Project4 Synchronization Primitives and IPC设计文档

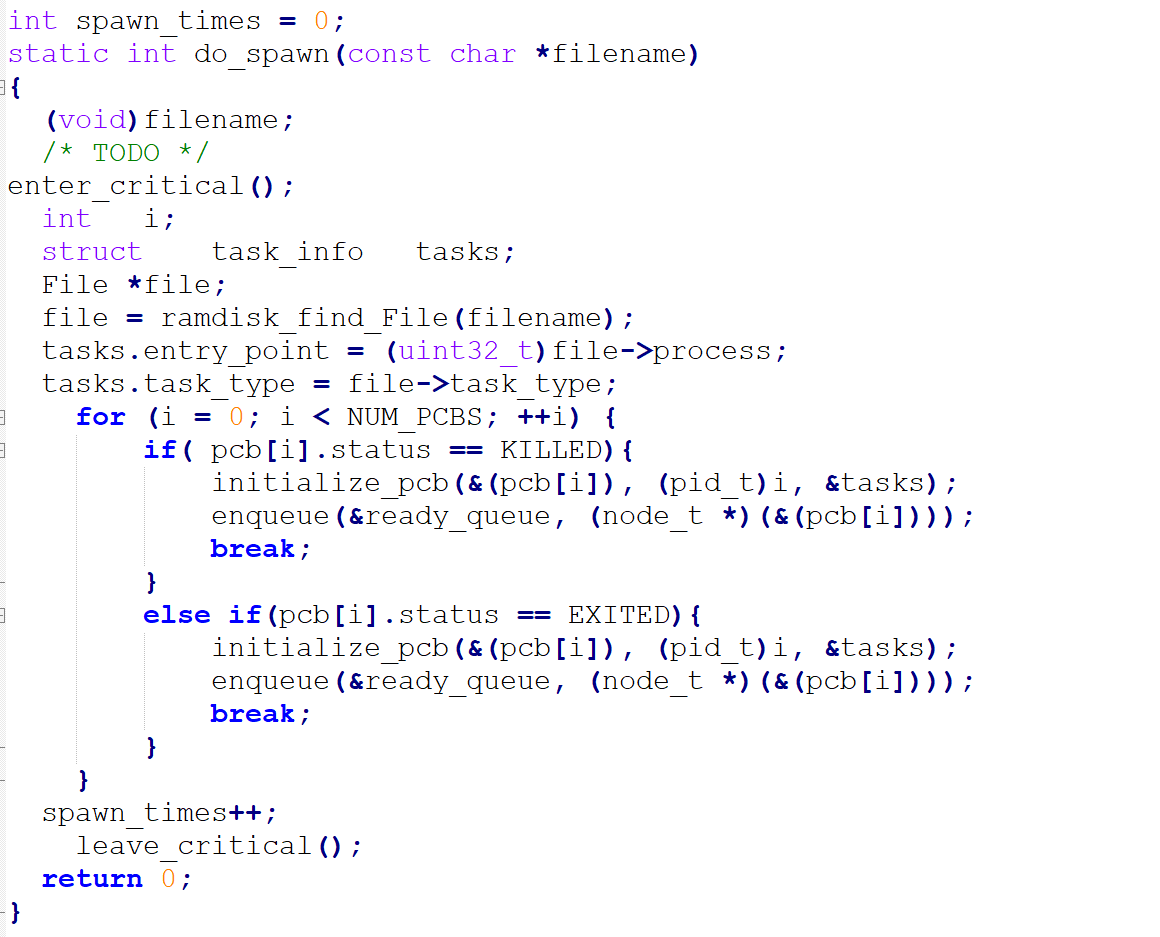
中国科学院大学

张旭

2017/11/29

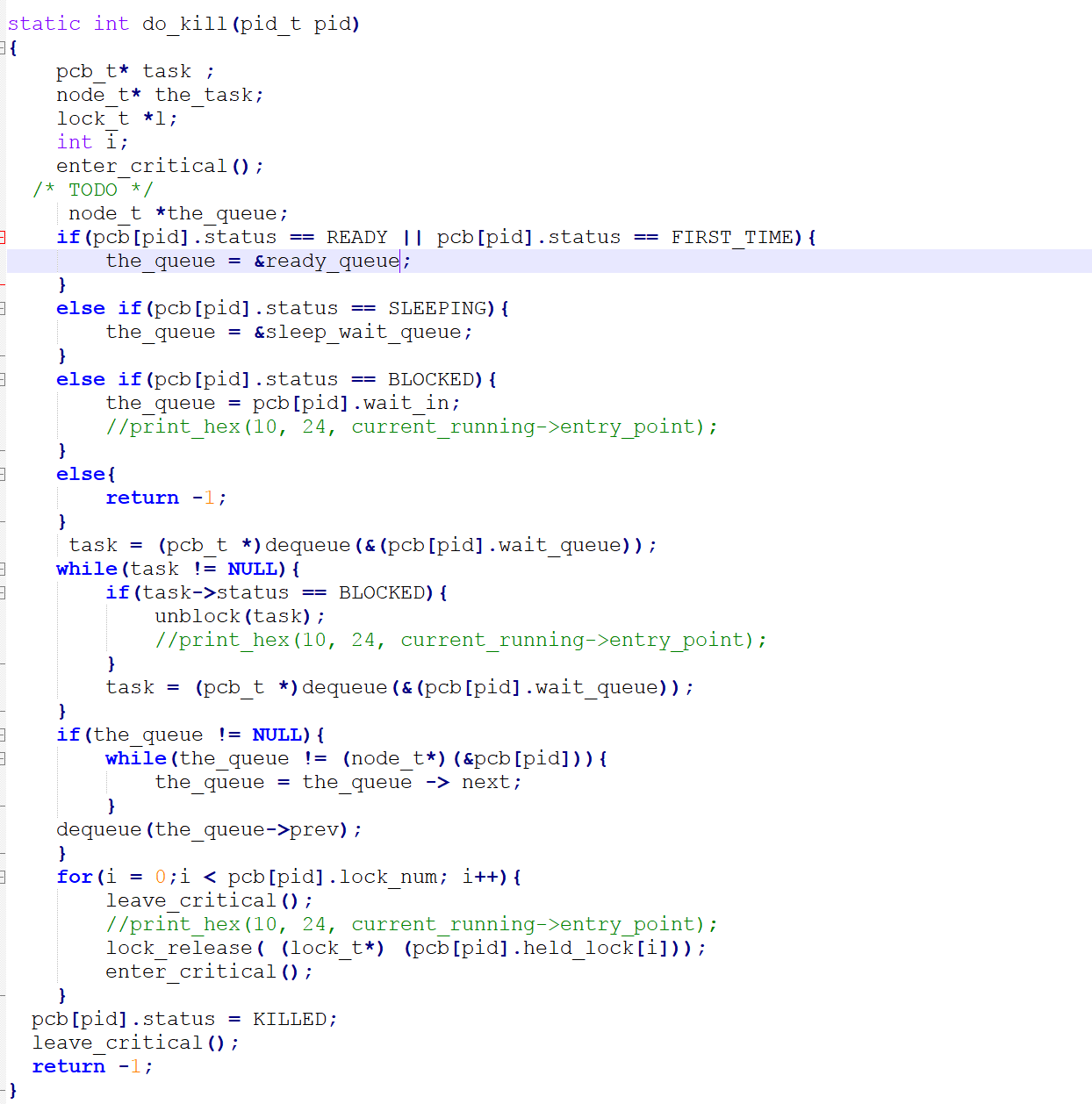
# do\_spawn, do\_kill和do\_wait设计

1. do\_spawn的处理过程，如何生成进程ID

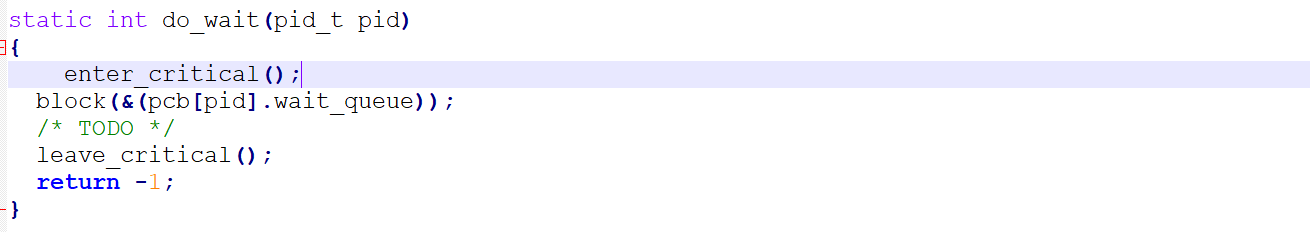


将进程的信息填入task结构中 🡪 查找PCB数组，回收状态为EXITED和KILLED的PCB 🡪 初始化PCB并加入准备队列。

1. do\_kill的处理过程。如果有做bonus，请在此说明在kill task时如何处理锁



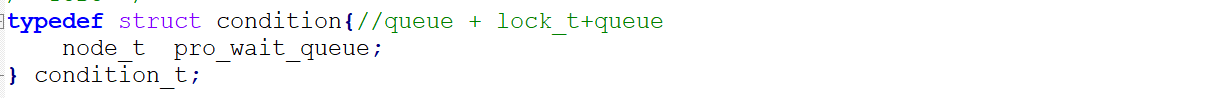
检查被杀进程的状态，并根据其状态记录其所在的等待队列 🡪 唤醒等该该进程的进程 🡪 将该进程从其所在的等待队列中删除 🡪 释放其保留的锁 🡪 修改PCB的状态为KILLED

1. do\_wait的处理过程  
   

将该进程加入pcb[pid]的等待队列中

# 同步原语设计

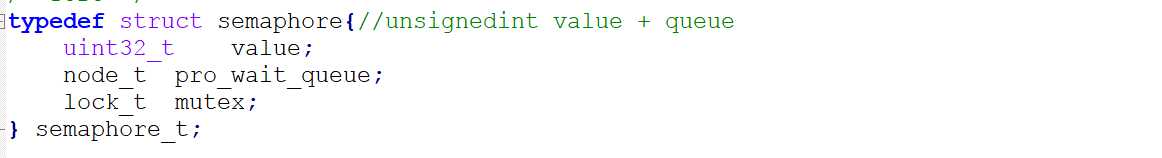
1. 条件变量的含义，及其所实现的各自数据结构的包含内容

条件变量允许线程由于一些未达到的条件而阻塞。

条件变量包含一个等待队列。

1. 信号量的含义，及其所实现的数据结构的包含内容

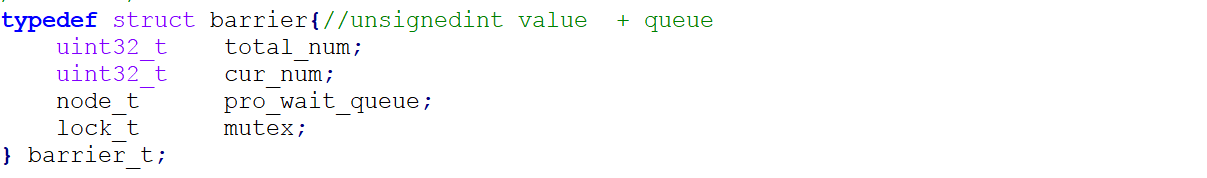
信号量取指可以为0（表示没有保存下来的唤醒操作）或者为正值（表示有一个或多个唤醒操作），并有UP和DOWN两种操作。



信号量结构包含一个锁变量，一个等待队列，一个信号量值。

1. 屏障的含义，及其所实现的各自数据结构的包含内容

屏障实现了，除非所有的进程都就绪准备着手下一阶段，否则任何进程都不能进入下一个阶段。



屏障结构包含一个等待进程数阈值，一个记录目前等待进程数的变量，一个等待队列，一个锁变量。

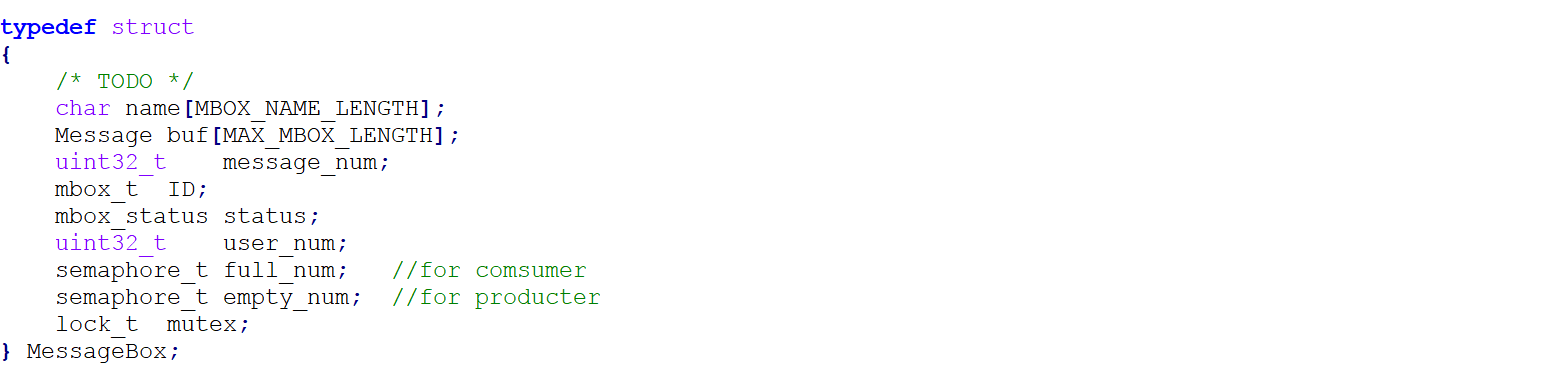
# mailbox设计

1. mailbox的数据结构以及主要成员变量的含义

MESSAGE的结构体：



MessageBox结构体：

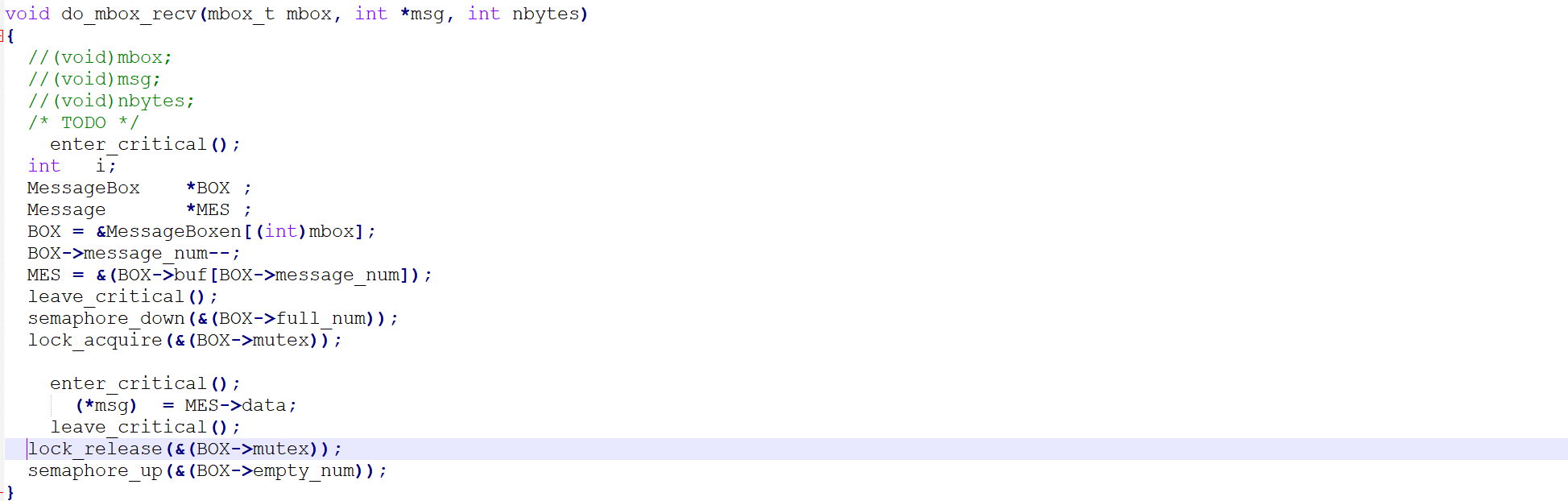


Name数组保存信箱的名称，buf数组保存最多MBOX\_NAME\_LENGTH个信息，message\_num记录目前的信息数，status记录信箱的状态，user\_num记录目前使用信箱的用户数，两个信号量和一个锁变量辅助完成信箱操作。

1. producer-consumer问题是指什么？你在mailbox设计中如何处理该问题？

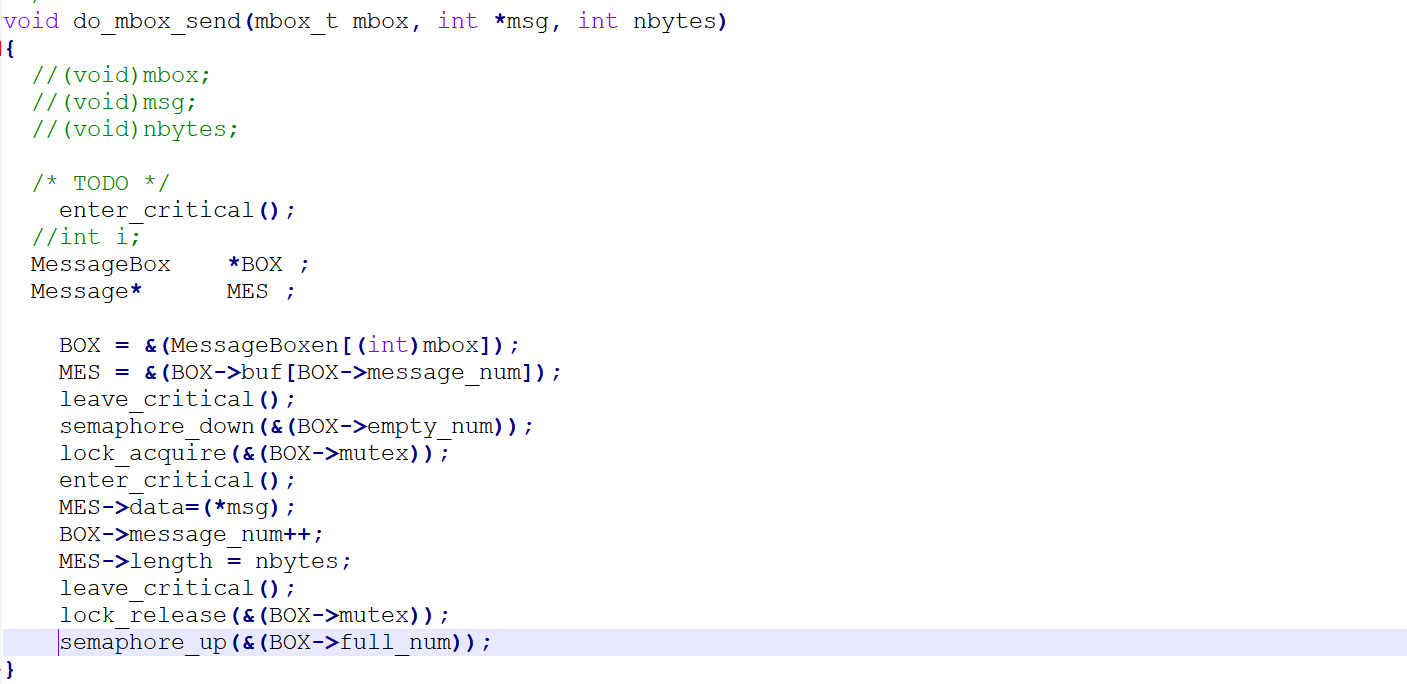
producer-consumer问题指两个进程共享一个公共的固定大小的缓冲区。其中一个是生产者，将信息放入缓冲区；另一个是消费者，从缓冲区中取出信息，问题在于当缓冲区已满，而此时生产者还想向其中放入一个新的数据项。

我用信号量和锁变量处理生产者-消费者问题。如下面的do\_mbox\_recv函数，对full\_num信号量（初始化为0）做一次down操作 🡪 申请锁 🡪 对信箱中的信息操作 🡪 释放锁 🡪 对empty\_num信号量（初始化为MBOX\_NAME\_LENGTH）做一次up操作



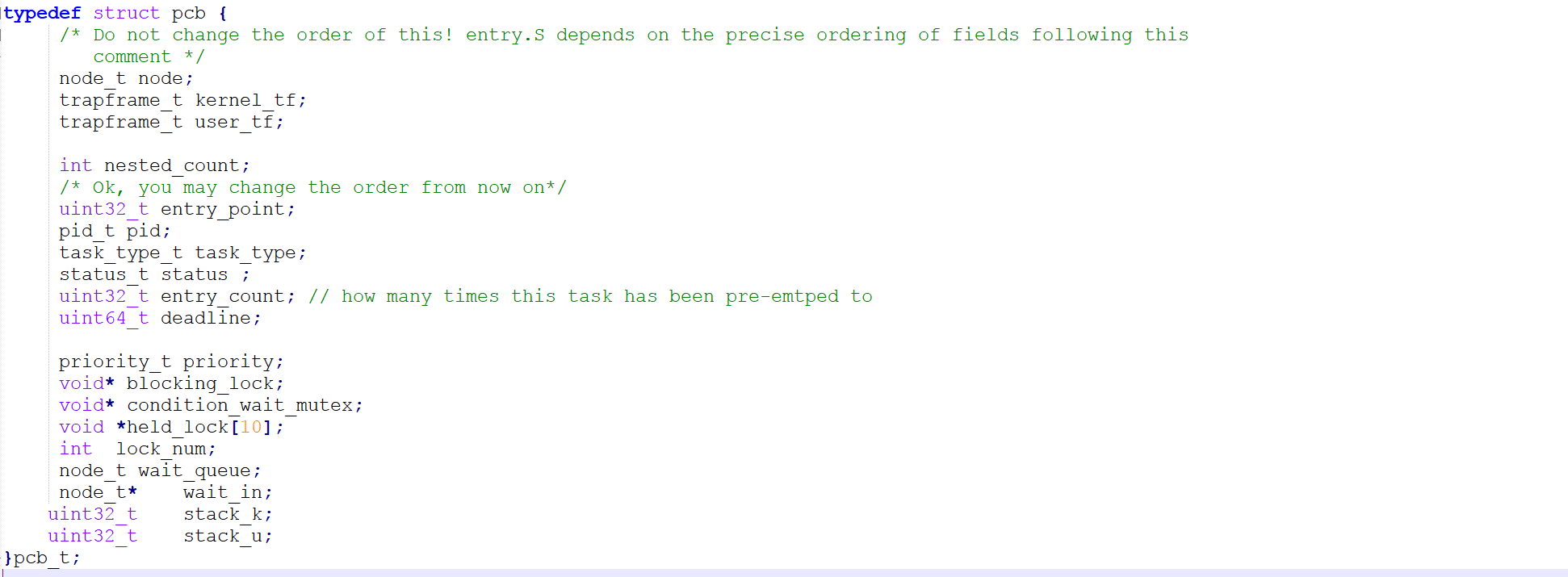
do\_mbox\_send函数：

对empty\_num信号量（初始化为MBOX\_NAME\_LENGTH）做一次down操作 🡪 申请锁 🡪 对信箱中的信息操作 🡪 释放锁 🡪 对full\_num信号量（初始化为0）做一次up操作



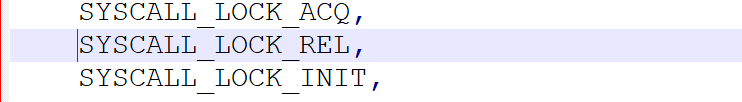
# 关键函数功能

在该实验中，修改PCB的结构体：



# 加分项测试案例设计

1. 增加三个系统调用，分别是



对应三个函数：



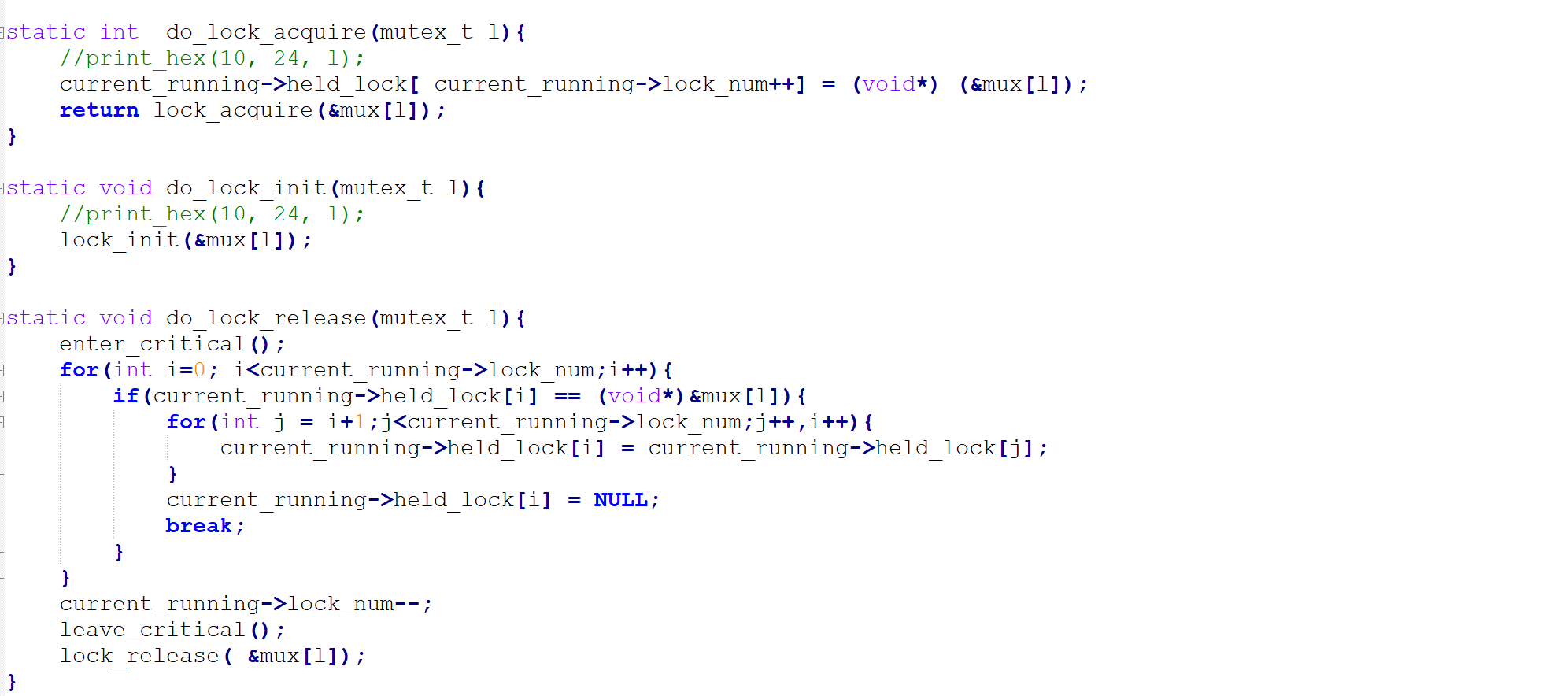
1. 我修改了三国的测试案例，刘备和孙权先申请自己的锁，并通过信箱操作来相互通信



然后二者分别申请对方的锁，此时二者都被阻塞，待曹操随机杀掉一个进程后，另一个进程得到对方的锁，并打印：



1. 三个系统调用函数如下：



我在PCB中加入了记录该进程申请锁的变量

