CS/CE/TE 6378: Advanced Operating Systems
Section 0U1
Programming Project 1

Instructor: Neeraj Mittal

Assigned on: Thursday June 7, 2012
Due date: Thursday June 28, 2012

This is a group project. A group may contain up to two students. *Code sharing among groups is strictly prohibited and will result in disciplinary action being taken.*

You can do this project in C, C++ or Java. Since the project involves socket programming, you can only use machines netXX.utdallas.edu, where XX $\in$ {01, 02, .., 45}, for running the project. You are expected to demonstrate the working of this project on netXX machines to the instructor and/or the TA.

1 Project Description

It consists of two parts.

1.1 Part 1

You are required to develop a simple distributed banking system. The banking system is implemented using a set of servers, where each server hosts a set of accounts. You can assume that the set of servers in the system and the set of accounts hosted by a server are both static and do not change at run time. The information required to initialize the banking system (*e.g.*, the number of servers, the location of each server, the set of accounts hosted by each server, the initial balance in each account) is stored in a configuration file. Develop your own format for the configuration file.

A user accesses his account using a client program. A client program allows a user to perform one or more of the following actions:

1. Check the balance of an account.
2. Deposit money into an account.
3. Withdraw money from an account.
4. Transfer money from one account to another.
5. Determine the total amount of money in the banking system by taking a consistent snapshot of the system. A snapshot of the system should display the amount of money in each account and the amount of money being transferred. You can choose to implement any of the algorithms for recording a consistent snapshot of the system discussed in the class.

Take suitable precautions to ensure that an account balance never becomes negative. Also, when transferring money, *add a random delay* between when the money is withdrawn from the source account and when the money is deposited into the destination account.

1.2 Part 2

Implement a testing mechanism to ascertain that your snapshot algorithm is indeed taking a consistent snapshot of the system. You will graded on the thoroughness of your testing mechanism.

2 Submission Information

You will have to submit your project using *eLearning*. You are required to submit all the source files necessary to compile the program and run it. Also, submit a README file that contains instructions to compile and run your program.