



C语言高级编程-结构体二

主讲：小美老师

创客引领未来

扫微信二维码 获取更多信息





lemon

昵称：小姜老师
华清创客学院，嵌入式讲师

makerU

结构体数组

- 结构体数组的定义
 - 具有相同结构体类型的结构体变量也可以组成数组，称它们为结构体数组。结构体数组的每一个数组元素都是结构体类型的数据，它们都分别包括各个成员（分量）项。
 - 定义结构体数组的方法和定义结构体变量的方法相仿，只需说明其为数组即可。可以采用三种方法：

结构体数组

1. 先定义结构体类型，再用它定义结构体数组。

结构体数组的定义形式如下：

```
struct 结构体名  
{  
    成员表列;  
};  
  
struct 结构体名
```

例如：

```
struct student  
{  
    char name[20];  
    char sex;  
    int age;  
    char addr[20];  
};  
struct student stu[3];
```

结构体数组

2. 在定义结构体类型同时定义结构体数组。

结构体数组的定义形式如下：

```
struct 结构体名  
{  
    成员表列;  
} 数组名[元素个数];
```

例如：

```
struct student  
{  
    char name[20];  
    char sex;  
    int age;  
    char addr[20];  
} stu[3];
```

结构体数组

3. 直接定义结构体数组

结构体数组的定义形式如下：

```
struct                //没有结构体名
{
    成员表列;
} 数组名[元素个数];
```

例如：

```
struct
{
    char name[20];
    char sex;
    int age;
    char addr[20];
} stu[3];
```

结构体数组

- 结构体数组的初始化

结构体数组在定义的同时也可以进行初始化，并且与结构体变量的初始化规定相同，只能对全局的或静态存储类别的结构体数组初始化。

结构体数组初始化的一般形式是：

```
struct 结构体名
```

```
{
```

成员列表；

```
};
```

```
struct 结构体名 数组名[元素个数]={初始数据表};
```

结构体数组

或者：

```
struct 结构体名
```

```
{
```

成员表列；

```
} 数组名[元素个数]={初始数据表};
```

由于结构体变量是由若干不同类型的数据组成，而结构体数组又是由若干结构体变量组成。所以要特别注意包围在大括号中的初始数据的顺序，以及它们与各个成员项间的对应关系。

结构体数组

- 结构体数组的使用

一个结构体数组的元素相当于一个结构体变量，因此前面介绍的有关结构体变量的规则也适应于结构体数组元素。以上面定义的结构体数组stu[3]为例说明对结构体数组的引用：

(1) 引用某一元素中的成员。

若要引用数组第二个元素的name成员，则可写为：

stu[1].name

(2) 可以将一个结构体数组元素值赋给同一结构体类型的数组中的另一个元素，或赋给同一类型的变量。如：

```
struct student stu[3], student1;
```

现在定义了一个结构体类型的数组，它有3个元素，又定义了一个结构体类型变量student1，则下面的赋值是合法的。

```
student1=stu[0];
```

```
stu[0]=stu[1];
```

```
stu[1]=student1;
```

结构体数组

(3) 不能把结构体数组元素作为一个整体直接进行输入输出。如：

```
printf(...,stu[0]);
```

或

```
scanf(...,&stu[0]);
```

都是错误的。

只能以单个成员为对象进行输入输出，如：

```
scanf(...,stu[0].name);  
scanf(...,&stu[1].num);  
printf(...,stu[0].name);  
printf(...,stu[1].num);
```

结构体与指针

- 结构体指针

可以设定一个指针变量用来指向一个结构体变量。此时该指针变量的值是结构体变量的起始地址，该指针称为结构体指针。

结构体指针与前面介绍的各种指针变量在特性和方法上是相同的。与前述相同，在程序中结构体指针也是通过访问目标运算“*”访问它的对象。结构体指针在程序中的一般定义形式为：

struct 结构体名 *结构指针名；

其中的结构体名必须是已经定义过的结构体类型。

例如，

对于上一节中定义的结构体类型 `struct student`，可以说明使用这种结构体类型的结构指针如下：

```
struct student *pstu;
```

其中 `pstu` 是指向 `struct student` 结构体类型的指针。结构体指针的说明规定了它的数据特性，并为结构体指针本身分配了一定的内存空间。但是指针的内容尚未确定，即它指向随机的对象。

当表示指针变量p所指向的结构体变量中的成员时，“(*结构体指针名).成员名”这种表示形式总是需要使用圆括号，显得很简炼。因此，对于结构体指针指向的结构体成员项，给出了另外一种简洁的表示方法，如下表示：

结构体指针名->成员名

它与前一种表示方法在意义上是完全等价的。例如，结构体指针p指向的结构体变量中的成员name可以表示如下：

`(*p).name` 或 `p->name`



扫微信二维码 获取更多信息