



第10章 复合数据类型的定义和使用

10.1 结构体

结构体定义、结构变量的定义

结构变量的访问、结构信息的自动计算

10.2 结构变量的数据存储

10.3 union联合体

- C语言与汇编语言中，结构体、联合体定义异同；
- 不同编译器对结构变量中字段空间分配位置的差异





10.1 结构体

结构体定义

结构名 **struct**
 字段定义语句序列

结构名 **ends**

```
course      struct
cid         dd    0           ; 课程编号
ctitle      db    20 dup(0)   ; 课程名
chour       db    0           ; 学时数
cteacher    db    10 dup(0)   ; 主讲教师
cterm       db    1           ; 开课学期
course      ends
```





10.1 结构体

结构体定义

```
course      struct
cid         dword    0          ; 课程编号
ctitle      byte     20 dup(0)  ; 课程名
chour       byte     0          ; 学时数
cteacher    byte     10 dup(0)  ; 主讲教师
cterm       byte     1          ; 开课学期
course      ends
```

字段的定义: db, byte, sbyte





10.1 结构体——结构体的嵌套

结构体定义

```
department struct
    dname      db  10  dup(0)  ; 系名
    daddress   db  10  dup(0)  ; 系的办公地址
    coursetable course < >  ; 课表
department ends
```

```
department2 struct
    dname      byte  10  dup(0) ; 系名
    daddress   byte  10  dup(0) ; 系的办公地址
    coursetable course   5 dup (< >)
                                ; 课表, 5门课
department2 ends
```





10.1 结构体——结构变量的定义

结构需先定义后
使用

[变量名] 结构名 <字段赋值表>

local 变量名[数量] : 结构名

ke1 course < > ; 5个字段均用结构定义时给的初值

ke2 course <2102, 'math', 40, 'liming', 2>

course <2103, 'chinese', 80, 'zhangsan', >

; cterm字段未重新赋值, 默认为1

ke3 course 5 dup(<2104, , 60, , >)

; 分配了5个course结构大小的空间,

; 对cid、chour赋了值

course 10 dup(<>) ; 分配了10个course结构大小的空间





10.1 结构体

结构变量的访问

直接用变量名加字段名的方式访问

```
mov    eax, ke2.cid          ; 将2102送到eax寄存器中
mov    al,  ke2.ctitle       ; ctitle中的字符 'm' 送到al中
mov    ah,  ke2.ctitle+2     ; ctitle中的字符 't' 送到ah中

mov    eax, dword ptr [ke2 (0C850C3h)]——反汇编
mov    al,  byte ptr ds:[00C850C7h]
mov    ah,  byte ptr ds:[0C850C9h]
```





10.1 结构体

结构变量的访问

寄存器间接加字段名的方式访问

```
mov ebx, offset ke2  
mov al, [ebx].course.chour  
mov al, (course ptr [ebx]).chour
```





10.1 结构体

结构变量的访问

寄存器间接寻址访问

```
mov  ebx, offset ke2  
add  ebx, 18H      ; chour在course中的偏移地址是18H  
mov  al, [ebx]     ; 将ke2中chour字段的值→al
```





10.1 结构体

结构信息的自动计算

取偏移地址运算符offset

- offset 变量名
对一个全局变量，获得其在段中的偏移地址。
- offset 结构名. 字段名
获得一个字段在一个结构中的偏移地址。
- offset 结构变量名. 字段名
获得一个结构变量中指定的字段在段中的偏移地址。





10.2 结构变量的数据存储

汇编语言中结构变量的存储

- 从结构的第1个字段开始，依次、紧凑存放各个字段
- 在汇编程序中可以使用伪指令“align bound”来对齐字段或变量的边界，其中bound的取值为1、2、4、8、16，即2的n次方（ $n=0, 1, 2, \dots$ ）。

用法：写在变量或字段的下一行





10.2 结构变量的数据存储

汇编语言中结构变量的存储 与C语言结构变量存储的差异

- 一个结构的大小是与其编译时采用的“结构体对齐”参数有关的。
- 在Visual Studio中，可以在“C/C++/代码生成/结构成员对齐”中选择不同的**对齐方式**；
- 也可以在程序中使用语句“`#pragma pack(n)`”来设置对齐方式。





10.3 union 联合体

联合体名称 union
 字段定义
[联合体名称] ends

```
myunion    union
            num      dword    0
            chars    byte 4   dup(0)
myunion    ends
```

联合体变量与结构变量的用法类似

联合体中字段num和chars具有相同的偏移地址

