



C语言高级编程-共用体及typedef

主讲：小美老师

创客引领未来

扫微信二维码 获取更多信息





lemon

昵称：小姜老师
华清创客学院，嵌入式讲师

makerU

- 共用体的概念

在C语言中，不同数据类型的数据可以使用共同的存储区域，这种数据构造类型称为共用体，简称共用，又称联合体。共用体在定义、说明和使用形式上与结构体相似。两者本质上的不同仅在于使用内存的方式上。

定义一个共用体类型的一般形式为：

```
union 共用体名  
{
```

```
    成员表列;  
};
```

共用体

例如：

```
union gy
{
    int i;
    char c;
    float f;
};
```

这里定义了一个共用体类型union gy，它由三个成员组成，这三个成员在内存中使用共同的存储空间。由于共用体中各成员的数据长度往往不同，所以共用体变量在存储时总是按其成员中数据长度最大的成员占用内存空间。

在这一点上共用体与结构体不同，结构体类型变量在存储时总是按各成员的数据长度之和占用内存空间。

例如，定义了一个结构体类型：

```
struct gy
{
    int i;
    char c;
    float f;
};
```

则结构体类型struct gy的变量占用的内存大小为 $2+1+4=7$ 个字节（不考虑字节对齐）。

在使用共用体类型变量的数据时要注意：在共用体类型变量中起作用的成员是最后一次存放的成员，在存入一个新的成员后原有的成员就失去作用。如有以下赋值语句：

```
a.i = 1;
```

```
a.c = 'a';
```

```
a.f = 1.5;
```

完成以上三个赋值运算以后，a.f是有效的，a.i和a.c已经无意义了。

共用体

在程序中经常使用结构体与共用体相互嵌套的形式。

即共用体类型的成员可以是结构体类型，或者结构体类型的成员是共用体类型。



共用体

例如，下列结构体类型datas的第三个成员是共用体类型：

```
struct datas
{
    char *ps;
    int type;
    union
    {
        float fdata;
        int idata;
        char cdata;
    } udata;
};
```


typedef

- 在C语言中，允许使用关键字typedef定义新的数据类型其语法如下：

```
typedef    <已有数据类型>    <新数据类型>;
```

如：

```
typedef    int    INTEGER;
```

这里新定义了数据类型INTEGER，其等价于int

```
INTEGER    i;    <==>    int    i;
```

typedef

- 在C语言中经常在定义结构体类型时使用typedef，例如

```
typedef struct _node_  
{  
    int data;  
    struct _node_ *next;  
} listnode, *linklist;
```

这里定义了两个新的数据类型listnode和linklist。其中listnode等价于数据类型struct _node_ 而 linklist等价于struct _node_ *



扫微信二维码 获取更多信息