Griffin Lynch- gal69
Arbazkhan Pathan - ap1415

Systems Programming - Spring 2019
Asst3 - Where's the File (WTF)

Implementation:

- To implement our program we first decided that we must create a server (WTFserver). This server would act as our repository which would allow us to use the client (WTFclient) to access the files and directories in the folder.
- Our code does a multitude of functions such as displays all sorts of error messages when the user inputs things in the incorrect format or the function they wish to complete is not possible. Such as if the user wanted to delete an item in the server which was never in the server.

WTFserver:

- Our server had to be multithreaded which means that it could be accessed from a number of sources. This is needed for our server because our server should be able to be accessed from multiple clients at the same time if needed. Our server makes a new thread every time another connection to a server is made. However, it is noteworthy to say that doing all of this threading without a safety net is rather dangerous.
- In this project our safety net was the use of "mutex". Mutex is a special hardware instruction. Our mutexes main purpose is to block other contexts from entering the critical section while on e context is in it. Mutex has a very important "lock" function which prevents data from becoming lost and or destroyed. Mutexes are also very dangerous because there are a few things we had to be very careful about when making this project. First we could not lock more than once, or unlock more than once, or accidently unlock something that was never locked. This was incredibly nerve wracking because we risked potentially deadlocking which would leave our code in a state where it could not run.

- Lastly, we needed to create a way to shutdown all of the threads working at the same time. We used the function atexit() and a signal to make sure that the server and the client could be shut down with a single command. Our code server and client can be stopped with “ctr+c”. We included the signal.h and when that command is hit then the signal flag is thrown and it will stop the program.

WTFclient

- Our client was made to run many functions just as “git” would. The client throws out all of the error and the complete messages to the user. First we needed to get a single thread from the server so that we could establish a connection to the server. The client is all the user would need once they got the server to start. In the client we needed to use sockets which allowed us to get filedescriptors. We could then read and write to the file which is exactly what we needed. We were then able to manipulate the files and make our functions do as they were meant to do.
- However when we downloaded the files we sometimes were stuck dealing with bits. This required us to do some bit manipulation. For this part we were forced to do a lot of research and learn by trial and error. We also had to be very careful not to mess up our metadata with a command or anything else.
- Lastly we used threading again to check if anything exists or doesn’t exist in the server.