



# 第七章 高级汇编语言技术

- 宏汇编
- 重复汇编
- 条件汇编

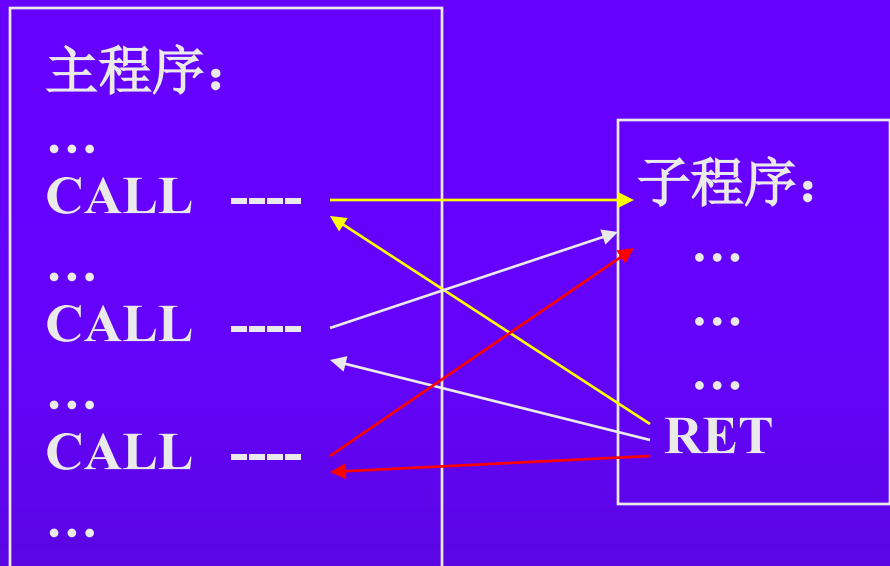


# 1. 宏汇编

**宏：**源程序中一段有独立功能的程序代码。

**宏指令：**用户自定义的指令。在编程时，将多次使用的功能用一条宏指令来代替。

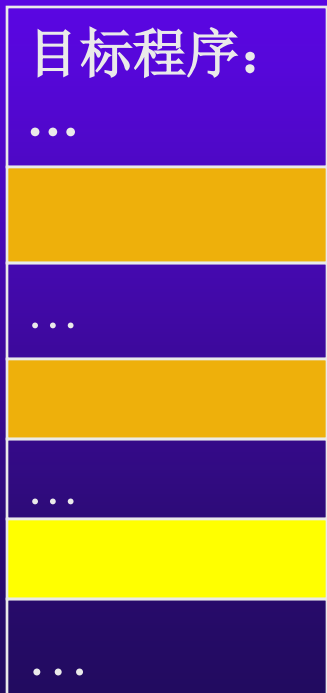
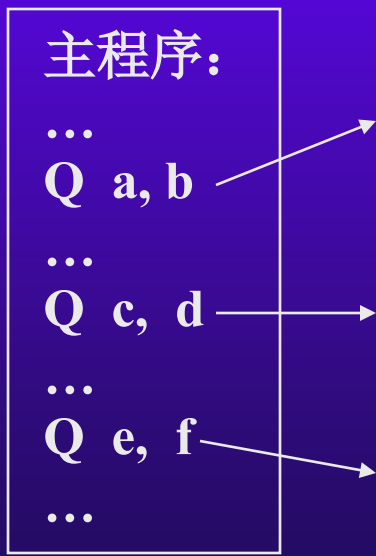
汇编语言程序 {  
指令  
伪指令（伪操作）  
宏指令



优: 模块化  
省内存  
缺: 开销大

宏定义:

```
Q macro x,y  
...  
endm
```



优:  
参数传送  
简单, 执行效率高  
缺:  
占用内存  
空间大



## 宏定义:

```
macro_name    MACRO [哑元表] ;形参/虚参  
              [LOCAL 标号表]  
              .....  
              ..... ;宏定义体  
              ENDM
```

## 宏调用: (必须先定义后调用)

```
macro_name    [实元表] ;实参
```

## 宏展开: 汇编程序把宏调用展开

宏定义体  $\longrightarrow$  复制到宏指令位置, 实参代虚参

LOCAL中的标号  $\longrightarrow$  ??0000~??ffff

## 例：保存寄存器

宏定义：

```
savereg  MACRO  
        push  ax  
        push  bx  
        push  cx  
        push  dx  
        push  si  
        push  di  
        ENDM
```

宏调用：  
savereg

宏展开：

```
1  push  ax  
1  push  bx  
1  push  cx  
1  push  dx  
1  push  si  
1  push  di
```



## 例：两个字相乘

宏定义：

```
multiply MACRO opr1,opr2,result  
    push    dx  
    push    ax  
    mov     ax,opr1  
    imul   opr2  
    mov     result,ax  
    pop     ax  
    pop     dx  
ENDM
```

宏调用：

```
multiply cx,var,xyz[bx]
```

宏展开：

```
1  push    dx  
1  push    ax  
1  mov     ax,cx  
1  imul   var  
1  mov     xyz[bx],ax  
1  pop     ax  
1  pop     dx
```

## 例：求绝对值

宏定义：

```
absol MACRO oper  
      LOCAL next  
      cmp oper,0  
      jge next  
      neg oper
```

**next:**

**ENDM**

宏调用：

```
.....  
absol var
```

```
.....  
absol bx  
.....
```

宏展开：

```
.....  
1      cmp   var,0  
1      jge   ??0000  
1      neg   var  
1  ??0000:  
.....
```

```
.....  
1      cmp   bx,0  
1      jge   ??0001  
1      neg   bx  
1  ??0001:  
.....
```





宏汇编操作符: & ; ; %

符号1 & 符号2

宏展开时, 合并前后两个符号形成一个符号。

;; 注释

宏展开时, ;;后面的注释不予展开。

% 表达式

汇编程序将%后面的表达式转换为数字, 并在展开期间用这个数取代哑元。





宏定义:

```
leap macro cond, lab  
    j&cond lab  
endm
```

宏调用:

```
leap z, there  
.....  
leap nz, here
```

宏展开:

```
1 jz there  
.....  
1 jnz here
```



宏定义:


```
msg macro Tab,num,xyz  
Tab&num db 'HELLO MR.&xyz'  
endm
```

宏调用:

```
msg ary,1,wang  
.....  
msg ary,2,zhang
```

宏展开:

```
1 ary1 db 'HELLO MR.wang'  
.....  
1 ary2 db 'HELLO MR.zhang'
```



例：定义宏指令**STRG**，实现存储字符串的功能

宏定义：**STRG** **MACRO** **STRING**  
**DB** **'&STRING&'**  
**ENDM**

宏调用：**STRG** **25-1**  
**STRG** **% 25 -1**

宏展开：**1** **DB** **'25-1'**  
**1** **DB** **'24'**



宏定义:

```
msg      macro  count,string
          msg&count db string
          endm

errmsg   macro  text
          cntr = cntr+1
          msg   %cntr,text
          endm
```

宏调用:

```
cntr=0
.....
errmsg  'SYNTAX ERROR'
.....
errmsg  'INVALID OPERAND'
```

宏展开:

```
1  cntr = cntr+1
1  msg  %cntr,'SYNTAX ERROR'
2  msg1 db 'SYNTAX ERROR'
.....
1  cntr = cntr+1
1  msg  %cntr,'INVALID OPERAND'
2  msg2 db 'INVALID OPERAND'
```



## 列表伪操作:

- .LALL** : 在LST清单中列出宏展开后的全部语句(包括注释)。
- .SALL** : 在LST清单中不列出任何宏展开后的语句。
- .XALL** : 缺省的列表方式, 只列出宏体中产生目标代码的语句。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/167104111026006146>