CS 214: Systems Programming Spring 2019

Shubham Mittal sm1841 Section 02

Devvrat Patel dap299 Section 08

# Project 3: Where's the file?

In this Assignment we had to create a version control system. The project was based on a client server architecture. The commands we had for the client side were the following:

1. Checkout:
   Client will request the project from the server. The server will look for the project and it will send the latest version. The client then will create the project.
2. Update:
   Client will request the server's current .Manifest file for the project. It will compare it with its own .Manifest file. The files will be recorded with specific tags signaling the changes that are made:
   - U (upload), M(modify), A(add), D(delete).
3. Upgrade:
   This will use the update the client and do three things based on the tags of the files that were given.
   - M : fetch the file from server and replace the client side file
   - A : same as the M tag
   - D : delete the client file
4. Commit:
   Same as the update in the client will get the server's manifest and compare them; except now we are going to change the server.
5. Push:
   Similar to the upgrade except this time the changes will be made on the server side. The server will look at the tags and make changes to the file based on them. The server will send a success message to the client. Upon receiving the message the client will delete the .Commit file.
6. Create:
   Client will send the create command with the project name. The server will create a subdirectory with the giver project name.
7. Destroy:
   The client will tell the server to destroy the project. The server will go in the project file and delete every file within.

8. Add:

   The client will add a file entry to .Manifest file of the project with a new version of hash.

9. Remove:

   The client will remove the file entry from the .Manifest file.

10. Currentversion:

    The currentversion command will ask the server for the current state of the project. The client will output a list of files under the project's name along with their version.

11. History:

    The client will request a history. The server will send a file that will contain all the pushes since the project was created. The output will be similar to the update output, but with a version number and newline separating each push's log of changes.

12. Rollback:

    The client will request a rollback. The server will find the version specified and delete all greater versions.