OSlab4实验报告

王亚楠 141220107

邮箱：[wyn1996@hotmail.com](mailto:wyn1996@hotmail.com)

# 实验进度

已经完成了所有实验要求，包括信号量的创建、销毁以及P、V操作，另外也实现了多创建线程来共享空间。

使用的gcc版本：32位

# 实验心得

1. 关于信号量  
   信号量有三个成员：int token（指示信号量的值）、int used（指示该信号量是否被使用）、ListHead wait（指示该信号量的等待队列）。  
   我模仿pcbpool创建了一个sempool数组来存储信号量，并且通过信号量数组的下标来对信号量进行相应的处理。  
   sem\_t sem\_open(int value);——在sempool中找到一个空闲的信号量，初始化信号量，并且标记该信号已使用，然后返回该信号量在sempool中的下标，以方便后续对该信号量的处理。  
   void sem\_close(int index);——根据信号量的下标江信号量销毁。  
   void sem\_wait(int index);——将信号量的值减1，并且如果信号量小于0，那么就将当前进程加入到该信号量的等待队列。  
   void sem\_post(int index);——将信号量的值加1，并且如果该信号量内有阻塞的进程，则将进程将到等待队列。  
   然后将信号量的相关操作封装成系统调用，以提供给用户进程使用。  
     
   我起初尝试将传给信号量的参数设置为semaphore\*，通过链表来存储各个信号量，仿照pcb创建free和run队列来存储信号来干，这样虽然能够很好的处理好单个信号量的操作，但是对于多个信号量，我一直无法处理好对特定的信号量的操作，难以从free或run队列中取出特定的信号量，从而使得多信号量的操作变得很困难。目前我也没有想好如何通过该方法实现信号量的相关操作。
2. 关于线程  
   我主要模仿了进程的fork操作，不过创建的线程之间共享数据区和代码区，而且基地址都是进程的基地址，每个线程之间有自己独立的栈。栈的分配也是类似于fork操作的，根据每个线程的线程号从栈的起始地址往上长0x400000的空间。  
   void create\_kthread(void\*);——我通过传递一个函数来创建线程，并且将函数的地址赋给了eip，esp的话指向该线程所占有的栈的栈顶。  
   通过多线程也使得能够共享内存得以实现。  
   同样，我也将线程的创建封装成了系统调用，以提供给用户进程使用。
3. 关于生产者和消费者的测试  
   我按照2012级讲义写了生产者和消费者问题，一开始我先P(mutex)再P(full)（或P(empty)），这样会使得生产者和消费者全部被挂起，加入到信号量的等待队列中，从而出现问题。通过查看每次mutex、full、empty的值才发现了这个问题，也是困惑了我很久，让我查了很久的P、V操作。  
     
   经过修改后，我实现的生产者消费者问题如下：有两个生产者和一个消费者，生产者和消费者的生产、消费效率一致。通过这个测试我也检验了我的信号量PV操作以及多线程的操作。