Readme.pdf
Dennis Kager
Ali Awari

Asst3: WTF?

Synopsis:
The objective of this assignment is to create a multi-threaded server which is capable of handling multiple clients and implements various tools for versioning control.

Design:
This project is intricately designed through many C and Header files. The following is a list of the files that were used to create this project:

1. TCPserver.c and TCPserver.h are responsible for creating the server and handling multiple clients along with their input. Multi-threading is achieved by having a loop to accept clients. Then when a new client connects, the connection file descriptor is stored in a global array to keep track and a detached-thread is created for the client to allow simultaneous operations of handling the client and accepting new connections. Detached threads are used so it is not needed to keep track of the threads to join them. Additionally, the thread identification numbers are kept track of in a global array. When the server is terminated via CTRL + C, the signal is caught and a function is run to shut down the threads before the server.
2. SocketLibrary.c and SocketLibrary.h are used to manage the socket connection for the server, such as initializing, binding, and listening. In addition, these are also responsible for handling the transfer of files between the server and client. Sending compressed Tar files is also supported through a different set of file commands, as it requires a more difficult algorithm to make sure absolutely no data is lost.
3. Client.c and Client.h are used as the interface in which the general user will interact with. These files take the user's input to contact the server and manage local projects through various commands. If the client sends a command and cannot connect to the server, it will repeatedly try to connect every 3 seconds until it can connect or is stopped with CTRL + C.
4. WTFCommands.c and WTFCommands.h are used primarily to store the client's commands such as add(), remove(), checkout(), create(), destroy(), update(), upgrade(), etc.
5. A single makefile is used to create both the server and client executables