

## C++Union联合体

利用union可以用相同的存储空间存储不同型别的数据类型，从而节省内存空间。当访问其内成员时可用"."和">"来直接访问。

当多个数据需要共享内存或者多个数据每次只取其一时，可以利用联合体(union)。在C Programming Language一书中对于联合体是这么描述的：

- 联合体是一个结构；
- 它的所有成员相对于基地址的偏移量都为0；
- 此结构空间要大到足够容纳最"宽"的成员；
- 其对齐方式要适合其中所有的成员；

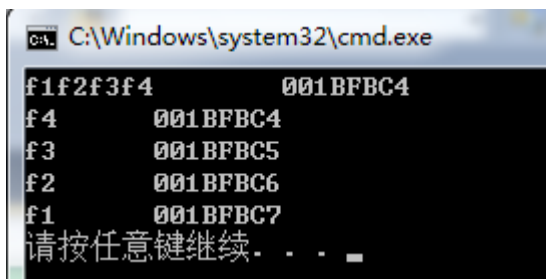
```
#include<iostream>
using namespace std;

union U
{
    unsigned int n;
    unsigned char s[4];
};

int main()
{
    U u;
    u.n = 0xf1f2f3f4;
    cout<<hex<<u.n<<'\t'<<&u.n<<endl;
    cout<<hex<<(int)u.s[0]<<'\t'<<(void*)&u.s[0]<<endl;
    cout<<hex<<(int)u.s[1]<<'\t'<<(void*)&u.s[1]<<endl;
    cout<<hex<<(int)u.s[2]<<'\t'<<(void*)&u.s[2]<<endl;
    cout<<hex<<(int)u.s[3]<<'\t'<<(void*)&u.s[3]<<endl;
}
```

上述代码中，联合体U中，n占4字节，s[4]也占4字节。故联合体共占4字节。

程序输出如下：



这说明，内存中数据低位字节存入低地址，高位字节存入高地址，而数据的地址采用它的低地址来表示。