

汇编语言程序包括三个类型的语句：

- 可执行指令或指令
- 汇编指令或伪操作
- 宏

可执行指令或简单指示告诉的处理器该怎么做。每个指令由操作码（操作码）可执行指令生成的机器语言指令。

汇编指令或伪操作告诉汇编有关汇编过程的各个方面。这些都是非可执行文件，并不会产生机器语言指令。

宏基本上是一个文本替换机制。

汇编语言语句的语法

汇编语言语句输入每行一个语句。每个语句如下的格式如下：

```
[label]    mnemonic    [operands] [;comment]
```

方括号中的字段是可选的。基本指令有两部分组成，第一个是要执行的指令（助记符）的名称和所述第二命令的操作数或参数的。

以下是一些典型的汇编语言语句的例子：

```
INC COUNT          ;Increment the memory variable COUNT
MOV TOTAL,48;Transfer the value 48in the
                   ; memory variable TOTAL
ADD AH, BH         ;Add the content of the
                   ; BH registerinto the AH register
AND MASK1,128;Perform AND operation on the
                   ; variable MASK1 and128
ADD MARKS,10;Add10 to the variable MARKS
MOV AL,10;Transfer the value 10 to the AL register
```

.data section(节)

这个section主要存放初始化的数据，.data section包含利用像文件名、缓冲大小，并且还可以用EQU定义常量(constant)，可以使用的一些指令如：DB,DW,DD,DQ,DT

例：

section .data

```
    message:          db 'Hello world!'    ;相当于char/unsigned char* Hello
```

world!

```
    msglength:        equ 12                ; 字符串长度12字节
```

```
        buffersize:    dw 1024                ;缓冲区大小1024个字长(相当于short类型)
```

```
.bss section        ;未初始化section
```

;这个section存放未初始化数据，可以用RESB,RESW,RESQ和REST指令来为你的变量申请为初始化空间。

```
section .bss
```

```
    filename: resb 255                ;255字节
```

```
    number: resb 1
```

```
    bignum: resw 1
```

```
    realarray: resq 10
```

```
.text section    ;代码section
```

这个section用于存放用户代码，.text section必须从global _start开始，来告诉内核程序从什么地方开始执行(类似于C或JAVA中的main函数，这里指一个开始位置)

```
section .text
```

```
    global _start
```

```
_start:
```

```
    pop ebx                ;这里是程序实际开始的地方
```

```
    .  
    .  
    .
```

正如你所看到的，到目前为止，或者多或少都有一点DOS的味道，下面我们通过讲解linux系统调用之后，便可以完成你的第一个linux汇编程序了。