



本节主题:

联合体(共用体)

联合体的概念

```
#include <stdio.h>

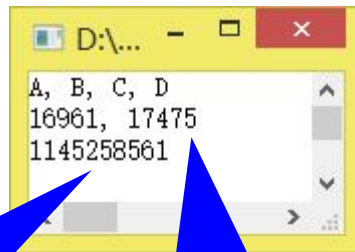
union un
{
    int i;
    short int si[2];
    char c[4];
};
```

| | | | | |
|-------|------------|-------|-----|-----|
| i | 1145258561 | | | |
| si[2] | 16961 | 17475 | | |
| c[4] | 65 | 66 | 67 | 68 |
| | [0] | [1] | [2] | [3] |

```
int main()
{
    union un x;
    x.c[0]='A';
    x.c[1]='B';
    x.c[2]='C';
    x.c[3]='D';

    printf("%c, %c, %c, %c\n", x.c[0], x.c[1], x.c[2], x.c[3]);
    printf("%d, %d\n", x.si[0], x.si[1]);
    printf("%d\n", x.i);
    return 0;
}
```

- 几个不同的变量共占同一段内存的结构，也称为共用体
- 联合体提供了从不同角度看待，以及用不同方式使用同一段内存空间的手段
- 共用体中的变量在内存中占的字节数不同，但都从同一地址开始使用，几个变量互相覆盖。



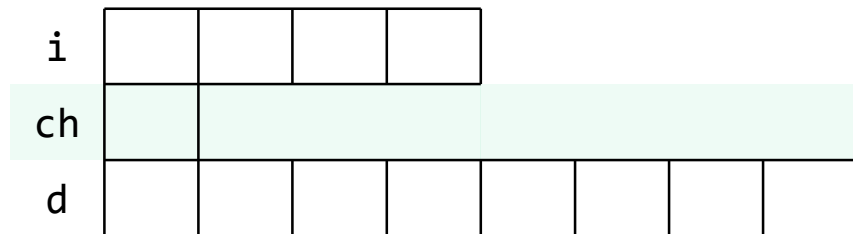
1145258561
 $= 17475 * 256 * 256 + 16961$

$17475 = 68 * 256 + 67$
 $16961 = \dots$

使用共用体

```
union data
```

```
{  
    int i;  
    char ch;  
    double d;  
} a,b,c ;
```



☐ 联合体的使用形式类似结构体

```
☐ scanf("%f", &a.d);  
☐ int x = a.i;  
☐ printf("%c\n", a.c);
```

☐ 区别：

```
☐ 结构体各成员使用的空间连贯，长度为各成员长度和；  
☐ 联合体共享存储空间，长度为最长的成员
```