



汇编语言程序设计

主讲：乔瑞萍

西安交通大学信通学院

汇编语言程序设计



内容提要

重要概念解读

典型题解析

学
习
要
点



重要概念解读

- 标号与变量的区别
- 指令与伪指令的区别
- 比较宏与子程序的异同
- 机器语言、汇编语言、高级语言、混合语言



1、标号与变量的区别

标号是某条指令所存放单元的符号化地址，只能在代码段中，可通过标号来引用所标识的指令（如跳转、调用指令等）；

变量是数据所存放单元的符号化地址，一般位于数据段或堆栈段中。可用各种寻址方式对变量进行存取。

变量和标号有段基址、偏移地址和类型三种共同属性，此外，变量还有两个特有属性：长度和字节数。



2、指令与伪指令的区别

指令语句是可执行语句，由硬件（CPU）完成其功能。

伪指令是为汇编和链接程序提供编译和链接信息，属**不可执行**语句，其功能由相应软件完成。



3、比较宏与子程序的异同

- 相同点：均可用来简化源程序，并可使程序对它们多次进行调用。
- 不同点：
 - ①定义方法及格式不同。
 - ②子程序省内存，宏指令则不省。
 - ③子程序执行速度慢，而宏运行速度快。



4、机器语言、汇编语言、高级语言、混合语言

- 机器语言：指令的二进制代码编程。
- 汇编语言：用一组字母或符号表示的指令编程。
- 高级语言：高级语言接近自然语言，一般总是独立于具体机器的，程序员可不必了解机器的指令系统和内部的具体结构，而把精力集中在正确掌握语言的语法规则和算法的程序实现上。一条高级语言指令对应一段汇编语言程序。
- 混合语言：采用两种或两种以上的编程语言加以组合编程，是一种程序接口技术，实现不同语言间的相互调用。



典型题解析

例1 有下列数据段定义：

```
DATA SEGMENT
```

```
MAX EQU 03F9H
```

```
VAL1 EQU MAX MOD 0AH
```


```
VAL2 EQU VAL1*2
```

```
BUFF DB 4, 5, '1234'
```

```
EBUFF DB ?
```


```
LEND EQU EBUFF-BUFF
```

```
DATA ENDS
```

**试写出数据段中MAX、VAL1、
VAL2、LEND符号所对应的值。**

答案： 3F9H, 7, 0EH, 6



例2 图示以下数据段在存储器中的
存放形式:

```
DATA      SEGMENT  
  A  DB  10,2,? , -5  
  B  DB  'ABCD'  
  C  DW  4 DUP(0)  
  N  EQU 12  
  X  DD  33H,02H,0AH  
  Y  DD  0ABCDH  
DATA      ENDS
```

DS:0000H

0AH

‘A

DS:0010H33H

‘X

02H

00H

?

00H

FAH

00H

DS:0004

41H

‘B

02H

42H

00H

43H

00H

44H

DS:0008

00H

‘C

00H

00H

0AH

00H

00H

00H

00H

00H

00H

00H

00H

DS:001CH

0CDH

‘Y

00H

0ABH

00H

00H

例3 用查表法将一位十六进制数转换成与它相应的ASCII码。（顺序程序）

DATA SEGMENT

```
TAB     DB 30H, 31H, 32H, 33H, 34H, 35H, 36H, 37H
         DB 38H, 39H, 41H, 42H, 43H, 44H, 45H, 46H
NUM     DB    4
ASCII    DB    ?
```


DATA ENDS

```
STK     SEGMENT        STACK   'STACK'
         DW 20H        DUP (0)
```

STK ENDS

CODE SEGMENT

```
ASSUME   CS: CODE, DS: DATA, SS: STK
```



```
BEGING:  MOV  AX, DATA
          MOV  DS, AX
          MOV  BX, OFFSET TAB
          MOV  AL, NUM
          XLAT
          MOV  ASCI, AL
          MOV  AH, 4CH
          INT  21H
CODE      ENDS
          END   BEGING
```

说明: XLAT ; $AL \leftarrow ((BX) + (AL))$



例5 设有一个有符号数数组，共M个字，试编一程序求其中最大的数。

STACKSG SEGMENT STACK

DW 32 DUP (?)

STACKSG ENDS

DATA SEGMENT

DAT DW -3,12,-8,7,1


M EQU (\$-DAT)/2

MAX DW ?

DATA ENDS

CODE SEGMENT

ASSUME CS: CODE, DS: DATA, SS: STACKSG



```
START:  MOV AX, DATA
        MOV DS, AX
        LEA SI, DAT
        MOV CX, M
        DEC CX
        MOV AX, [SI]
        MOV MAX, AX
L1:      ADD SI, 2
        MOV AX, [SI]
        CMP AX, MAX
        JNG LAB
        MOV MAX, AX
LAB:     LOOP L1
        MOV AH, 4CH
        INT 21H
CODE    ENDS
        END  START
```



例6 请完成下列汇编语言源程序的数据段和代码段，其功能是利用DOS系统功能调用INT 21H，在计算机的CRT上显示“PASSWORD: ”。

提示：用到的DOS系统功能调用INT 21H的9号子功能说明如下：

； 功能：把缓冲区中的字符串在显示器上显示出来（该串以‘\$’为结束符）

； 入口参数：AH=9，DS:DX=缓冲区首地址

； 出口参数：无。

```
SSEG  SEGMENT STACK
```

```
        DW 256 DUP(?)
```

```
SSEG  ENDS
```

```
DATA  SEGMENT
```

```
        PASSW DB      ...      ...
```

```
DATA  ENDS
```

```
CODE  SEGMENT
```

```
        ...      ...
```

```
CODE  ENDS
```

```
        END  START
```



Data **segment**
PASSW **dB** **‘PASSWORD: ’ , 0DH, 0AH, ‘\$’**

Data **ends**

stack **segment** **stack**
 dW 256 dup(?)

stack **ends**

code **segment**
 assume cs:code,ds:data,ss:stack

start: **mov ax,data**
 mov ds,ax

LEA DX,PASSW
 MOV AH,09H
 INT 21H

mov ah,4ch
 int 21h
code **ends**
 end start

例7 设有两个等字长、字节型字符串，试编写一汇编语言子程序，比较它们是否完全相同；若相同屏幕显示Y，否则显示N。

MODEL SMALL

.386

.DATA

STR1 DB 'ASdAS'

STR2 DB 'ASASd'


CN EQU STR2-STR1

.CODE

.STARTUP

```
MOV    AX,DS
MOV    ES,AX
LEA    SI,STR1
LEA    DI,STR2
MOV    CX,CN
CLD
REPZ   CMPSB
JNZ    LABNE
```

```
MOV    DL,'Y'
MOV    AH,02H
INT     21H
JMP     EXIT
LABNE:  MOV    DL,'N'
        MOV    AH,02H
        INT     21H
EXIT:   .EXIT
        END
```

```
DATA SEGMENT
X      DW 00F2H
RESULT DW ?
DATA  ENDS
STACK SEGMENT STACK
      DB 20 DUP (?)
STACK ENDS
CODE  SEGMENT
      ASSUME
CS:CODE,DS:DATA,SS:STACK
START: MOV  AX,DATA
      MOV  DS,AX
      MOV  BX,X
      CALL NEAR PTR NOTR
```

```
      MOV  RESULT,BX
      MOV  AH,4CH

      INT  21H

NOTR  PROC  NEAR

      NEG  BX

      RET

NOTR  ENDP

CODE  ENDS

      END  START
```

*****子程序设计例7*****