Report by group 12

T3-Remote File Sync

Intruduction

This file synchronization is an example of a multi-threaded client-server program. It handles the synchronization of a file between client and server and can be executed with either client or server pushing.

The source code in the one package that contains Client, FileSender,MessageSender and Server classes.

What we have done:

1: FileSender.java

This class implements **java.io.Serializable** interface. We used serialization in our code to convert files (object) into bytes to send it over the network. We need serialization because the network infrastructure cannot send java objects because it understands just bytes and not java objects.

2: MessageSender.java

This class implements serializable interface to send and receive messages, It has the same serialVersionUID, because if the reciever has loaded a class for the object that has different UID than that of corresponding sender's class, the Deserialization will result in an **InvalidClassException**. A Serializable class can declare its own UID explicitly by declaring a field name, we declared it as static, final and of type long.

3: Client.java

Our client class collects information about all files in a given directory through the java.io.File class, stores them in a vector data structure. The client then establishes a connection with a file synchronization server and sends the serialized list of files to the server along with the client identifier.

4: Server.java

After de-serializing the list, the server then checks its record of files for the given client and identifies the files that are new or different The list of the new files is sent back to the client .

How to use our programs:

If you are going to test run the program then you must change the file FileSender.java and change both filepaths to some folders in your own computer.

Client can take 2 commands, exit to shut down, and type sync, then file comparison begins between server folder and client folder