

嵌入式C语言之- 结构体指针与动态内存

讲师：叶大鹏

助力你成为优秀的电子工程师！



结构体成员包含指针类型

- typedef struct

{

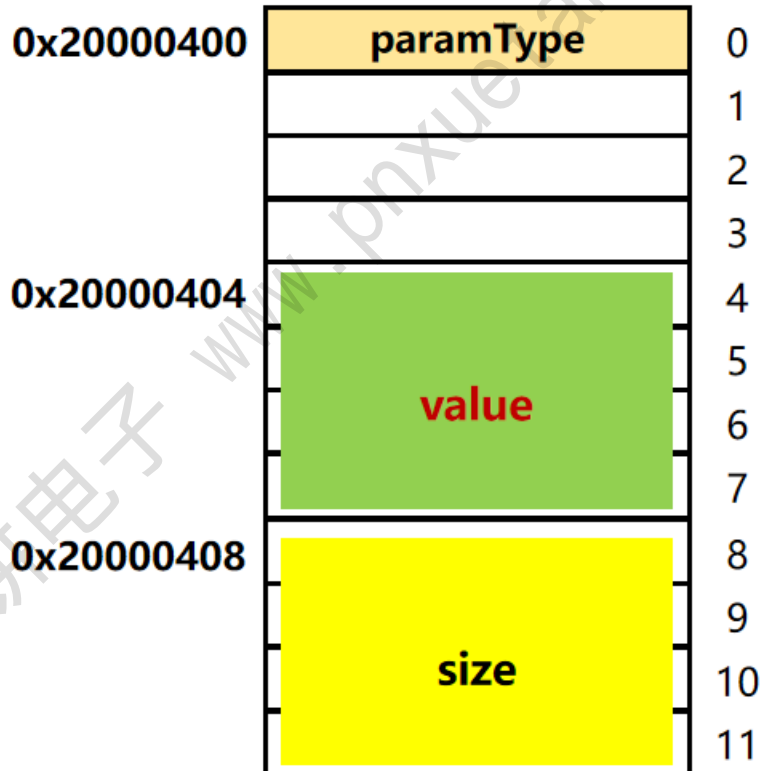
uint8_t paramType;

uint8_t *value;

uint32_t size;

} CfgParam;

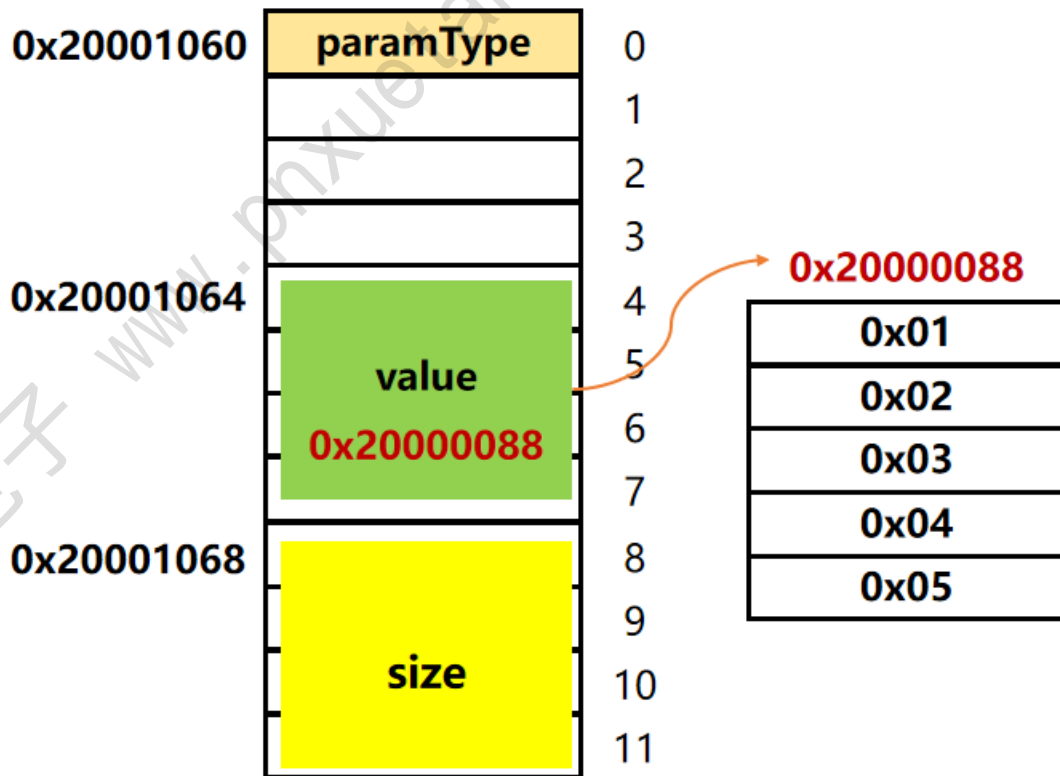
CfgParam cfgParam;



结构体成员包含指针类型

- typedef struct

```
{  
    uint8_t paramType;  
    uint8_t *value;  
    uint32_t size;  
} CfgParam;  
  
CfgParam cfgParam;  
cfgParam.value = (uint8_t *)malloc(5);  
cfgParam.value[0] = 0x01;  
cfgParam.value[1] = 0x02;  
cfgParam.value[2] = 0x03;  
cfgParam.value[3] = 0x04;  
cfgParam.value[4] = 0x05;
```



结构体指针变量与动态内存

```
CfgParam *cfgParam;
```

```
cfgParam = (CfgParam *)malloc(sizeof(CfgParam));
```

cfgParam 0x20000088
0x20001068~106B

```
cfgParam->value = (uint8_t *)malloc(5);
```

```
cfgParam->value[0] = 0x01;
```

```
cfgParam->value[1] = 0x02;
```

```
cfgParam->value[2] = 0x03;
```

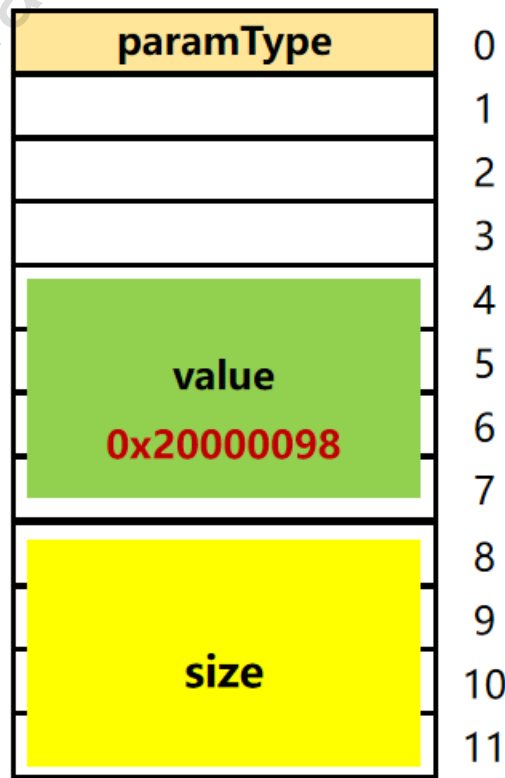
```
cfgParam->value[3] = 0x04;
```

```
cfgParam->value[4] = 0x05;
```

0x20000088

0x2000008C

0x20000090



结构体指针变量与动态内存

```
CfgParam *cfgParam;  
cfgParam = (CfgParam *)malloc(sizeof(CfgParam));
```

cfgParam 0x20000088
0x20001068~106B

```
cfgParam->value = (uint8_t *)malloc(4);
```

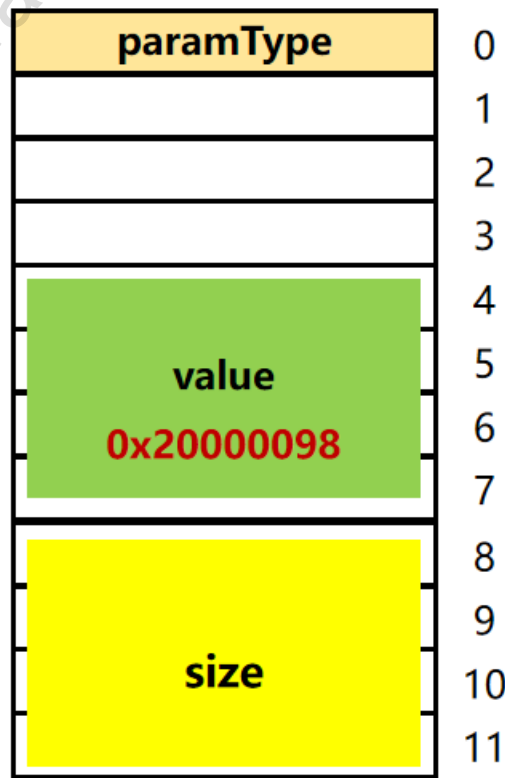
```
free(cfgParam->value);  
cfgParam->value = NULL;  
free(cfgParam);  
cfgParam = NULL;
```

注意释放的顺序，先释放
成员的，再释放结构体的

0x20000088

0x2000008C

0x20000090



位运算优先级

运算符（优先级从上往下）	运算符说明及应用场景	结合性
() [] -> .	括号（函数等），数组，结构体指针变量的成员访问，普通结构体变量的成员访问	由左向右
! ~ ++ -- + -	逻辑非，按位取反，自增1，自减1，正号，负号	由右向左
* & (类型) sizeof	指针，取地址，强制类型转换，求占用空间大小	
* / %	乘，除，取模	由左向右
+ -	加，减	由左向右
<< >>	左移，右移	由左向右

THANK YOU!