

第四章 C语言的补充概念

模块4.1: 共用体

4.1 共用体



例:定义一个用于一卡通管理系统的结构,要求包含卡号、余额、消费限额、消费密码等公共信息,此外,若持卡人是学生,要包含学号、姓名、专业等学生特有的信息,若持卡人是教师,则包含工号、姓名、职称等教师特有的信息

```
struct student{
   定义学生信息;
struct teacher {
   定义教师信息;
struct ykt {
   公共信息;
   student sinfo;
   teacher tinto;
int main()
   ykt y1;//定义变量
```

```
对y1的成员的访问:
int main()
  ykt y1;
  y1. 卡号
  yl. sinfo. 学号
  yl. tinfo. 工号
  return 0;
```

能否使sinfo/tinfo共用一段空间:

- 1) 当持卡人是学生时,这段空间按 student方式访问
- 2) 当持卡人是教师时 , 按teacher 方式访问
- =>(共用体)

4.1 共用体



```
      union 共用体名 {
      union data {

      共用体成员1 (类型名 成员名)
      short a;

      long b;
      thar c;

      计用体成员n (类型名 成员名)
      char c;

      };
```

- 所有成员从同一内存开始,共用体的大小为其中占用空间最大的成员的大小
- 给一个共用体成员赋值后,会覆盖其它成员的值,因此只有最后一次存放的成员是有效的
- 其它所有定义、使用方法同结构体

```
#include <iostream>
using namespace std;
                                                     struct data1 d1:
struct data1 {
                                                           2000
    short a;
                                                      d1
                                                           2001
   long b; //12:所有成员所占空间之和(含填充字节)
                                                           2002
                                                           2003
   char c;
                                                           2004
                                                           2005
                                                           2006
                                                                 C
union data2 {
   short a;
                                                     union data2 d2;
  long b; //4:所有成员中最大成员所占空间
                                                          3000
                                                                   C
                                                      d2
                                                          3001
  char c;
                                                          3002
                                                          3003
int main()
    cout << sizeof(data1) << ' ' << sizeof(data2) << end1; //12 4
```

```
#include <iostream>
using namespace std;
union data {
   int a;
                                  70000=00000000 00000001 00010001 01110000
   short b;
                                               d:低位在前存放
   char c;
                                                    01110000
                                               2000
                                                    00010001
                                               2001
int main()
                                                    00000001
                                               2002
                                               2003
                                                    00000000
    union data d;
    d. a=70000;
    cout << d.a << ' ' << d.b << ' ' << d.c << endl;
                                                          70000 4464 p
    d. b=7000;
    cout << d.a << ' ' << d.b << ' ' << d.c << endl; 72536 7000 X
    d. c=' A':
    cout << d.a << ' ' << d.b << ' ' << d.c << end1; 72513 6977 A
    return 0;
```

```
#include <iostream>
                            72536=00000000 00000001 00011011 01011000
using namespace std;
union data {
                                              d:低位在前存放
   int a;
                                                   01011000
                                              2000
   short b;
                                                   00011011
                                              2001
   char c;
                                                   0000001
                                              2002
                                                   00000000
                                              2003
int main()
                                                7000=00011011 01011000
    union data d;
    d. a=70000;
    cout << d.a << ' ' << d.b << ' ' << d.c << endl;
                                                         70000 4464 p
    d. b=7000;
    cout << d.a << ' ' << d.b << ' ' << d.c << endl;
                                                          72536 7000 X
    d. c='A';
    cout << d.a << ' ' << d.b << ' ' << d.c << endl:
                                                         72513 6977 A
    return 0;
```

```
#include <iostream>
                           72513=0000000 00000001 00011011 01000001
using namespace std;
union data {
                                                d:低位在前存放
   int a;
                                                    01000001
                                                2000
                                                                С
   short b;
                                                    00011011
                                                2001
   char c;
                                                2002
                                                    00000001
                                                    00000000
                                                2003
int main()
                                                         A=0100 0001
    union data d;
    d. a=70000;
    cout << d.a << ' ' << d.b << ' ' << d.c << endl; 70000 4464 p
    d. b=7000;
    cout << d.a << ' ' << d.b << ' ' << d.c << end1; 72536 7000 X
    d. c=' A':
    cout << d.a << ' ' << d.b << ' ' << d.c << endl:
                                                        72513 6977 A
    return 0;
```

• 所有成员从同一内存开始,共用体的大小为其中占用空间最大的成员的大小

```
#include <iostream>
using namespace std;
union data {
 int a;
 short b;
 char c;
int main()
{ union data d;
   d. c=' A':
```

d. c=' A'	
2000	01000001
2001	???
2002	???
2003	???

d. b=7000	
2000	01011000
2001	00011011
2002	???
2003	???

d. a=70000		
2000	01110000	
2001	00010001	
2002	00000001	
2003	00000000	

```
union data d;
d. c='A';
cout << d. a << ' ' << d. b << ' ' << d. c << endl; //不确定 不确定 A
d. b=7000;
cout << d. a << ' ' << d. b << ' ' << d. c << endl; //不确定 7000 X
d. a=70000;
cout << d. a << ' ' << d. b << ' ' << d. c << endl; //70000 4464 p
return 0;
```

4.1 共用体

```
struct student {
 定义学生信息
struct teacher {
 定义教师信息;
struct ykt {
 公共信息:
 student sinfo; 空间
 teacher tinto; 浪费
int main()
{ ykt y1;//定义变量
```

```
struct student {
 定义学生信息;
struct teacher {
 定义教师信息;
union owner {
 student s;
          此处保证s/t
 teacher t;
           共用一段空间
struct ykt {
 公共信息:
 char type; //持卡人类别
 owner info;
```

```
int main()
  ykt y1;//定义变量
  y1. 卡号...;
  if (y1. type=='s') {
    y1. info. s. 学号;
  else{
    y1. info. t. 工号;
  return 0;
```