103011227 周延儒

1. [20 points] 7.4 In Section 7.4.4, we describe a situation in which we prevent deadlock by ensuring that all locks are acquired in a certain order. However, we also point out that deadlock is possible in this situation if two threads simultaneously invoke the `transaction()` function. Fix the `transaction()` function to prevent deadlocks.

```
void transaction(Account from, Account to, double amount) {
   Semaphore lock1, lock2, lock3;
   wait(lock3);
   lock1 = getLock(from);
   lock2 = getLock(to);
   wait(lock1);
         wait(lock2);
               withdraw(from, amount);
               deposit(to, amount);
          signal(lock3);
          signal(lock2);
   signal(lock1);
}
```

2. 7.7 Consider a system consisting of four resources of the same type that are shared by three processes, each of which needs at most two resources. Show that the system is deadlock free.

Suppose the system is deadlocked. And it will imply that each process is holding one resource and is waiting for one more.

Because there are 3 processes and 4 resources, one process must be able to obtain two resources. And This process requires no more resources. Therefore, it will return its resources when done. In brief the system is deadlock free.