

# FreeRTOS学习笔记（七）

## FreeRTOS信号量

简述：信号量用于控制共享资源访问的场景相当于一个上锁机制，代码只有获得了这个锁的钥匙才可以执行。

### 1. 二值信号量

- 简述：相当于是一个只有一个列表项的队列，要么是满的（有信号），要么是空的（没信号）
- 二值信号量的相关API

```
/**
 * 描述：创建一个二值信号量（也就是创建一个长度为1的消息队列）
 * 返回：NULL ：创建失败
 *      其他值：创建成功的二值信号量的句柄
 */
SemaphoreHandle_t xSemaphoreCreateBinary(void);

/**
 * 描述：释放信号量
 * 参数：xSemaphore：要释放的信号量句柄
 * 返回：pdPASS ：释放成功
 *      errQUEUE_FULL：释放失败
 */
void xSemaphoreGive(SemaphoreHandle_t xSemaphore);

/**
 * 描述：获取信号量
 * 参数：xSemaphore：要获取的信号量句柄
 *      xBlockTime：阻塞时间
 * 返回：pdTRUE ：获取成功
 *      pdFALSE：获取失败
 */
void xSemaphoreTake(SemaphoreHandle_t xSemaphore,
                    TickType_t xBlockTime);
```

### 2. 计数型信号量

- 描述：相当于一个长度大于1的队列（长度用户可以自行设置）
- 计数型信号量相关的API

```
/**
 * 描述：创建一个计数型信号量
 * 参数：uxMaxCount ：计数允许的最大值
 *      uxInitialCount：起始计数值
 * 返回：Null ：创建失败
 *      其他值：创建成功的信号量句柄
 */
```

```
SemaphoreHandle_t xSemaphoreCreateCounting(UBaseType_t uxMaxCount,  
                                             UBaseType_t uxInitialCount);
```

```
/*  
*****  
/* 释放信号量和获取信号量函数与上述二值信号量的相同 */  
*****  
*/
```

### 3. 优先级反转

- 描述：例：高中低优先级的三个任务，高优先级运行完之后释放二值信号量，低优先级的任务获取信号量之后进入长时间延时一直没有释放信号量，然后中优先级任务一直在运行，等到低优先级的任务运行完释放信号量之后，高优先级任务才开始运行。

### 4. 互斥信号量

- 描述：一个拥有优先级继承的二值信号量（尽可能降低优先级翻转带来的影响）
- 互斥信号量的相关API

```
/*  
* 描述：创建一个互斥信号量  
* 返回：Null ：创建失败  
* 其他值：创建成功的信号量句柄  
*/  
SemaphoreHandle_t xSemaphoreCreateMutex(void);  
  
/*  
*****  
/* 释放信号量和获取信号量函数与上述二值信号量的相同 */  
*****  
*/
```

### 5. 递归互斥信号量

- 描述：可以获取多次互斥信号量，不过要释放相应次数。