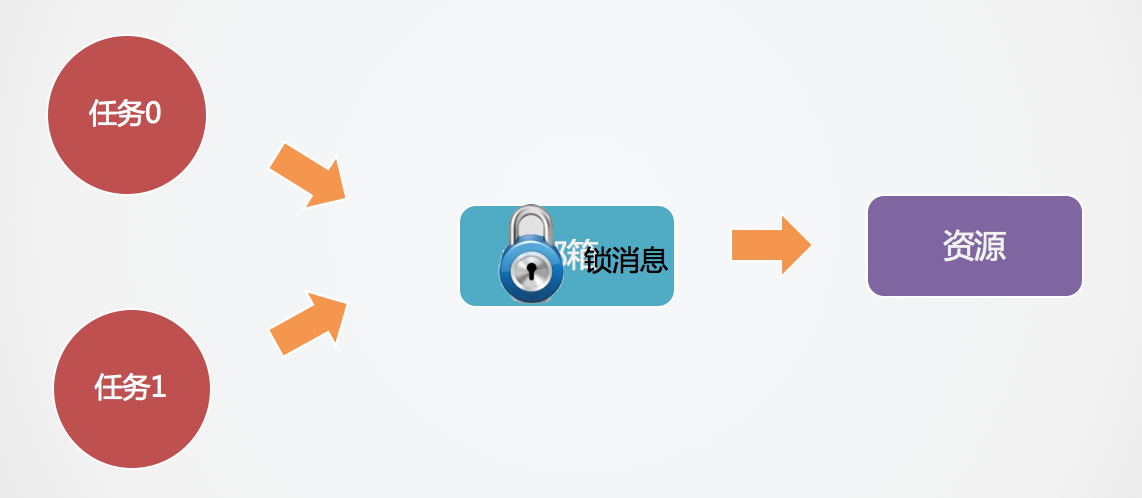
# 内容回顾

在本节课程中，我们学习的怎么样用邮箱去解决任务之间共享自愿的问题。它在具体工作原理主要是这样的。

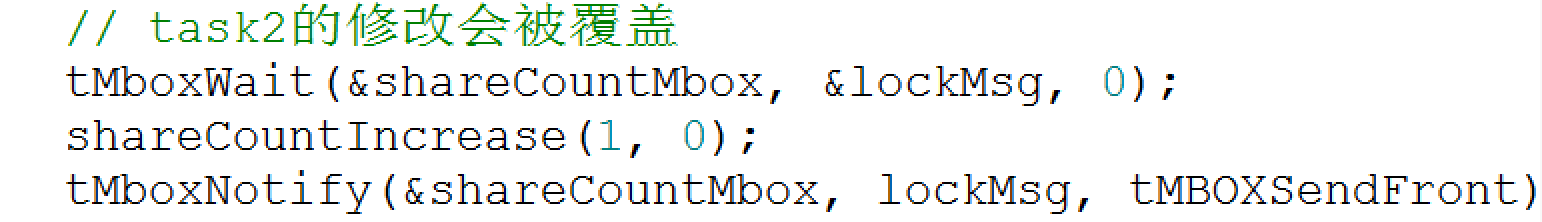


我们将**邮箱当作了一个装锁的容器**。在初始化的时候我们中邮箱中存入一个消息，**这个消息的就相当于一把锁**。

当任务需要去访问资源的时候，首先从邮箱中获取这把锁，然后再去访问资源。

在这个过程中，如果有其他任务尝试去访问资源，由于邮箱中已经没有这把锁，那么这个任务就需要等待。

通过这种方式，就实现了资源之间共享的一个互斥。具体的代码演示如下：



# 其它OS示例

待补充

# 思考练习

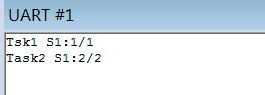
# 视频勘误

暂无

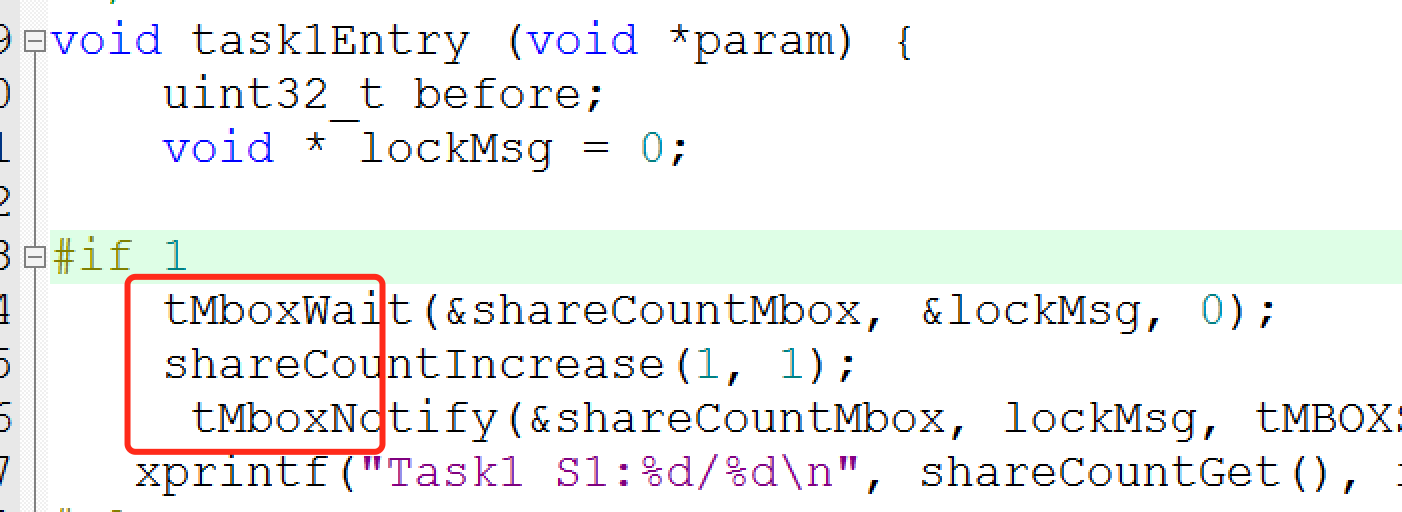
# 重点答疑

1. 请解释下视频的输出结果

视频中可以看到，程序的输出结果如下。



首先是任务1开始运行，首先从邮箱中获取锁，访问资源，最后再释放。



在释放的时候，虽然这个时候任务2已经在邮箱中等待消息，但是由于任务2的优先级比较低所以他没有办法立即运行。任务1继续执行，输出结果：1/1。

之后任务2可以运行时候，它再对计数器进行自增。将计数器的值增加到2，然后打印结果：2/2

1. 在对邮箱进行初始化的时候，为什么要对邮箱传入一个初始化的数组？

因为我们的邮箱中只包含了对消息队列进行控制的相关操作变量，但是没有包含这个消息队列的具体定义。因为这个消息队列具体有多长是不确定的， 所以我们需要外部传入，方便我们进行自定义。