第20章 互斥量

主讲内容

20.1 互斥量的基本概念

20.2 互斥量的优先级继承机制

20.3 互斥量应用场景

20.4 互斥量运作机制

20.5 互斥量控制块

20.6 互斥量函数接口讲解

20.7. 模拟优先级翻转实验

20.8 互斥量实验

20.9 总结

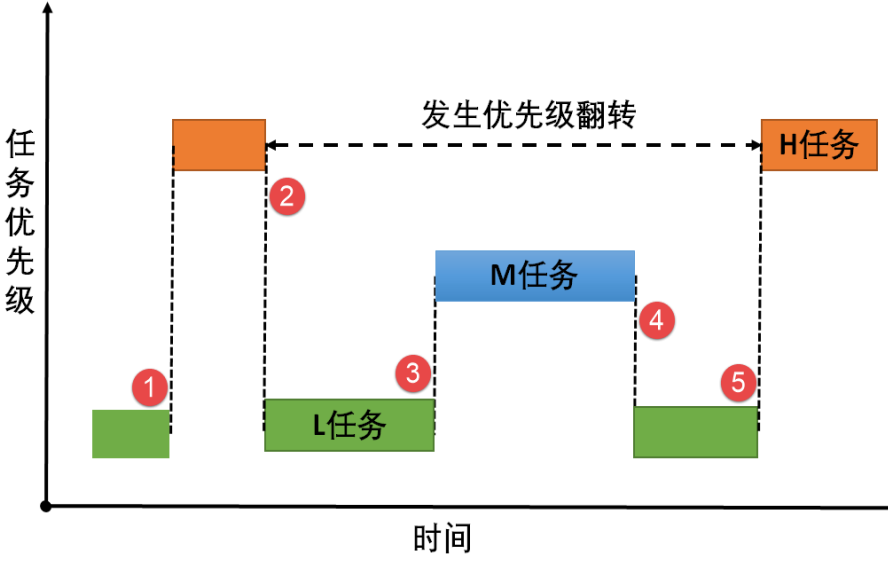
参考资料:《μCOS-III内核实现与应用开发实战指南》

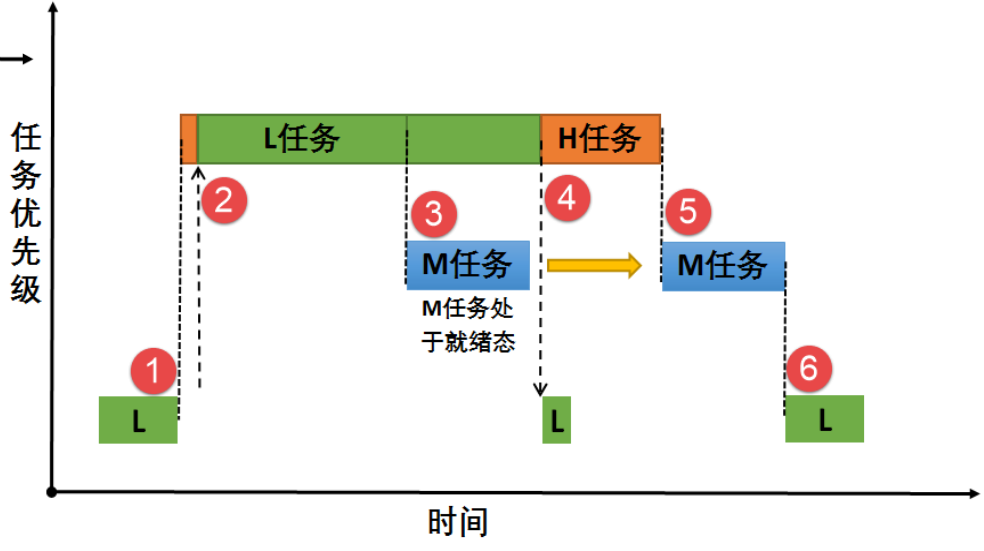
互斥量的基本概念

互斥量又称互斥信号量（本质也是一种信号量，不具备传递数据功能），是一种特殊的二值信号量，它和信号量不同的是，它支持互斥量所有权、递归访问以及防止优先级翻转的特性，用于实现对临界资源的独占式处理。任意时刻互斥量的状态只有两种，开锁或闭锁。

互斥量的优先级继承机制

在μC/OS操作系统中为了降低优先级翻转问题利用了优先级继承算法。优先级继承算法是指，暂时提高某个占有某种资源的低优先级任务的优先级，使之与在所有等待该资源的任务中优先级最高那个任务的优先级相等，而当这个低优先级任务执行完毕释放该资源时，优先级重新回到初始设定值。


任务优先级
发生优先级翻转
H任务
M任务
L任务
1
2
3
4
5
时间


任务优先级
L任务
H任务
M任务
M任务
M任务处
于就绪态
L
L
L
1
2
3
4
5
6
时间

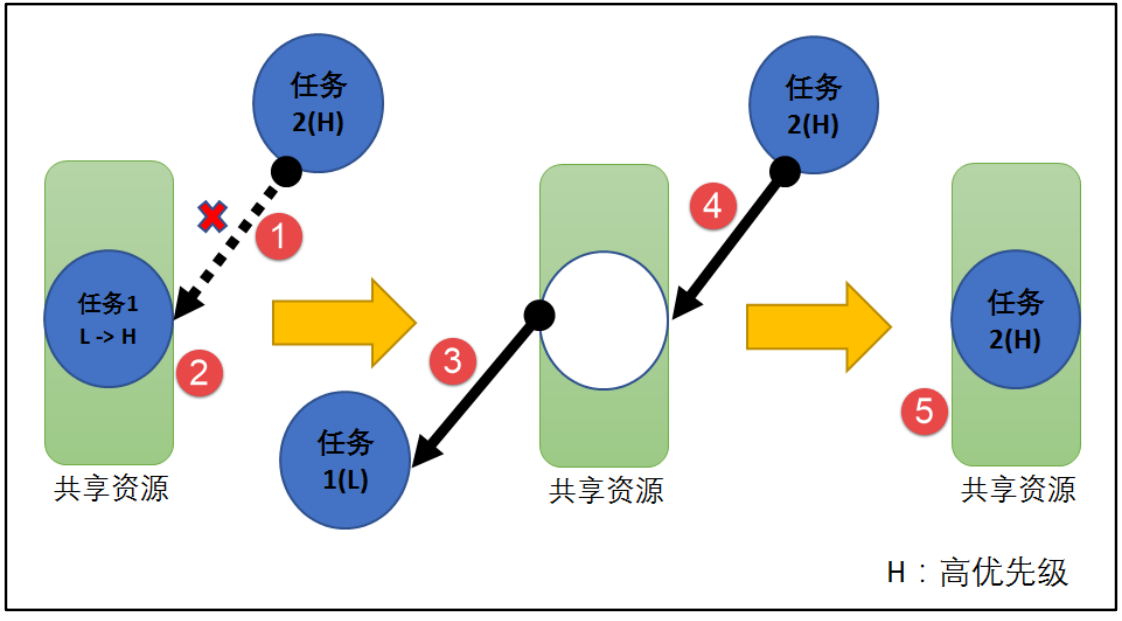
互斥量应用场景

互斥量的使用比较单一，因为它是信号量的一种，并且它是以锁的形式存在。在初始化的时候，互斥量处于开锁的状态，而被任务持有的时候则立刻转为闭锁的状态。互斥量更适合于：

可能会引起优先级翻转的情况。

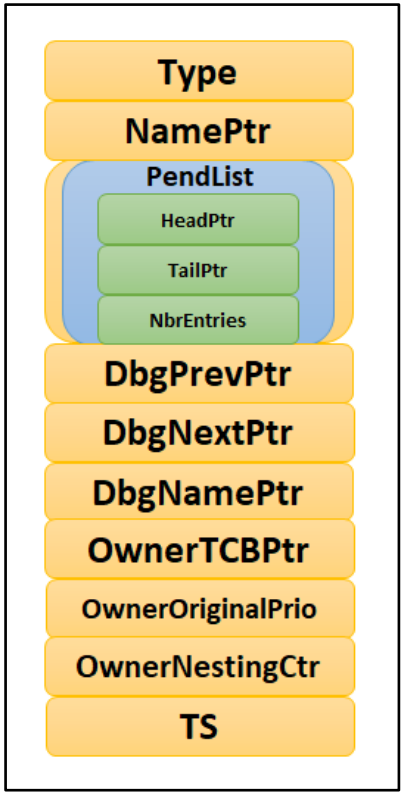
任务可能会多次获取互斥量的情况下，这样可以避免同一任务多次递归持有而造成死锁的问题。

互斥量运作机制

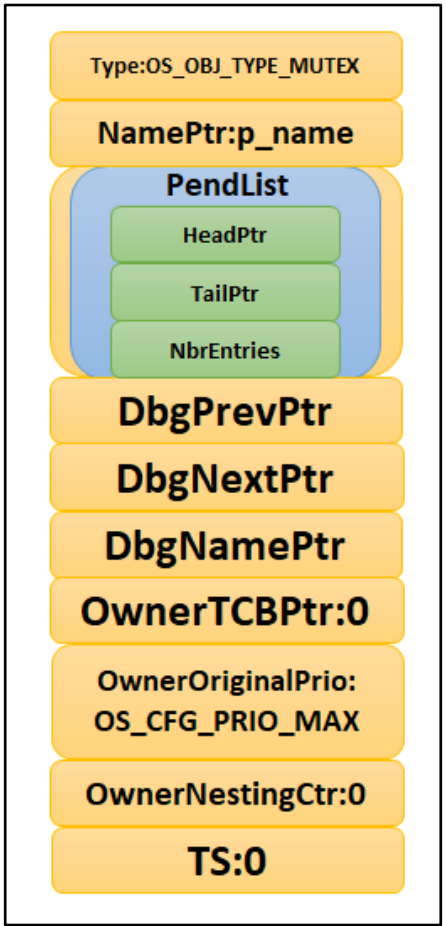


互斥量控制块

μC/OS的互斥量由多个元素组成，在互斥量被创建时，需要由我们自己定义互斥量（也可以称之为互斥量句柄）



创建互斥量函数OSMutexCreate()



删除互斥量函数OSMutexDel()

OSSemDel()用于删除一个互斥量，互斥量删除函数是根据互斥量结构（互斥量句柄）直接删除的，删除之后这个互斥量的所有信息都会被系统清空，而且不能再次使用这个互斥量了

获取互斥量函数OSMutexPend()

任务对互斥量的所有权是独占的，任意时刻互斥量只能被一个任务持有，如果互斥量处于开锁状态，那么获取该互斥量的任务将成功获得该互斥量，并拥有互斥量的使用权；如果互斥量处于闭锁状态，获取该互斥量的任务将无法获得互斥量，任务将被挂起，在任务被挂起之前，会进行优先级继承。

释放互斥量函数OSMutexPost()

只有已持有互斥量所有权的任务才能释放它，当任务调用OSMutexPost()函数时会释放一次互斥量，当互斥量的成员变量OwnerNestingCtr为0的时候，互斥量状态才会成为开锁状态，等待获取该互斥量的任务将被唤醒。如果任务的优先级被互斥量的优先级翻转机制临时提升，那么当互斥量被完全释放后，任务的优先级将恢复为原本设定的优先级

模拟优先级翻转实验

模拟优先级翻转实验是在μC/OS中创建了三个任务与一个二值信号量，任务分别是高优先级任务AppTaskLed3，中优先级任务AppTaskLed2，低优先级任务AppTaskLed1，用于模拟产生优先级翻转。低优先级任务在获取信号量的时候，被中优先级打断，中优先级的任务开始执行，因为低优先级还未释放信号量，那么高优先级任务就无法取得信号量继续运行，此时就发生了优先级翻转，任务在运行中，使用串口打印出相关信息

互斥量实验

互斥量实验是基于优先级翻转实验进行修改的，将信号量改为互斥量，目的是为了测试互斥量的优先级继承机制是否有效

总结

互斥量更适用于保护各个任务间对共享资源的互斥访问，当然系统中对于这种互斥访问的资源可以使用很多种保护的方式，如关闭中断方式、关调度器方式、信号量保护或者采用互斥量保护，但是这些方式各有好坏，下面就简单说明一下这4种方式的使用情况

共享资源保护方式	说明
关闭中断方式	什么时候该用：当系统能很快地结束访问该共享资源时，如一些共享的全局变量的操作，可以关闭中断，操作完成再打开中断即可。但是我们一般不推荐使用这种方法，因为会导致中断延迟。
锁调度器方式	当访问共享资源较久的时候，比如对一些列表的操作，如遍历列表、插入、删除等操作，对于操作时间是不确定的，如一些 os 中的内存分配，都可以采用锁定调度器这种方式进行共享资源的保护。
信号量保护方式	当该共享资源经常被多个被使用时可以使用这种方式。但信号量可能会导致优先级翻转，并且信号量是无法解决这种危害的。
互斥量保护方式	推荐使用这种方法访问共享资源，尤其当任务要访问的共享资源有阻塞时间的时候。μ C/OS-III 的互斥量有内置的优先级，这样可防止优先级翻转。然而，互斥量方式慢于信号量方式，因为互斥量需执行额外的操作，改变任务的优先级。

THANKS

淘宝：fire-stm32.taobao.com

论坛： www.firebbs.cn

扫描进入淘宝店铺