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
    y1.sinfo.学号
    y1.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

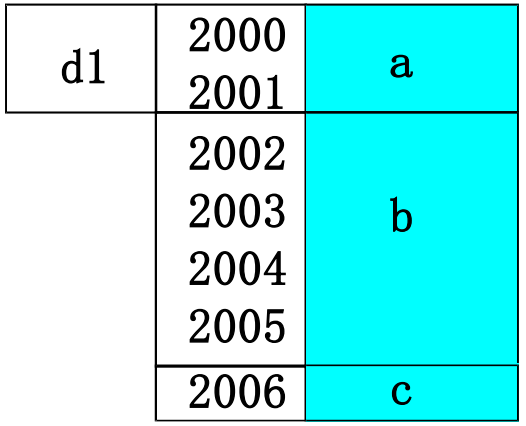
```
union 共用体名 {
    共用体成员1（类型名 成员名）
    ...
    共用体成员n（类型名 成员名）
};
```

```
union data {
    short a;
    long b;
    char c;
};
```

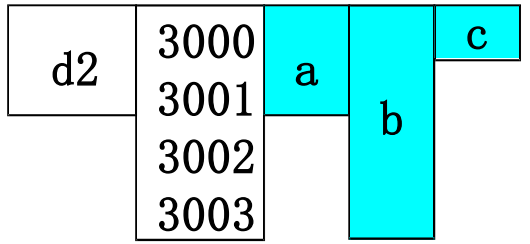
- 所有成员从同一内存开始，共用体的大小为其中占用空间最大的成员的大小
- 给一个共用体成员赋值后，会覆盖其它成员的值，因此只有最后一次存放的成员是有效的
- 其它所有定义、使用方法同结构体

```
#include <iostream>
using namespace std;
struct data1 {
    short a;
    long b;    //12:所有成员所占空间之和(含填充字节)
    char  c;
};
union data2 {
   short a;
   long b;     //4 :所有成员中最大成员所占空间
   char  c;
};
int main()
{    cout << sizeof(data1) << ' ' << sizeof(data2) << endl;  //12 4
}
```

struct data1 d1;

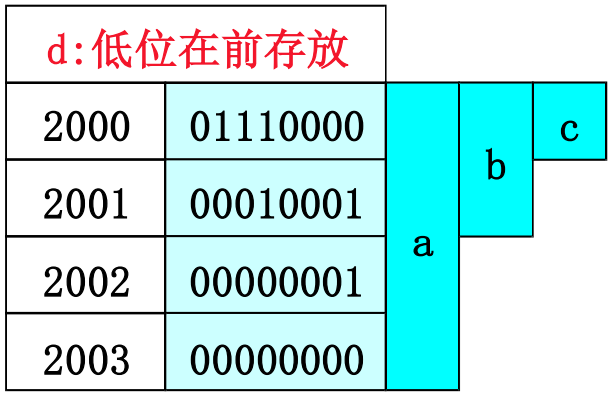


union data2 d2;



```
#include <iostream>
using namespace std;
union data {
    int a;
    short b;
    char  c;
};
int main()
{
     union data d;
     d.a=70000;
     cout << d.a << ' ' << d.b << ' ' << d.c << endl;
     d.b=7000;
     cout << d.a << ' ' << d.b << ' ' << d.c << endl;
     d.c='A';
     cout << d.a << ' ' << d.b << ' ' << d.c << endl;
     return 0;
}
```

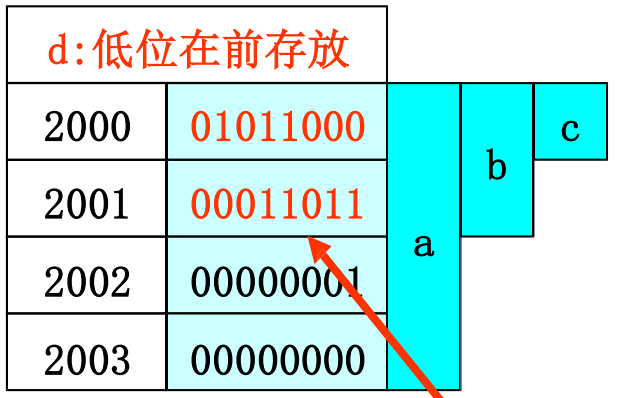
70000=00000000 00000001 00010001 01110000




```
70000 4464 p
72536 7000 X
72513 6977 A
```

```
#include <iostream>
using namespace std;
union data {
    int a;
    short b;
    char  c;
};
int main()
{
     union data d;
     d.a=70000;
     cout << d.a << ' ' << d.b << ' ' << d.c << endl;
     d.b=7000;
     cout << d.a << ' ' << d.b << ' ' << d.c << endl;
     d.c='A';
     cout << d.a << ' ' << d.b << ' ' << d.c << endl;
     return 0;
}
```

72536=00000000 00000001 00011011 01011000



7000=00011011 01011000

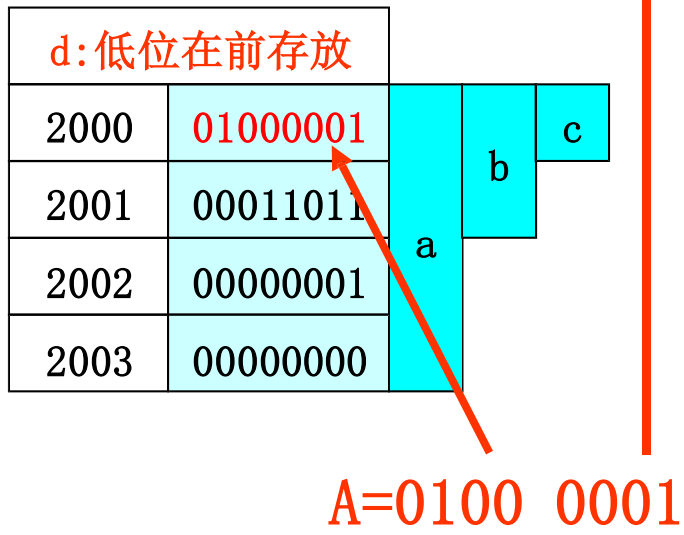
70000 4464 p

72536 7000 X

72513 6977 A

```
#include <iostream>
using namespace std;
union data {
    int a;
    short b;
    char  c;
};
int main()
{
     union data d;
     d.a=70000;
     cout << d.a << ' ' << d.b << ' ' << d.c << endl;
     d.b=7000;
     cout << d.a << ' ' << d.b << ' ' << d.c << endl;
     d.c='A';
     cout << d.a << ' ' << d.b << ' ' << d.c << endl;
     return 0;
}
```

72513=00000000 00000001 00011011 01000001



70000 4464 p

72536 7000 X

72513 6977 A

• 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';
   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;
}
```

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

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;
  teacher tinto;
};
int main()
{  ykt y1;//定义变量
}
```

空间浪费

```
struct student {
  定义学生信息;
};
struct teacher {
  定义教师信息;
};
union owner {
  student s;
  teacher t;
};
struct ykt {
  公共信息;
  char type; //持卡人类别
  owner info;
};
```

此处保证s/t共用一段空间

```
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;
}
```