Ivan Bogdev

27053444

COMP 346 – Operating Systems

Programming Assignment 1

Task 1.

A consistent state of the account array is not maintained because the different threads interfere with each other. This happens when the "withdraw" or "deposit" methods are called. For instance, when "balance = balance – amount" is called, the java VM does three steps:

1. Gets the value of balance

2. Calculates balance – amount

3. Stores the resulting value in balance

The consistency is not maintained because when two or more threads try to change the same variable concurrently, sometimes the result of one overwrites the result of the other and consistency is lost. To fix this, each atomic action needs to be fully executed before another one is started. The threads need to wait for one another before changing the value of balance.

Task 2.

The start order of the threads is determined by the order in which the "thread.start();" is executed. In this case deposit [1], withdraw [1], deposit [2], withdraw [2], and so on.

Lifetime of a thread

First a thread is created with the state Runnable, then run by doing "thread.start();". The thread scheduler takes control and determines which thread will be run at the current moment and gives it the state Running. Depending on the time slicing, a thread can be Blocked or waiting, in which case it waits. When it is done waiting, it will continue its execution until finally, it finishes. When it finishes, the state will be changed to Dead.

The consistency of the accounts is not preserved when changing the start order of the threads.

Task 3.

See .java files in task3.

Task 4.

See .java files in task4.

Task 5.

Both of the solutions preserve the consistency of the accounts. However, synchronized block is more efficient than synchronized method when used properly. Synchronized block will make the execution of the thread not interruptible, while the rest of the method can be interrupted. In the case of the synchronized method, the whole method cannot be interrupted.

The synchronized block is more efficient because the concurrency is only interrupted during the execution of a small chunk of atomic operations, while the synchronized method can block other threads for longer periods of time especially if it does a lot of computations.