操作系统Lab7实验报告

2012011350 计24 周琳钧

练习0：填写已有实验

Lab7无需新补充任何内容，直接merge即可。

练习1：理解内核级信号量的实现和基于内核级信号量的哲学家就餐问题

1. 实验过程：本实验需要解决哲学家就餐问题，不需要编码，只需理解基于信号量的同步互斥的实现过程。信号量是反映当前资源的使用情况的一个变量，如果value>0说明资源空闲，value=0说明资源被使用且无等待进程，value<0则说明资源被使用且有等待进程。为此我们实现了两个函数\_\_up和\_\_down来描述这一过程。\_\_up函数当无等待进程时将value加1，有等待进程时则直接切换唤醒队列头进程，\_\_down函数在value>0时只需将其减1，否则将当前进程置入等待队列阻塞，然后调度器选择另一个进程执行。我们回到哲学家就餐问题，主函数在check\_sync.c中的check\_sync函数的第一部分。mutex是临界区信号量，初始化为1（一开始临界区中什么都没有，资源空闲），s[i]是每一个哲学家独有的信号量，初始化为0，对每个进程，都要循环执行思考-饥饿-吃饭-思考……这一过程。首先每个进程在拿起叉子前要先进入临界区，临界区保证了每次最多只有1个进程能够询问是否能获取资源，然后如果该进程发现左右邻居都没在吃饭，则它可以享用两个叉子，并up自己的信号量，否则进入阻塞状态。当进餐结束后进程再次进入临界区，放下叉子并且通知他左右邻居现在可以使用他的叉子，退出临界区，本轮循环结束，如此进行下去可以保证同步互斥。

练习2：完成内核级条件变量和基于内核级条件变量的哲学家就餐问题

1. 实验过程：本实验要填写4部分的代码，分别对应两个文件：monitor.c（管程控制文件）和checksync.c（管程执行文件）。下面具体描述一下管程机制。实际上，管程就是一个封装了共享资源的结构，任何要访问该资源的进程必须通过管程来访问并获取，外部进程无法访问。因此，我们要实现的管程结构实际就是对练习1的一次封装。练习1中的信号量mutex变成了管程结构中的互斥变量，而练习1中信号量s[i]变成了管程结构中的条件变量。需要说明的是，由于管程一次只允许一个进程进入，若想防止死锁的出现，就要能够让该进程去激活其他进程，为此我们定义了两个新变量next和next\_count，表示睡眠进程链表和其个数。然后，我们将up和down两个函数扩展成为了cond\_signal和cond\_wait两个函数作为管程的总控函数。cond\_wait的原理是：若有进程在等待，则激活之，否则打开管程以接收新进程，然后让本进程进入等待状态，cond\_signal的原理是唤醒其他进程然后让本进程进入休眠状态。具体到ucore程序而言见代码即可。