

嵌入式C语言之- 宏定义中为什么要使用do-while(0)

讲师：叶大鹏

助力你成为优秀的电子工程师！



do-while(0)的特殊用法

```
#define DISK_LOCK(mux) do { \
    if (pthread_mutex_lock(mux) != 0) { \
        PRINT_ERR("%s %d, mutex lock failed\n", __FUNCTION__, __LINE__); \
    } \
} while (0)
```

```
#define SWITCH_QUOTES_STATUS(qu) do { \
    if ((qu) == TRUE) { \
        (qu) = FALSE; \
    } else { \
        (qu) = TRUE; \
    } \
} while (0)
```

#define的注意事项

- 由于#define只对当行生效，当宏定义的值在一行内写不下时，需要换行，这时候需要使用续行符'\ '实现，除最后一行外，每行行尾都必须加续行符。这样#define的作用范围就可以延续到下一行，不然编译会报错，例如：

```
#define USA "The United \
States of \
America"
```

do-while(0)的特殊用法

```
#define EXCH(x, y) do { \
    int32_t tmp; \
    tmp = x; \
    x = y; \
    y = tmp; \
} while (0)
```

typedef和#define的区别

- #define 与typedef功能都是给一个对象取一个别名，增强程序的可读性，但它们在使用时有以下几点区别：

1. 应用场景不同：

- 1) typedef用来给一种数据类型定义别名；
- 2) #define用来给数字、表达式、代码语句定义别名；

2. 执行时机不同：

- 1) typedef在编译阶段执行；
- 2) #define在预编译阶段执行；

3. 定义方法不同：

- 1) #define别名在替换对象的前面，并且定义后面不用加分号；

```
#define PI 3.14159
```

- 2) typedef的别名在替换对象的后面，并且定义后面需要加分号；

```
typedef signed char int8_t;
```

typedef的用法

```
typedef struct {  
    unsigned int  consoleID;  
    ...  
    char  shellBuf[SHOW_MAX_LEN];  
    char  shellWorkingDirectory[PATH_MAX];  
} ShellCB;
```

```
typedef enum {  
    CMD_TYPE_SHOW = 0,  
    CMD_TYPE_STD = 1,  
    CMD_TYPE_EX = 2,  
    CMD_TYPE_BUTT  
} CmdType;
```

typedef和#define的区别

- typedef不同于#define，不是简单的字符串替换：

```
typedef int * PINT1;
```

```
#define PINT2 int *
```

```
// i1、i2 都为int *，typedef为int *引入了一个新的别名
```

```
PINT1 i1, i2;
```

```
// 相当于int * i3, i4; i3是int *, 而i4是int
```

```
PINT2 i3, i4;
```

THANK YOU!