

MOV AX,A
MOV Nozero,0 ;like a boolean variable
;-----
;输出千位
Output0:
;输出数字 A存入AX中
XOR DX,DX
MOV Num,1000
DIV Num
MOV BX,AX ;商AX
MOV CX,DX ;余数DX
CMP BX,0
JNE OutputNum0 ;商AX!=0 输出数字

;重点:添加该判断防止0不能输出
CMP Nozero,0 ;变量Nozero!=0输出数字0
JNE OutputNum0 ;相当于0003前面3个0过滤 2000后面3个0显示
;-----
;输出百位
Output1:
MOV AX,CX ;DX:AX 余数CX赋值至AX
XOR DX,DX
MOV Num,100
DIV Num
MOV BX,AX ;商AX
MOV CX,DX ;余数DX
CMP BX,0
JNE OutputNum1
CMP Nozero,0
JNE OutputNum1
;-----
;输出十位
Output2:
MOV AX,CX ;DX:AX CX赋值至AX
XOR DX,DX
MOV Num,10
DIV Num
MOV BX,AX ;商AX
MOV CX,DX ;余数DX
CMP BX,0
JNE OutputNum2
CMP Nozero,0
JNE OutputNum2
;-----
;输出个位
Output3:
MOV AX,CX ;DX:AX CX赋值至AX
XOR DX,DX
MOV Num,10
DIV Num
ADD DX,30H ;余数DX
MOV AH,2
INT 21H

;退出程序
JMP Exit
;-----
OutputNum0:
;程序输出数字
MOV Nozero,1 ;标志变量赋值为1
ADD BX,30H ;数字加30H为ASC码数字
MOV DX,BX ;必须赋值到DX来调用中断,否则会报错

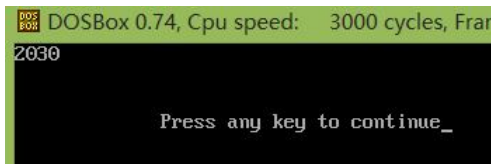
```

```

        MOV AH,2          ;调用2号功能输出字符
                                INT 21H
        JMP Output1
;-----
OutputNum1:
        ;程序输出数字
        MOV Nozero,1      ;标志变量赋值为1
        ADD BX,30H        ;数字加30H为ASC码数字
        MOV DX,BX         ;必须赋值到DX来调用中断,否则会报错
        MOV AH,2          ;调用2号功能输出字符
        INT 21H
        JMP Output2
;-----
OutputNum2:
        ;程序输出数字
        MOV Nozero,1      ;标志变量赋值为1
        ADD BX,30H        ;数字加30H为ASC码数字
        MOV DX,BX         ;必须赋值到DX来调用中断,否则会报错
        MOV AH,2          ;调用2号功能输出字符
        INT 21H
        JMP Output3
;-----
Exit:
        ;程序结束模板:使用AH来选择功能,其中4CH功能让你退出程序
        MOV AH,4CH
        INT 21H
;-----
CODES ENDS
        END START

```

输出的结果是:而当A赋值为3会输出相应的3而不是0003.



此方法似乎很笨拙,输出数字每次都是书写了4遍重复的输出代码,而且来回的跳转,但是更重要的是提供一个Nozero的思想,在C、C++、C#、Java中常用到的设置bool型变量的方法在汇编程序中同样有用并好用.在实验中我也采用的是下面的定义子程序方法输出数字.

## 四.采用函数输出4位数字

下面是百度知道里面的一种方法,调用函数输出数字的方法,由于还没有学到子函数,所以这里没有验证该方法,但感觉此方法还是相当有用的一种方法.上面设置的Nozero也是参照了它的思想.仅供大家学习.网址:<http://zhidao.baidu.com/question/54461195.html>

```

outd proc
    push cx
    mov nozero,0 ;like a boolean variable
    mov cx, 10000
    call decdiv
    mov cx, 1000
    call decdiv
    mov cx, 100
    call decdiv
    mov cx, 10
    call decdiv
    mov cx, 1

```

```

call decdiv      ;cmp nozero, 0
;jne outdexit
;mov dl, 30h
;mov ah, 2
;int 21h
outdexit:
pop cx
ret
;call: 被除数在bx传入,除数在cx传入
;ret: 打印商,余数送bx
decdiv proc
mov ax, bx
mov dx, 0
div cx
mov bx, dx
mov dl, al
cmp dl, 0
jne dispdigit
cmp nozero, 0
jne dispdigit
jmp decdivexit
dispdigit:
mov nozero, 1
add dl, 30h
mov ah, 2
int 21h
decdivexit:
ret
decdiv endp
outd endp

```

## 五.总结

最后,上面的文章主要是给刚学习汇编的同学参考,大神略过.同时,代码也不是很优化,比较冗余,但它还是提供了一些很好的汇编思想供大家学习,希望大家能学到一些思想.反正对我刚学习汇编的此时,上面4个代码还是非常有用的.

由于刚学习发表博客,可能内容和格式不是很好,还请海涵! By: Eastmount



**Eastmount**   博客专家

原创文章 462 获赞 6725 访问量 525万+

关注

他的留言板