Malav Doshi- md1378

Herik Patel-hdp48

Where is the file?

Objective of the project

Purpose of the assignment is to create a version control system that performs task like storing, updating ,removing ,adding and to perform few other functions that are to similar to the functions performed by the a version control used people to working on one project

Design of the algorithm

For the version control system, we need to make connection between clients and servers for this we are using socket to connect them. But this creates one problem when multiple clients are connected to the server and trying to make changes to one file so to deal with this issue we are using multi thread with mutex locks so that this does not cause a problem because it may led to situation in which different people on same server are editing different versions of the file. So mutex lock unlock ensures that changes are only made to latest version of the file. While editing different versions of the project are also being stored to protect from loss of latest version or if we need see any previous changes made.

So basically, we are storing all the changes made to the file since the beginning. We are using commit/push so that we store changes that are being made locally to the repository. When we push a, new directory within the project directory is created which will be the new version directory with its own .Manifest . The version number starts with 1. The assignment supports command necessary to create a project folder with the given name which will then set local version on client side and the changes would record here and then sent to server. We can also delete any file on the server by locking the repository and add will let the client to add entry for file which is stored on client side and it is his/her own file. And the client able to access the current version of the file and make changes and these changes will be updated as current version of the project. Client has option to rollback to its previous version of the project by using rollback command which gives client access of the previous version of the project. While requesting any of

the commands it will make sure that project/file is already present in the directory and if there is no file with the given name then the command given to fetch the file from the server will fail.

Name of the files in the assignment and their purpose

Socket.h: This header file is used to read from sockets. This file gives us a linked list which contains all the data read from the socket. The data is stored in form of tokens which are separated by the delimiter it is given. It can take any char as delimiter.

Manifest.h: This header file works as a medium to transfer linked list between the server and the client. With the help of this header file a .Manifest file can be read as a linked list which can then be easy to traverse and get all the information for a particular file.

Commit.h: The changes made by client are stored locally first in a linked list and then they are uploaded to the server. This function is completed by commit.h which is uploading the changes made to the project locally to the repository and ensure the current version the repository is up to date.

Server.c: This c file serves as servers and performs functions which are requested by the client .

Client.c: This c file serves as client and it allows you to make changes to the project by making connections with the server and requesting stuff like current project version or previous version.

Filehash.h: It computes the MD5 hash of the file used to store the versions and then this stored hash is used to compare with the client's version.

Final Result

We tested our program using several different test cases and tried to simulate a version control system which is almost identical to the one used in the real world and tried to operate and test it with consideration of all the problems that version control might face in real world.