

第7章 结构体和共用体





本章学习内容

- ✓ 结构体类型、变量定义，成员访问
- ✓ 结构体与函数
- ✓ 共用体和枚举类型

用户自定义的数据类型

■ 结构体 (Struct)

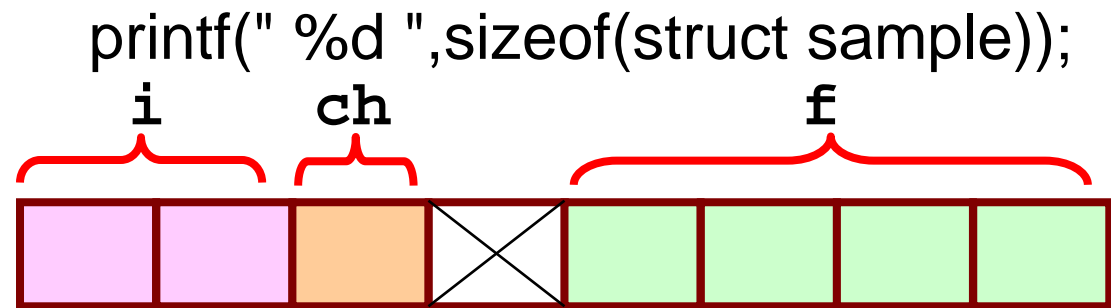
- 把关系紧密且逻辑相关的多种不同类型的变量，组织到统一的名字之下
- 占用相邻的一段内存单元

■ 共用体，也称联合 (Union)

- 把情形互斥但逻辑相关的多种不同类型的变量，组织到统一的名字之下
- 占用同一段内存单元，每一时刻只有一个数据起作用

7.6 共用体(联合体)

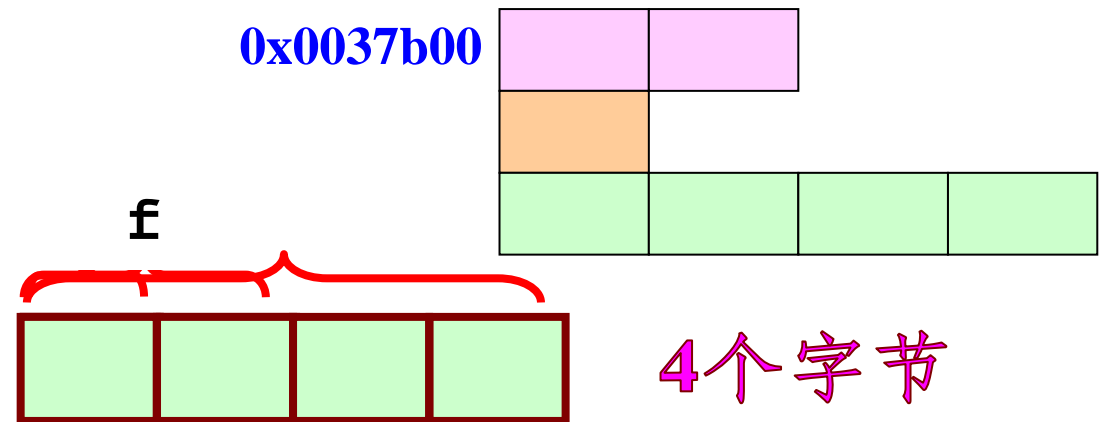
```
struct sample
{
    short  i;
    char   ch;
    float  f;
};
```



8个字节

printf(" %d ", sizeof(union sample));

```
union sample
{
    short  i;
    char   ch;
    float  f;
};
```



4个字节

7.6 共用体

```
struct member:0x28fed8 0x28feda 0x28fedc
union member:0x28fed4 0x28fed4 0x28fed4
```

```
struct sample1
{
    short    i;
    char     ch;
    float    f;
};
```

```
union sample2
{
    short    i;
    char     ch;
    float    f;
};
```

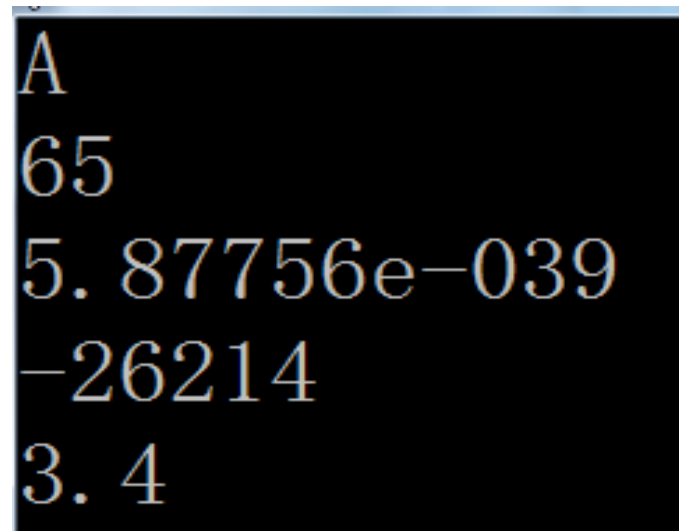
```
int main( )
{
    struct sample1 s1;
    union sample2 s2;
    printf(“%d %d”, &s1.i ,(void*)&s1.ch);
    printf(“%f\n”, &s1.f);
    printf(“%d %d”, &s2.i , (void*)&s2.ch);
    printf(“ %d”, &s2.f);
}
```

7.6 共用体

- 同一内存单元在每一瞬时只能存放其中一种类型的成员
- 起作用的成员是**最后一次存放**的成员
- 不能进行比较操作，只能对第一个成员初始化

```
int main( )
{
    union sample2 s2;
    s2.i = 65;
    printf("%c\n", s2.ch);
    printf("%d\n", s2.i);
    printf("%f\n", s2.f);
    s2.f = 3.4;
    printf("%d\n", s2.i);
    printf("%f\n", s2.f);
}
```

2019-11-30



```
A
65
5.87756e-039
-26214
3.4
```


7.7.1 枚举数据类型

■ 枚举 (*Enumeration*) 数据类型

- 描述的是一组整型值的集合
- 用于当某些量仅由有限个数据值组成时
- 默认时，第一个量为0

```
enum weeks {SUN , MON, TUE, WED, THU, FRI, SAT};
```

```
enum weeks today;
```

```
enum response {no, yes, none};
```

其值为2

```
enum response answer;
```

```
today = TUE;
```

```
answer = yes;
```

其值为1

```
enum response {no = -1, yes = 1, none = 0};
```

7.7.1 枚举数据类型

```
#include <stdio.h>

enum weeks {SUN , MON, TUE, WED, THU, FRI, SAT};
enum response {no = -1,  yes = 1,  none = 0};

int main( )
{
    weeks w;
    w = THU; //只能采用枚举类型定义时给定的枚举值赋值.
    printf("%d", w); //4
    //w = 4; //错误
    response r;
    r = no;
    printf("%d", r); //-1
}
```


提问与解答

