```
第十章 结构体与共用体
            10.1 结构体
程
            10.2 共用体
序
设
计
       结构体、共用体类型都是用户自己定
基
   义的数据类型,这些类型的数据是用户根
础
   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
            10.1 结构体
                            难点
     重点
     10.1.1 结构体类型的说明
C
     10.1.2 结构体变量的定义
程
序
     10.1.3 结构体类型指针的定义
设
     10.1.4 结构体成员的引用、赋值
计
基
     10.1.5 结构体数组
础
     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
\mathbf{C}
   定,类型相同或不同的若干有序变量组成的集
程
   合。组成结构体的每个数据都称为结构体的"成
序
   员",或称"分量"。
设
      结构体类型说明的形式如下:
计
      struct [结构体标识名]
                     标识名可以省略
基
础
             结构体成员名列表1;
             结构体成员名列表2;
       类型名n 结构体成员名列表n;
          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
               该结构体名称为 struct data
    struct data
C
    { int year;
程
     int month;
               该结构体有三个成员、分别
序
     int day;
                 是 year、month、day
设
    };
   说明:
计
    (1) 结构体类型的说明可以嵌套。
基
    (2) struct是关键字。"结构体标识名"可以省略;右
础
   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
(Vis
   兄,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。
6.0
                        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
C
     typedef是一个关键字,利用它可以将已存在的
程
   数据类型命一个新的名称,可以理解为给已有的数
序
   据类型取一个别名。
设
计
   例如: typedef int INT; //INT 相当于是 int 的别名
基
    INT x, y; //等价于 int x, y;
础
     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。
(Vis
     由于结构体类型通常都比较复杂,利用tryedef
   可以も有な的は物体米刑会―へ答番<del>感性を学<sup>汁信学院</sup></del>
       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
C
   struct date
程
     int year; int month; int day; };
   typedef struct data DATE;
序
设
  方法二:说明结构体类型的同时就使用typedef
计
   typedef struct date
基
   { int year; int month; int day; } DATE;
础
》
  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;
       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
C
   义该类型对应的"结构体变量"。
程
序
      前面提到,结构体类型说明后,系统只是认
设
   可有这样一种用户自己构造的数据类型,但并
计
   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
基
   空间。
础
      结构体变量占用的存储容量由结构体类型决
定。
(Vis
      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
----
       10.1.2 结构体变量的定义
  重点
   方法一: 先说明结构
                   方法二: 使用typedef直
C
   体类型, 然后定义结
                   接为结构体类型命名,
程
   构体变量
                   然后定义结构体变量
序
   struct date
                   typedef struct date
设
计
                     int year;
     int year;
基
     int month;
                     int month;
础
                     int day;
     int day;
                   }DATE; _ 结构体类型名
》
   };
                   DATE x, y;
(Vis
   stuct date x, y;
                        结构体变量名
       10.1.2 结构体变量的定义
  续上
   方法三: 先说明结构体类
                    方法四: 说明结构体
                    类型的同时, 就定义
   型,然后使用typedef为结
C
程
   构体类型重命名, 最后定
                    结构体变量
   序
   struct date
                    struct date
设
   {
                    {
计
    int year;
                      int year;
基
    int month;
                      int month;
础
    int day;
                      int day;
》
          结构体类型名
   };
(Vis
                    }x, y;
   typedef stuct date DATE;
   DATE x, y; 一结构体变量名
6.0
       10.1.2 结构体变量的定义
  练习题
     (1) 以下叙述不正确的是 (R)
\mathbf{C}
程
   struct ex
序
   { int x; float y; char z; } example;
设
计
   A) struct结构体类型的关键字
基
   B) example是结构体类型名
础
>>
   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(
C
程
    typedef struct
    { int n;
序
设
     char ch[8];
计
    }PER;
基
    A) PER 是结构体变量名
础
    B) PER是结构体类型名
>>
    C) typedef struct 是结构体类型
(Vis
    D) struct 是结构体类型名
6.0
                        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。 程 需要注意的是,结构体指针定义后,与普通类 序 型的指针一样,必须将其指向有效的存储空间,然 设 后才能使用一县空指针是不能使用的。 计 例如: typedef struct stu { char name[30]; 以上说明了一个 基 名称为STU的结构体 char sex; 础 类型、并定义了该结 int age; double score; 构体类型的变量x及 指针p,并且对指针进 STU x, \*p = &x; 行初始化址大学计信学院

```
«
                 有了以上定义, 系统便根
   typedef struct stu
               据结构体类型各成员占用的存
C
   { char name[30];
程
               储单元总和为结构体变量 x 分
    char sex;
序
    int age;
               配 30+1+4+8=43 个字节的存
设
    double score;
               储空间。
计
   }STU;
                  指针p初始化为指向变量
   STU x, *p = &x;
基
               x, 即指向这43个字节存储空
础
     间的首地址。
该结构体变量 x 代表一名学生的信息,它由4
》
(Vis
  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,<u>以下介绍引用方法。</u>
一种两种业大学计信学院
     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
\mathbf{C}
   针来说,引用成员的方法不同,有以下三种形式:
程
序
        (1) 结构体变量名. 成员名
设
   重点
        (2)
           结构体指针变量名->成员名
计
        (3) (*结构体指针变量名).成员名
基
础
   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->age
                 西南林业大学计信学院
     方法三: (*p).age
     10.1.4 结构体成员的引用、赋值
   说明:
C
    (1) 结构体成员本身是什么类型,引用后也
程
   是什么类型。例如: "x.name"代表一个字符
序
   数组的首地址; "x.sex"代表一个字符型的变
设
计
   量; "x.age"代表一个整型变量。
基
    (2) 对成员变量的使用与普通变量完全相
础
   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。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:
\mathbf{C}
   方法一:定义结构体变量的同时使用初始化
程
         方法为它的各个成员赋初值
序
       typedef struct stu
设
        char name[30];
计
         char sex;
基
         int
            age;
础
         double score;
STU;
                初始化
(Vis
       STU x = {\text{"LiPin", 'M', 19, 80}};
          结构体成员的引用、赋值
   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
C
程
        变量
序
     struct stu
设
       char name[30];
计
       char sex;
基
       int
           age;
础
                 结构体变量 x 已经有初值
       double score;
     x = {\text{"LiPin"}, 'M', 19, 80}, y;
(Vis
     y = x;
            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
    10.1.4 结构体成员的引用、
```

10.1.3 结构体类型指针的定义

```
struct stu
程
       char name[30];
序
        char sex;
设
        int
             age;
计
        double score:
基
     x, *p = &x;
                     注: 不能写成 x.name = "LiPin";
础
     strcpy(p->name, "LiPin");
     p->sex = 'M';
(Vis
     p->age = 19;
                      指针名->成员名
C++
     p->score = 80;
6.0
```

「「 「方法四: 从键盘输入为各结构体成员赋值

```
struct stu
程
         char name[30];
序
         char sex;
设
         int
              age;
计
         double score;
      x, *p = &x;
基
础
      gets((*p).name);
      scanf("%c", &(*p).sex);
      scanf("%d", &(*p).age); (*指针名).成员名
(Vis
      scanf("%lf", &(*p).score);
C++
```

10.1.4 结构体成员的引用、赋值 练习题 C (1) 设有如下定义: 程 struct sk 序 { int a; 设 float b; 计 }data; 基 int \*p; 础 若要使p指向data中的a域,正确的赋值语句是() A) p=&a; B) p=data.a; (Vis C) p=&data.a; D) \*p=data.a 西南林业大学计信学院 10.1.5 结构体数组 一、结构体数组的定义 C 问题: 以上例题中, 定义一个 struct stu 类型 程 的结构体变量 x 仅能保存一名学生的信息, 若要 序 保存多名学生的信息,应如何定义变量? 设 应引入"数组"的概念,即定义一个 struct stu 计 类型的数组,数组中的每个元素保存一名学生信 基 typedef struct stu { char name[30]; char sex; 础 **>>** int age; double score; (Vis } STU:  $\mathbf{STU} \mathbf{x}[\mathbf{10}];$  //定义一个结构体数组 $\mathbf{x}$ ,保存 $\mathbf{10}$ 个学生 西南林业大学计信学院 10.1.5 结构体数组 二、数组元素及结构体成员的引用 C 定义了结构体数组后,涉及两个层面的引用: 程 是数组元素的引用;二是结构体成员的引用。 序 typedef struct stu 设 { char name[30]; char sex; int age; double score; 计 基 } STU: 础 STU x[10]; //定义一个结构体数组x 例如以上定义的结构体数组x, 有10个数组元 (Vis 素 x[0]~x[9], 每 个 数 组 元 素 由 四 个 成 员name、sex、age、score组成,以下介绍引用方 6.0 10.1.5 结构体数组 例如以上定义的结构体数组x,有10个数组元 素 x[0]~x[9], 每 个 数 组 元 素 由 四 个 成 C 员name、sex、age、score组成,以下介绍引用方 程 法。 序 如果需要引用第i个学生的四个成员,分别赋值 设 <del>为 "LiPin",'M',<u>19</u>,80 方法为:</del> 计 思考: x[i].name前面要加 & 基 scanf("%s", x[i].name);//引用成员name 础 x[i].sex = 'M'; //引用成员sex **》** x[i].age = 19; //引用成员age (Vis x[i].score = 80; //引用成员score <del>西南林业人学计信</del>学院 10.1.5 结构体数组 三、结构体数组的初始化 C typedef struct stu 程 { char name[30]; char sex; 序 int age; double score; 设 } STU; 计 //定义一个结构体数组 x 并对其进行初始化 基 STU  $x[4] = \{\{\text{"LiPin"}, \text{'M'}, 18, 80\},\$ 础 {"LinYi", 'M', 19, 85}, {"SunDan", 'W', 18, 90}, {"WuLi", 'W', 18, 70}}; 西南林业大学计信学院 10.1.5 结构体数组 【例1】定义一个大小为4的结构体数组,存放4名学 生的信息,查找这4名学生中的最高分,并打印该同 C 学的所有信息。 程 序 typedef struct stu { char name[30]; char sex; int age; double so 设 int age; double score; 计 } STU; 基 main() 础 **STU**  $x[4] = \{\{\text{``LiPin''}, \text{'M'}, 18, 80\},$ {"LinYi", 'M', 19, 85}, {"SunDan", 'W', 18, 90}, {**"WuLi"**, **'W'**, **18**, **70**}}; 西南林业大学计信学院 10.1.5 结构体数组 绿上 double max s; int i, max n;  $\mathbf{C}$ 程 max s = x[0].score; //结构体成员的引用  $\max n = 0$ ; 序 for(i=1; i<4; i++) 设 计 if(x[i].score > max s)基 { 础  $\max s = x[i].score;$ max n = i;} 西南林业大学计信学院 10.1.5 结构体数组 printf("最高分为: %lf\n", max s); printf("最高分的学生信息:"); 程 printf("姓名: %s\n", x[max\_n].name); 序 printf("姓别: %c\n", x[max\_n].sex); 设 printf("年龄: %d\n", x[max\_n].age); 计 printf("分数: %lf\n", x[max n].score); 基 础 } 西南林业大学计信学院 10.1.6 结构体与函数实参 结构体变量、结构体指针、结构体数组名,结 构体成员都可以作函数实参,遵循的仍然是"传值" C 或者"传地址"的方式。 程 序 结构体变量作实参 设 当结构体变量作实参时,实现的是"传值"方式, 计 对应的形参也是相同结构体类型的变量。 基 础 结构体指针作实参 当结构体指针作实参时,实现的是"传地址"方 对应的形参是相同结构体类型的指针变量。 西南林业大学计信学院 10.1.6 结构体与函数实参 续上 C 三、结构体数组名作实参 程 当结构体数组名作实参时,实现的是"传地址" 序 方式、对应的形参是相同结构体类型的指针变量。 设 计 四、结构体成员作实参 基 当结构体成员作实参时,要根据具体的成员类 础 型来确定是实现"传值"还是"传地址"方式?如果是 **》** 普通类型的成员,则实现"传值"方式;如果是指针 (Vis 类型的成员或者数组名等,则实现"传地址"方式。 西南林业大学计信学院 10.1.6 结构体与函数实参 (1) 以下程序的运行结果是( ) { char name[10]; int num; }; C void f1(struct STU c) 程 { struct STU b={"LiPin", 2042}; c=b; } 序 void f2(struct STU \*c) 设 struct STU b={"SunYi", 2044}; \*c=b; } 计 main() \_struct STU a={"LiYan",2041}, b={"QiYin",2043}; 基 传值 (a); f2(&b); printf("%d %d\n", a.num, s.num); 础 **>>** A) 2041 2044 B) 2041 2043 C) 2042 2044 D) 2042 2043 (Vis 西南林业大学计信学院 10.1.7 链表 链表是一种常见的、重要的数据结构,它通常采 用动态的分配办法为一个结构体分配内存空间。 C 程 一方面<mark>需要时就分配</mark>一块空间用来存放,从而节 序 约了宝贵的内存资源;且便于删除与加入。 设 另一方面,在动态分配时,结点与结点之间可以是 计 不连续的,结点之间的联系通过指针来实现。即第一 基 个结点中存放第二个结点的地址,第二个结点中存放 础 第三个结点的地址, ....., 最后一个结点因无后续结 **》** 点连接,其指针域可赋为NULL。 (Vis 这样一种连接方式,如同一条一环扣一环的链 班表 例如:一个存放学生学号和成绩的结点可以定 **«** 义为以下结构: C 程 struct stu num[20]| score | next 序 char num[20]; 设 数据域 指针域 int score: 计 struct stu \*next; 基 **}**; 础 在该结构体中前两个成员项组成数据域、最后 一个成员项next构成指针域,它是一个指向struct stu类型的结构指针变量。 西南林业大学计信学院 10.1.7 链表 对链表的主要操作有: ① 建立链表; ②输出 链表;③链表的删除操作;④链表的插入操作 C 程 (1) 建立链表: 就是指从无到有地建立起一个链 序 表,即首先分配每个结点的存储空间,并为每个结点 设 的数据域输入数值,同时建立起前后相邻结点的连接 计 指针域 数据域 指针域空 础 3 头结点 结点1 结点2 末尾结点 数据域空 西南林业大学计信学院 10.1.7 链表 续上 **«** (2) 输出链表: 就是指链表的第一个结点开始,将 C 每个结点的数据域输出。链表输出时,首先要知道第 程 一个结点的地址,通常是通过一个"头结点"获得的, 序 当搜索到链表的最后一个结点时,输出工作完成。 设 计 基 础 西南林业大学计信学院 10.1.7 链表 绿上 (3) 链表的删除操作: 从一个链表中删除一个结  $\mathbf{C}$ 点,并不是真正从内存中把它抹去,而是把它从链表 程 中分离开来,可以通过改变链表的链接关系完成。 序 <u>例如:将以下链表中的结点2删除。</u> 设 计 基 础 p1(Vis 以上示图中、结点1的指针域初始时指向结点2、 修改为指向结点3后,结点2便从链表中删除掉了。 10.1.7 链表 续上 (4) 链表的插入操作: 在一个链表的指定位置插入 首先要求链表本身必须是已经按某种规律排好 结点, 序的。按照排序的规律确定好新结点在链表中的位 程 置,并将其插入链表中即可。 序 例如:将数据3所在的新结点插入到以下链表的结 设 点1和结点2中间,使<mark>链表仍保持升序排序。</mark> 计 新结点 基 础 建立、 链表 输出 **«** 【例2】建立一个动态链表,链表中有10个数据结 C 点,向每个结点的数据域中填入一个整数值, 程 出该链表。 序 分析: 设 (1) 链表中的每个结点使用动态存储分 计 基 通常有一个"头结点 指针域指向"结点1"。 建立链表时, 础 **》** (Vis 链表 "末尾结点"的指针域为NULL,即不 ual 指冋任何有效的存储空间,标志链表的结束。 6.0 10.1.7 建立、 #include <stdio.h> #include <stdlib.h> C typedef struct node 程 data; //结点的数据域 序 struct node \*next; //结点的指针域 **}NODE**; //结构体名 设 main() 计 { //h指向头结点,p1指向前一个结点,p2指向后一个结点 基 **NODE** \*h, \*p1, \*p2; 础 int i; h = (NODE \*)malloc(sizeof(NODE)); //头结点 建立 p1 = h: Yor(i=0; i<10; i++) //构造10个结点,并连接成链表 6.0 西南林业大学计信学院 C:\Documents and Settings\lj... -□× 续上 p2 = (NODE \*)malloc(sizeof(NODE)); //新结点 p2->data = 2 \* i + 1; //填充新结点的数据域 C p1->next = p2; //使前一个结点的指针域指向新结点 程 p1 = p2; //p1始终指向链表当前的最末-序 p1->next = NULL; //填充链表末尾结点的指针域 p1 = h->next; //指向链表的"结点1",而非"头结点" while(p1 != NULL) //判断链表是否结束 printf("%d -> ", p1->data); //输出结点的数据域 础 p1 = p1->next; //使p1指向下一个结点 (Vis printf("NULL \n"); 6.0 西南林业大学计信学院 10.1.7 链表 **«** 【例3】建立一个具有10个结点的链表,假定链表中 C 的各结点已按升序排序,并且各结点中的数值不重 程 复。从键盘输入一个整数,查找链表中是否有该 序 数,如果有,就将链表中存放该数的结点删除。 设 分析: 计 (1) 5先要在链表中查找一个指定数值,如 便定位了该数值所在的结点。同时需要 点前,后两个结点的位置,例如用指针 基 础 位该结点前、 **》** p1、p2分别指向前、后的两个结点。 将p1的指针域修改为指向p2所指的约 (Vis 6.0 西南林业大学计信学院 10.1.7 链表 续上 #include <stdio.h> #include <stdlib.h> typedef struct node int data;//数据域 序 struct node \*next; //指针域 **NODE**; 设 main() 计 { 基 **NODE** \*h, \*p1, \*p2; 础 int i, x,  $a[10] = \{1,2,3,4,5,6,7,8,9,10\};$ h = (NODE \*)malloc(sizeof(NODE)); //头结点 **>>** p1 = h; (Vis for(i=0; i<10; i++) //建立链表(有10个结点) 6.0 西南林业大学计信学院 10.1.7 链表——删除 续上 p2 = (NODE \*)malloc(sizeof(NODE)); p2->data = a[i];C p1->next=p2;程 p1 = p2;序 p1->next = NULL; //建立链表到此结束 设 计 printf("请输入一个整数: "); 基 scanf("%d", &x); 础 **p1** = h; //p1指向头结点 while(p1 != NULL) (Vis **p2** = p1->next; //p2指向p1之后的结点 6.0 西南林业大学计信学院 C:\Documents and Settings\ljq\Debug\t1.exe -> 5 -> 6 -> 7 -> 8 -> 9 -> 10 -> NULL : -> 6 -> 7 -> 8 -> 9 -> 10 -> NULL if(p2->data == x) //查找是否有指定数值  $\mathbf{C}$ break: 程 p1 = p1 - next;序 p1->next = p2->next; //删除指定数值的结点 设 计 p1 = h - next: 基 while(p1 != NULL) //输出删除结点后的链表 础 printf("%d ", p1->data); **》** p1 = p1 - next; (Vis } 西南林业大学计信学院 10.1.7 链表—— 【例4】建立一个具有10个结点的链表,假定链表中 C 的各结点已按升序排序,并且各结点中的数值不重 程 复。从键盘输入一个整数,将该数插入到链表中, 序 使该链表各结点仍然保持升序排序。 设 分析: 计 (1) 这是一个"链表插入"操作的题目。 基 (2) 与删除操作类似,编程时应设置两个指针 础 p1、p2,让它们指向插入结点前、后的结点。 **》** 插入新结点前,首先要进行比较, 数据依次与各结点的数据域比较, (Vis 据域的值大于待插数据时,循环结束。 <del>西南林业大学计信学院</del> 10.1.7 链表-续上 #include <stdio.h> C #include <stdlib.h> 程 typedef struct node 序 int data;//数据域 设 struct node \*next; //指针域 计 **NODE**; main() 基 { 础 NODE \*h, \*p1, \*p2, \*p3; int i, x,  $a[10] = \{1,3,5,7,9,11,13,15,17,19\}$ ; (Vis h = (NODE \*)malloc(sizeof(NODE)); //头结点 p1 = h;西南林业大学计信学院 10.1.7 链表——插入 续上 for(i=0; i<10; i++) //建立链表(10个结点) C p2 = (NODE \*)malloc(sizeof(NODE)); 程 p2->data = a[i];序 p1->next=p2;设 p1 = p2;计 p1->next = NULL; //建立链表到此结束 基 础 printf("请输入一个整数:"); scanf("%d", &x); (Vis p3 = (NODE \*)malloc(sizeof(NODE)); p3->data = x; 西南林业大学计信学院 链表— 10.1.7 续上 **p1** = h; //p1指向头结点 while(p1 != NULL) C 程 <mark>p2</mark> = p1->next; //p2指向p1之后的结点 序 **if(p2->data > x)** //查找是否有指定数值 设 break; 计 p1 = p1 - next;基 础 p1->next = p3; //插入新结点 p3->next=p2;p1 = h - next;6.0 西南林业大学计信学院 10.1.7 链表— 续上 while(p1!=NULL)//输出插入新结点后的链表 printf("%d ", p1->data); 程 p1 = p1 - next;序 设 } 计 "C:\Documents and Settings\ljq\Debug\t1.exe" \_ 🗆 × 基  $\rightarrow$  9  $\rightarrow$  11  $\rightarrow$  13  $\rightarrow$  15  $\rightarrow$  17  $\rightarrow$  19  $\rightarrow$  NULL 础 > 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> NULL (Vis 西南林业大学计信学院 10.2 共用体

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 程 序 体"。 设 "共用体"与"结构体"有一些相似之处,但两 计 者有本质上的不同。 基 (1) 在结构体中各成员有各自的内存空间, 础 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



