CS214 Project 3: Where's The File

Michala Rose, Section: 3 ; Nikita Kolotov, Section: 2

Compilation:

- **Client**
  gcc -o WTFclient WTFclient.c -lpthread

- **Server**
  gcc -o WTFserver WTFserver.c -lpthread

- **WTFtest**
  gcc -o WTFtest WTFtest.c

Design:

- **WTFclient.c**
  - **void sighandler(int sig)** - Used in main(). Will clean up all loose ends and safely exit the program when an interrupt or segfault occurs.
  - **int main(int argc, char const *argv[])** - Will read and format the user arguments, then puts them into appropriate helper functions to connect to the server and achieve whatever goal was set forth in the arguments, this assumes and checks to make sure that the arguments are all in order.
  - **Helper Functions:**
    - Tarring Functions:
      - Create a tar file from an existing folder
      - Create an empty tar file to be written into by the opposing connection
      - Remove a designated tar file after untarring to a directory
    - Manifest manipulation:
      - Add new entry to manifest with new version and hashcode
      - Remove an entry from the manifest and then reformat it to keep structure
      - Read in manifest and break up every project file into tokens
    - Scanlib: Scans the entire folder that the client is native to and brings back the address of an item of interest. If no item is found then nothing is returned and we use this to make sure that the client fulfills all requirements for different commands.
    - Readdit: Will can the client directory for the .configure file. If it cannot find it then it will return nothing which prompts main to know that it does not exist. This will trigger the fail condition.

- **WTFserver.c**
    - **int SocketHandler(char* name, int sock)** - Invoked in main. When a new connection is accepted and a thread is created, it is sent into this function to be the hub function. Here any command that the client will have will be processed and carried out, with the help of a couple functions. These functions are similar to the ones in the client side where it will check libraries for project existence and complete tar commands.
    - **void handle_sigint(int sig) -** Waits for each thread to return. Destroys all semaphores and mutexes and then exit(0).
    - **int main(int argc, char const *argv[]) -** Creates the mutex and socket. Binds to the client. Listens for incoming connections. Once a connection is made it is then sent off into the socket handler function. After that it takes care of joining all connection.
- **WTFtest.c**
    - Will test the different test situations listed in testcases.txt.

Assumptions:

- The server and client executables would be in different folders to symbolize that the server and client are not the same machine. Otherwise this project would be
- Inputs will always be correct
- No more than 50 connections will be attempted on one server.

Difficulties:

- We had a lot of confusion with how update, upgrade and the other UMAD functions were meant to be implemented. Piazza only furthered confusion. We assumed that this sequence of commands would be entered when the client wished to add a version to the repository but that did not appear to be the case when we read about commit and push.