

数据定义伪指令

伪指令

- 由汇编程序执行的“指令系统”

- 作用：

- 定义变量；
- 分配存储区
- 定义逻辑段；
- 指示程序开始和结束；
- 定义过程等。

帮助计算机理
解助记符指令
编写的汇编语
言源程序

常用伪指令



数据定义伪指令

符号定义伪指令

段定义伪指令

结束伪指令


过程定义伪指令

宏命令伪指令


一、数据定义伪指令

- 用于定义数据区中变量的类型及其所占内存空间大小
- 格式:


[变量名] 伪指令助记符 操作数, ... ; [注释]



符号地址



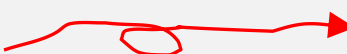

定义变量类型



变量值, 可以是常数, 表达式或字符串。其大小不能超过伪指令助记符所限定的范围

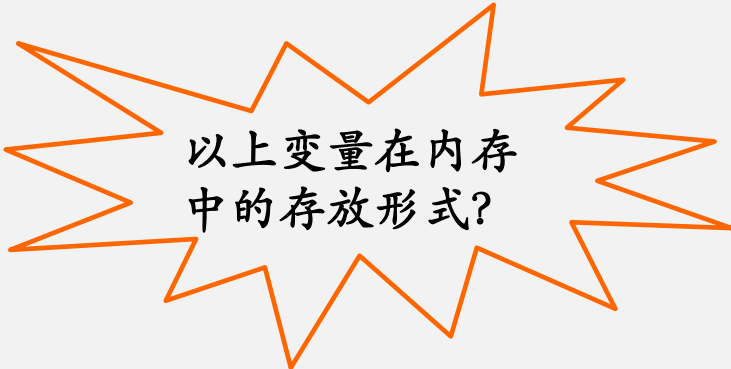
变量的类型及其操作数的个数决定了该变量所在内存空间的大小

1. 数据定义伪指令助记符

- DB (Define Byte) :  指向的每一个操作数占1个字节单元
 - 定义的变量为字节型
- DW (Define Word) :  指向的每一个操作数占1个字单元
 - 定义的变量为字类型
- DD (Define Double Word) :
 - 定义的变量为双字型
- DQ (Define Quadword) :
 - 定义的变量为4字型
- DT (Define Tenbytes) :
 - 定义的变量为10字节型

数据定义伪指令例

- **DATA1 DB 11H, 22H, 33H, 44H**
- **DATA2 DW 11H, 22H, 3344H**
- **DATA3 DD 11H*2, 22H, 33445566H**



以上变量在内存
中的存放形式?

数据定义伪指令例_____变量在内存中的分布

DATA1

11

22

33

44

DATA2

11

00

22

00

44

33

DATA3

22

0

0

0

22

0

0

0

66

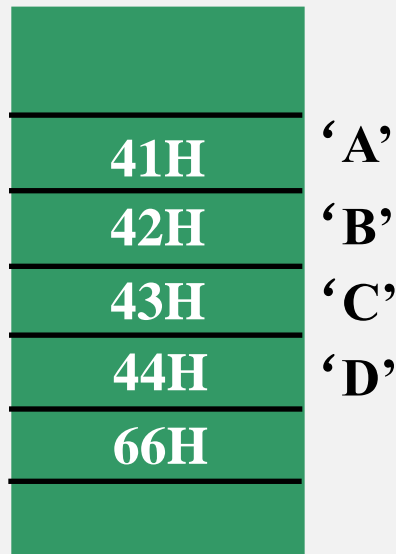
55

44

33

数据定义伪指令的几点说明

- 数据定义伪指令决定所定义变量的类型；
- 定义字符串必须用DB伪指令
- 例：
 - DATA1 DB 'ABCD', 66H



2. 重复操作符

- 当同样的操作数重复多次时，可以使用重复操作符。

- 作用：

- 为一个数据区的各单元设置相同的初值

- 格式：

[变量名] 伪指令助记符 **n** DUP (初值 [,初值,...])

- 例：

- M1 DB 10 DUP (0)

重复次数

重复的内容

常用于声明一个数据区


3. “？”的作用

- 表示随机值，用于预留存储空间


- 例：

- MEM1 DB 34H, 'A', ?

随机数
占1个字节单元



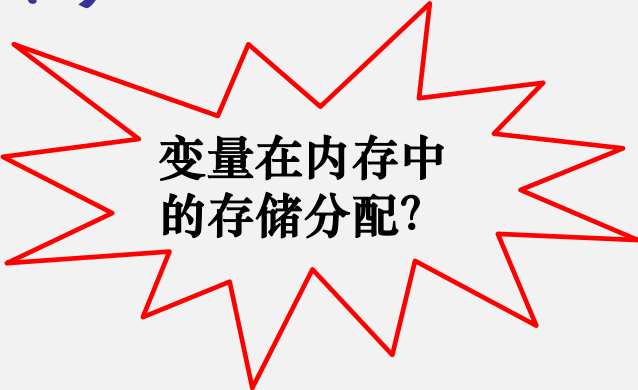
- DW 20 DUP (?)



预留40个字节单元，每单元为随机值

数据定义伪指令例

- M1 DB 'How are you?'
- M2 DW 3 DUP(11H), 3344H
- DB 4 DUP (?)
- M3 DB 3 DUP (22H, 11H, ?)



变量在内存中的存储分配？

数据定义伪指令例

M1

48H	'H'
6FH	'o'
77H	'w'
20H	' '
61H	'a'
72H	'r'
65H	'e'
20H	' '
79H	'y'
6FH	'o'
75H	'u'
3FH	'?'

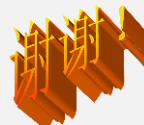
M2

11H
00H
11H
00H
11H
00H
44H
33H
XX
XX
XX
XX

M3

22H
11H
XX
22H
11H
XX
22H
11H
XX

随机数



调整偏移量伪指令

- 规定程序或变量在逻辑段中的起始地址

- 格式:

- ORG 表达式

计算值为
非负常数

- 例:

DATA SEGMENT

ORG 1200H

BUFF DB 1,2

DATA ENDS

变量BUFF的偏
移地址=1200H