当计数值大于0,代表有信号量资源 当释放信号量,信号量计数值(资源数)加一 1,信号量的简介(了解) 当获取信号量,信号量计数值(资源数)减一 特点: 信号量: 用于传递状态 当信号量如果最大值被限定为1,那么它就是二值信号量;如果最大值不是1,它就是 计数型信号量。 简介:二值信号量的本质是一个队列长度为1的队列,该队列就只有空和满两种情况,这就是二值。 创建二值信号量函数: SemaphoreHandle t xSemaphoreCreateBinary(void) 2, 二值信号量 (熟悉) 相关API函数介绍 释放二值信号量函数: BaseType t xSemaphoreGive(xSemaphore) 获取二值信号量函数: BaseType t xSemaphoreTake(xSemaphore, xBlockTime) 简介: 计数型信号量相当于队列长度大于1的队列, 因此计数型信号量能够容纳多个资 源,这在计数型信号量被创建的时候确定的 事件计数: 当每次事件发生后, 在事件处理函数中释放计数型信号量(计数值+1), 其他任务会获取计数型信号量(计数值-1),这种场合一般在创建时将初始计数值设 置为0 计数型信号量适用场合: 资源管理:信号量表示有效的资源数目。任务必须先获取信号量(信号量计数值-1) 3, 计数型信号量 (熟悉) 才能获取资源控制权。当计数值减为零时表示没有的资源。当任务使用完资源后,必须 FreeRTOS信号量 释放信号量(信号量计数值+1)。信号量创建时计数值应等于最大资源数目 创建计数型信号量: xSemaphoreCreateCounting( uxMaxCount , uxInitialCount ) 相关API函数介绍 获取信号量当前计数值大小: uxSemaphoreGetCount(xSemaphore) 计数型信号量的释放和获取与二值信号量相同! 简介: 优先级翻转: 高优先级的任务反而慢执行, 低优先级的任务反而优先执行 优先级翻转在抢占式内核中是非常常见的,但是在实时操作系统中是不允许出现优先级 翻转的,因为优先级翻转会破坏任务的预期顺序,可能会导致未知的严重后果。 4, 优先级翻转简介(熟悉) 总结: 高优先级任务被低优先级任务阻塞, 导致高优先级任务迟迟得不到调度。但其他 中等优先级的任务却能抢到CPU资源。从现象上看,就像是中优先级的任务比高优先级 任务具有更高的优先权 (即优先级翻转) 简介:互斥信号量其实就是一个拥有优先级继承的二值信号量,在同步的应用中二值信 号量最适合。 互斥信号量适合用于那些需要互斥访问的应用中! 优先级继承: 当一个互斥信号量正在被一个低优先级的任务持有时, 如果此时有个高 优先级的任务也尝试获取这个互斥信号量,那么这个高优先级的任务就会被阻塞。不过 这个高优先级的任务会将低优先级任务的优先级提升到与自己相同的优先级。 5, 互斥信号量(熟悉) (1) 互斥信号量有任务优先级继承的机制, 但是中断不是任务, 没有任务优先级, 所以 互斥信号量只能用与任务中,不能用于中断服务函数。 注意: 互斥信号量不能用于中断服务函数中, 原因如下: (2) 中断服务函数中不能因为要等待互斥信号量而设置阻塞时间进入阻塞态。 注意: 创建互斥信号量时, 会主动释放一次信号量

简介:信号量是一种解决同步问题的机制,可以实现对共享资源的有序访问