课时2 汇编语言编程基础



考点	重要程度	占分	题型
1.程序结构	****	5~10	
2.指令语句和伪指令	***		大题
3.数据变量定义	必考	0~3	选择 填空
4.数据段定义及规范			
5.代码段定义及规范			

考点1 程序结构

一、汇编语言源程序结构

```
DATA SEGMENT
   ;此处输入数据段代码
DATA ENDS
STACK SEGMENT
   ;此处输入堆栈段代码
STACK ENDS
CODE SEGMENT
   ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
   MOV AX, DATAS
   MOV DS,AX
   ;此处输入代码段代码
   MOV AH,4CH
   INT 21H
CODE ENDS
```

二、Hello world 程序实例

定义一个名为string的字符串,string是字符串的名称,db是定义字节说明,字符串内容需用单引号括起, 其中\$是串的结束标志

调用9号DOS功能(显示字符串)

ATA SEGMENT ;数据段 string db 'Hello, World!\$' DATA ENDS STACK SEGMENT PARA STACK; 堆栈段定义开始 dw 20h dup (0);为堆栈段分配空间 STACK ENDS 这是一条汇编伪指令,含 CODE SEGMENT :代码段 义是指定code段与CS寄存 assume cs:code,ds:data 器关联, data段与DS寄存 START: 器关联。 mov ax,data ;获取段基址 mov ds,ax;将段基址送入寄存器 mov dx,offset string 获取string的偏移地址。 mov ah,9 int 21h mov ah,4ch int 21h CODE ENDS 调用程序结束功能 **END START**

【题1】若要源程序从符号START开始执行,源程序结束伪指令格式为

(END START)



汇编语言的语句可以分为指令语句和伪指令语句

一条指令语句最多可以包含4个字段

每一条指令语句在汇编时都要产生一个可供CPU执行的机器目标代码,它又叫可执行语句

伪指令本身并不产生对应的机器目标代码。它仅仅是告诉汇编程序对其后面的指 令语句和伪指令语句的操作数应该如何处理。

【题2】伪质量与指令的本质差别是在汇编过程为指令不形成任何代码(√)

【题3】以后会遍程序在汇编过程中执行的指令叫(伪指令)

考点3 数据变量定义

一、数据变量5种定义命令

DB: 定义变量为字节类型, 其后的每个操作数都占一个字节。

DW: 定义变量为字类型, 其后的每个操作数都占两个字节。

DD: 定义变量为双字类型,其后的每个操作数都占二个字,即四个字节。

DQ: 定义变量为四个字类型, 其后的每个操作数都占四个字, 即八个字节。

DT: 定义变量为十个字节, 其后的每个操作数都占十个字节。

如:

STR DB 'STRING' NUM DW 0AAH, 23H LAB0 DQ 01A4578H



二、字节、字、双字的概念

如:

DATA_B DB 10, 5, 10H

DATA_W DW 100H,-4

DATA_D DD OFFFBH

三、过程定义伪指令

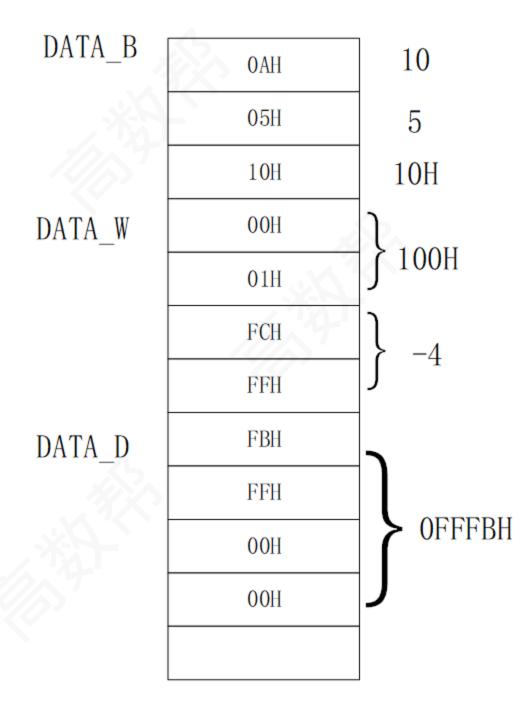
其格式为:

过程名 PROC [类型]

过程体

RET

过程名 ENDP



四、字符串表达式

1.对DB伪指令,给每一个字符分配一个字节单元。字符串按从左到右,将字符的 ASCII编码值以地址递增的排列顺序依次存放。

2.对DW伪指令

注意:两个字符的存放顺序是前一个字符放在高地址,后一字符放低地址单元。

如: STRING1 DB 'ABCDEF'

如: STRING2 DW 'AB', 'CD', 'EF'

'в'	41H
'A'	42H
'D'	43H
·c·	44H
'F'	45H
"E'	46H
	'A' 'D' 'C' 'F'

'A'

'B'

'D'

'E'

'F'

3.对DD伪指令

注意: DW和DD伪指令不能用两个以上字符构成的字符串赋初值,否则将出错。

如: STRING3 DD 'AB', 'CD'

42H	'B'
41H	'A'
0	
0	
44H	'D'
43H	'c'
0	
0	

考点4 数据段定义及规范

一、复制操作符'DUP'的使用

DUP称为重复数据操作符。 使用DUP表达式的一般格式为:

变量名
$$\left\{ egin{array}{l} DB \\ DW \\ DUP \end{array}
ight\}$$
 表达式1 DUP (表达式2)

其中: 表达式1是重复的次数, 表达式2是重复的内容。

【题4】 DATA_B DB 20H DUP('AB')

分配20H*2=40H个字节,其内容为重复字符串'AB'

[题5] DATA_C DB 10H DUP(4 DUP(2),7) //"2, 2, 2, 2, 7", 共占用 10H*5=50H个字节。

二、地址计数器\$ 保存当前正在汇编的指令的地址。

如: JNE \$+6; 条件满足时转向的地址是JNE的首址+6;

JMP \$+2; JMP只占2B, 所以无条件转向下一条指令。

三、对准伪操作org 设置当前地址计数器\$的值。

其格式是: ORG 数值表达式

如: DATA SEGMENT USE16

ORG 10; ;设置\$为10,此段目标代码从偏移地址10开始

BUF DB 'ABCD' ; BUF的偏移地址为10

ORG \$+5 ; \$增加5, 即在ABCD之后空出5个字节

NUM DW 50 ; NUM的偏移地址为19

DATA ENDS

考点5 代码段定义及规范

一、算术运算符

+, —, *, /, MOD, SHL, SHR, []

如:

NUM=15 * 8; NUM=120

NUM=NUM/7; NUM=17

NUM=NUM MOD 3; NUM=2

NUM=NUM+5; NUM=7

NUM= -NUM - 3; NUM= - 10

NUM=-NUM-NUM; NUM=20

1.使用"+"、"-"、"*"、和"/"运算符时,参加运算的数和运算结果都是整数。

2.下标运算符"[]"具有相加的作用

下面的语句是等效的。

MOV AX, DA_WORD[20H]

MOV AX, DA_WORD+20H

下面语句是等效的

MOV AX, ARRAY[BX][SI]; 基址变址寻址

MOV AX, ARRAY[BX+SI]

MOV AX, [ARRAY+BX][SI]

MOV AX, [ARRAY+SI][BX]

MOV AX, [ARRAY+BX+SI]



二、逻辑运算符

NOT、AND、OR和XOR

三、关系运算符

EQ(等于)、NE(不等于)、LT(小于)、LE(小于等于)、GT(大于)、GE(大于等于)

如:

MOV AX, 0FH **EQ** 1111B => MOV AX, 0FFFFH MOV BX, 0FH **NE** 1111B => MOV BX, 0

四、移位运算符

移位运算符:某一具体的数

移位指令:对一个寄存器或存储单元内容在程序运行时执行移位操作

"SHR"为右移,左边移出来的空位用0补入

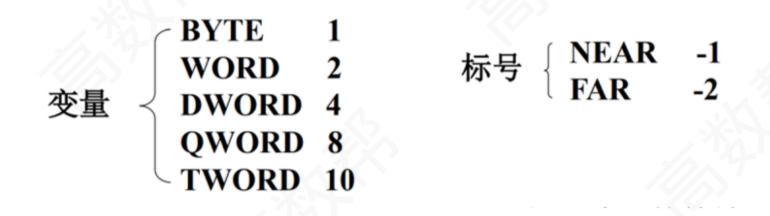
"SHL"为左移,右边移出来的空位用0补入

如: MOV AX, NUM SHL 1



五、数值返回运算符

- 1.SEG运算符: 取变量或标号所在段的段基值
- 2.OFFSET运算符:该运算符的作用是取变量或标号在段内的偏移量。
- 3.TYPE运算符: 取变量或标号的类型属性,并用数字形式表示。对变量来说就 是取它的字节长度。



如: V1 DB 'ABCDE'

V2 DW 1234H, 5678H

V3 DD V2

•••••

MOV AL, TYPE V1 MOV CL, TYPE V2 MOV CH, TYPE V3

经汇编后的等效指令序列如下: MOV AL, 01H

MOV CL, 02H

MOV CH, 04H

4.LENGTHEN运算符: 取变量的长度。

```
K1 DB 10H DUP (0)

K2 DB 10H, 20H, 30H, 40H

K3 DW 20H DUP (0, 1, 2 DUP (0)

K4 DB 'ABCDEFGH'
```

```
MOV AL, LENGTH K1; (AL)=10H
MOV BL, LENGTH K2; (BL)=1
MOV CX, LENGTH K3; (CX)=20H
MOV DX, LENGTH K4; (DX)=1
```

5.SIZE运算符: LENGTH和 TYPE两个运算符返回值的乘积。

【题4】如TABLE为数据段中0032单元的符号名,其中存放的内容为1234H,试问以下两条指令有什么区别?指令执行完后AX寄存器的内容是什么?

MOV AX, TABLE

MOV BX, TABLE

MOV AX, TABLE是将TABLE单元的内容送到AX, (AX)=1234H LEA AX, TABLE是将TABLE单元的有效地址送到AX, (AX)=0032H

