

第十章 结构体与共用体

10.1 结构体

10.2 共用体

结构体、共用体类型都是用户自己定义的数据类型，这些类型的数据是用户根据实际需要来组织的。使用结构体类型的指针能够构成链表，使用链表编程能够有效的利用存储空间。

西南林业大学计信学院

重点 10.1 结构体 难点

10.1.1 结构体类型的说明

10.1.2 结构体变量的定义

10.1.3 结构体类型指针的定义

10.1.4 结构体成员的引用、赋值

10.1.5 结构体数组

10.1.6 结构体与函数实参

10.1.7 链表

西南林业大学计信学院

10.1.1 结构体类型的说明

“结构体”是一种构造类型，是由数目固定，类型相同或不同的若干有序变量组成的集合。组成结构体的每个数据都称为结构体的“成员”，或称“分量”。

结构体类型说明的形式如下：

```
struct [结构体标识名]
{
    类型名1  结构体成员名列表1;
    类型名2  结构体成员名列表2;
    ...
    类型名n  结构体成员名列表n;
};
```

标识名可以省略
分号不能省略

西南林业大学计信学院

10.1.1 结构体类型的说明

例如：以下定义了一个日期型的结构体类型

```
struct data
{
    int year;
    int month;
    int day;
};
```

该结构体名称为 struct data
该结构体有三个成员，分别是 year、month、day

说明：

- (1) 结构体类型的说明可以嵌套。
- (2) **struct**是关键字。“结构体标识名”可以省略；右侧花括号后面的分号；不能省略。
- (3) 结构体类型的说明只是列出了该结构的组成情况，标志这种类型的结构模式已存在，但是编译程序并没有因此而分配任何存储空间。

西南林业大学计信学院

10.1.1 结构体类型的说明

用 typedef 定义类型

typedef是一个关键字，利用它可以将已存在的数据类型命一个新的名称，可以理解为给已有的数据类型取一个别名。

例如：**typedef int INT;** //INT 相当于是 int 的别名
INT x, y; //等价于 **int x, y;**

注意：**typedef**的使用并不是定义了一种新的类型，而仅仅是将已有的数据类型取了一个新名称。

由于结构体类型通常都比较复杂，利用**typedef**可以书写的结构体类型会一个简单的新变量。

西南林业大学计信学院

10.1.1 结构体类型的说明

方法一：先说明结构体类型，再使用typedef命别名

```
struct date
{
    int year; int month; int day; };
typedef struct date DATE;
```

方法二：说明结构体类型的同时就使用typedef

```
typedef struct date
{
    int year; int month; int day; } DATE;
```

方法三：与方法二相同，区别仅是缺省了结构体标识名

```
typedef struct
{
    int year; int month; int day; } DATE;
```

院

10.1.2 结构体变量的定义

对于已经声明的结构体类型，就可以定义该类型对应的“结构体变量”。

前面提到，结构体类型说明后，系统只是认可有这样一种用户自己构造的数据类型，但并不为之分配存储空间，只有定义了该结构体类型的变量之后，系统才为变量分配相应的存储空间。

结构体变量占用的存储容量由结构体类型决定。

以下几种是定义结构体变量的方法：

西南林业大学计信学院

重点 10.1.2 结构体变量的定义

方法一：先说明结构体类型，然后定义结构体变量

```
struct date
{
    int year;
    int month;
    int day;
};
struct date x, y;
```

方法二：使用typedef直接为结构体类型命名，然后定义结构体变量

```
typedef struct date
{
    int year;
    int month;
    int day;
}DATE;
DATE x, y;
```

结构体类型名
结构体变量名

西南林业大学计信学院

续上 10.1.2 结构体变量的定义

方法三：先说明结构体类型，然后使用typedef为结构体类型重命名，最后定义结构体变量

```
struct date
{
    int year;
    int month;
    int day;
};
typedef struct date DATE;
DATE x, y;
```

结构体类型名
结构体变量名

方法四：说明结构体类型的同时，就定义结构体变量

```
struct date
{
    int year;
    int month;
    int day;
}x, y;
```

结构体变量名

西南林业大学计信学院

10.1.2 结构体变量的定义

练习题

(1) 以下叙述不正确的是 (B)

```
struct ex
{
    int x; float y; char z; } example;
```

A) struct结构体类型的关键字

B) example是结构体类型名

C) x,y,z都是结构体成员名

D) struct ex是结构体类型

西南林业大学计信学院

10.1.2 结构体变量的定义

练习题

(2) 以下叙述正确的是 (B)

```
typedef struct
{
    int n;
    char ch[8];
}PER;
```

A) PER 是结构体变量名

B) PER是结构体类型名

C) typedef struct 是结构体类型

D) struct 是结构体类型名

西南林业大学计信学院

10.1.3 结构体类型指针的定义

结构体指针的定义方法与结构体变量的定义方法相同，也有多种方法实现。

需要注意的是，结构体指针定义后，与普通类型的指针一样，必须将其指向有效的存储空间，然后才能使用。空指针是不能使用的。

例如：

```
typedef struct stu
{   char name[30];
    char sex;
    int  age;
    double score;
}STU;
STU x, *p = &x;
```

以上说明了一个名称为STU的结构体类型，并定义了该结构体类型的变量x及指针p，并且对指针进行初始化。

10.1.3 结构体类型指针的定义

```
typedef struct stu
{ char name[30];
  char sex;
  int age;
  double score;
}STU;
STU x, *p = &x;
```

有了以上定义，系统便根据结构体类型各成员占用的存储单元总和为结构体变量 x 分配 $30+1+4+8=43$ 个字节的存储空间。

指针 p 初始化为指向变量 x ，即指向这43个字节存储空间的首地址。

该结构体变量 x 代表一名学生的信息，它由4个成员（姓名、性别、年龄、分数）组成，对这4个成员的引用要分开进行，以下介绍引用方法。

西南林业大学计信学院

10.1.4 结构体成员的引用、赋值

对结构体变量的访问是通过对结构体变量各成员的引用来实现的，对于结构体变量和结构体指针来说，引用成员的方法不同，有以下三种形式：

重点

- (1) 结构体变量名. 成员名
- (2) 结构体指针变量名->成员名
- (3) (*结构体指针变量名). 成员名

例如：上例中引用“age”这个成员可以写成

方法一： $x.age$

方法二： $p \rightarrow age$

方法三： $(*p).age$

西南林业大学计信学院

10.1.4 结构体成员的引用、赋值

说明：

(1) 结构体成员本身是什么类型，引用后也是什么类型。例如：“ $x.name$ ”代表一个字符数组的首地址；“ $x.sex$ ”代表一个字符型的变量；“ $x.age$ ”代表一个整型变量。

(2) 对成员变量的使用与普通变量完全相同，可根据其数据类型进行相应的运算。

(3) 如果成员本身又是一个结构体类型时，则必须逐级找到最低一级的成员才能使用。

西南林业大学计信学院

10.1.4 结构体成员的引用、赋值

以下是为结构体成员赋值的几种方法：

方法一：定义结构体变量的同时使用初始化方法为它的各个成员赋初值

```
typedef struct stu
{ char name[30];
  char sex;
  int age;
  double score;
}STU;
STU x = {"LiPin", 'M', 19, 80};
```

初始化

西南林业大学计信学院

10.1.4 结构体成员的引用、赋值

续上

方法二：将一个已经赋过初值的结构体变量整体赋值给另一个同类型的结构体变量

```
struct stu
{ char name[30];
  char sex;
  int age;
  double score;
}x = {"LiPin", 'M', 19, 80}, y;
y = x;
```

结构体变量 x 已经有初值

赋值号的左右两侧是同类型的结构体变量，因此可以相互赋值

西南林业大学计信学院

10.1.4 结构体成员的引用、赋值

续上

方法三：使用赋值语句对结构体各成员分开

```
struct stu
{  char name[30];
    char sex;
    int  age;
    double score;
} x, *p = &x;
```

注：不能写成 `x.name = "LiPin";`

```
strcpy(p->name, "LiPin");
p->sex = 'M';
p->age = 19;
p->score = 80;
```

指针名->成员名



续上

10.1.4 结构体成员的引用、赋值

方法四：从键盘输入为各结构体成员赋值

```
struct stu  
{  char name[30];  
    char sex;  
    int  age;  
    double score;  
} x, *p = &x;
```

```
gets((*p).name);  
scanf("%c", &(*p).sex);  
scanf("%d", &(*p).age);  
scanf("%lf", &(*p).score);
```

(*指针名).成员名

在C语言中，可以将几种不同类型的变量存放在同一段内存空间里，称这种结构为“共用体”。

“共用体”与“结构体”有一些相似之处，但两者有本质上的不同。

(1) 在结构体中各成员有各自的内存空间，一个结构变量的总长度是各成员长度之和。

(2) 在共用体中，各成员共享一段内存空

10.2 共用体

一、共用体类型的说明

关键字 说明一个共用体类型的一般形式为：

```
union <共用体名>
{  类型说明符1  成员名1;
  类型说明符2  成员名2;
  .....
};
```

该共用体类型中有三个成员，它们使用同一段存储空间，即20个字节

例如：

```
union stu
{  char name[20];
  char sex;
  int  age;
};
```

10.2 共用体

二、共用体变量的定义

方法一：先说明共用体类型，再定义变量。

```
union stu
{  char name[20];  double score;  };
union stu x;  变量
```

方法二：说明共用体类型的同时使用typedef，然后再定义变量。

```
typedef union stu  类型
{  char name[20];  double score;  }STU;
STU x;  变量
```

西南林业大学计信学院

10.2 共用体

续上

方法三：先说明共用体类型，再使用typedef，最后定义变量。

```
union stu
{  char name[20];  double score;  };
typedef union stu STU;  类型
STU x;  变量
```

方法四：说明共用体类型的同时定义变量。

```
union stu  变量
{  char name[20];  double score;  } x;
```

西南林业大学计信学院

10.2 共用体

三、共用体成员的引用及赋值

对共用体变量的赋值、使用都只能针对共用体变量的成员进行。成员的引用形式如下：

共用体变量名.成员名

说明：

- (1) 一个共用体变量，每次只能赋一个成员值。换句话说，一个共用体变量的值就是共用体变量某一个成员的值。
- (2) 共用体变量中起作用的成员是最后一次存放的成员。
- (3) 不允许对共用体变量作初始化赋值，赋值只能在程序中进行。

西南林业大学计信学院

10.2 共用体——练习题

(1) 若有以下定义和语句

```
union data
{  int i;  char c;  float f;  }x;
int y;
则以下语句正确的是 ( B )
A) x=10.5          B) x.c=101;
C) y=x;            D) printf("%d\n",x);
```

【分析】x是共用体变量，不能直接使用或对其赋值，只能通过引用它的成员来实现。因此选项A、C、D的写法都是错误的。

西南林业大学计信学院

10.2 共用体——练习题

(2) 有以下程序

```
main()
{  union {  unsigned int n;
          unsigned char c;  }u1;
  u1.c = 'A';
  printf("%c", u1.n);
}
```

通过赋值语句 u1.c = 'A'; 则u1.n 中的也就相当于65。而printf语句是用字符形式打印u1.n 中的值，因此打印A。

执行后输出结果是 (C)

A) 产生语法错 B) 随机值 C) A D) 65

【分析】u1是共用体变量，它有两个成员——n 和 c，由于所有成员共用一段存储空间，因此为其中一个变量赋值，其它变量的值与该值相同。

西南林业大学计信学院

10.2 共用体——练习题

(3) 有以下定义

```
union data
{  int d1;  float d2;  }demo;
```

则下面叙述中错误的是 (D)

- A)变量demo与成员d2所占的内存字节数相同
B)变量demo中各成员的地址相同
C)变量demo和各成员的地址相同

D)若给demo.d1赋99后，demo.d2中的值是99.0

【分析】共用体变量，以及共用体变量的各成员占用同一段存储空间，因此它们的地址相同。虽然为共用体变量的一个成员赋值，其它成员也就相当于同时被赋值，但由于实型数据的存储模式与整型或字符型不同，因此

西南林业大学计信学院

