Akshath Sivaprasad

Basic Storage Design Plan for Gitlet

Overview
To avoid redundant files are hashed and stored. In addition, they are compressed to save space. When files are hashed, only the content is used (not the file name), so that when we have two different files, we will output the same hash, so it is not necessary to copy the file again. Files, directories, subfiles, subdirectories etc. are tracked with .tree files. In other words, a tree represents a snapshot of the project.

To be more specific with how files are saved and tracked, there will be three folders: commits(which will store all commit files), trees(which will store all tree files, one for each commit), and files(which stores all files/folders in the working directory). Commits consist of a parent commit, a tree, and a message. Trees contain lists of hashed files, as well as their respective filenames.

Error Handling

- Try-catch in Gitlet main that catches all unhandled exceptions; catch prints stack trace to help debugging
- "Warning: The command you entered may alter the files in your working directory. Uncommitted changes may be lost. Are you sure you want to continue? (yes/no)" is handled in one place (copyFile() method)
- Code is structured to retrieve maximum info possible even when errors are encountered (e.g. trying to restore an old version of a file that doesn't exist because a user went into the .gitlet folder and deleted the backup)

File list

- Gitlet.java – parse user input; call commands based on input from Commands.java
- State.java – class to manage the state (serialization/deserialization)
- Commands.java – implements all user-facing Gitlet commands
- FileHelper.java – back-end interface for storage of files (uses hashing to optimize)

Sequential Example

For each of the commands below, describe in detail what files on your computer change after the command is executed. Only list changes made to files. Do not list details regarding the reading of files.

```
$ java Gitlet init
```

- If it does not already exist, a new folder called .gitlet is created in the current directory
- .gitlet subfolders (commits, files, trees) created to keep track of state

- Empty file state.ser is created under .gitlet (where state is stored)

```
$ java Gitlet add dog.txt
```

- state.ser modified to include dog.txt

```
$ java Gitlet commit “initial commit of dog.txt”
```

- check if files exist in state.ser, in this case, yes
- since there is no parent commit, pointer to parent points to null
- commit file created in .gitlet/commits
- a tree corresponding to this commit is created in .gitlet/trees
- dog.txt is hashed and stored in .gitlet/files
- clear state.ser

```
$ java Gitlet add dog.txt
```

- check to see if **any changes have been made (using hash values)** or file has not been previously added
- in this case, no changes have been made, print the following error message: File has not been modified since the last commit.

```
$ cp manyDogs.txt dog.txt
$ java Gitlet add dog.txt
```

- check to see if **any changes have been made (using hash values)** or file has not been previously added
- assuming changes have been made, add to state.out

```
$ java Gitlet commit “replaced dog.txt with manyDogs”
```

- check if files exist in state.out, in this case, yes
- parent commit points to “initial commit…” commit made above
- commit file added to .gitlet/commits
- a tree corresponding to this commit is created in .gitlet/trees
- dog.txt is hashed and stored in .gitlet/files
- clear state.out

```
$ java Gitlet add manyDogs.txt
```

- check to see if any changes have been made (using hash values) or **file has not been previously added**
- add manyDogs to state.out

```
$ java Gitlet add ketchupFriend.txt
```

- check to see if any changes have been made (using hash values) or **file has not been previously added**
- add ketchupFriend to state.out

```
$ java Gitlet commit “added manyDogs and ketchupFriend”
```

- check if files exist in state.out, in this case, yes
- parent commit points to “replaced dog.txt…” commit made above
- commit file created in .gitlet/commits
- a tree corresponding to this commit is created in .gitlet/trees
- ketchupFriend.txt is hashed and stored in .gitlet/files
- manyDogs.txt is NOT stored in .gitlet/files separately, because dog.txt already exists with identical content
- clear state.out

```
$ java Gitlet commit YOLO
```

- check if files exist, in this case, no, print “File does not exist” error

```
$ cp doge.txt manyDogs.txt
$ java Gitlet add manyDogs.txt
```

- check to see if **any changes have been made (using hash values)** or file has not been previously added
- assuming changes have been made, add to state.out

```
$ java Gitlet commit “replaced manyDogs with doge”
```

- check if files exist in state.out, in this case, yes
- parent commit points to “initial commit…” commit made above
- commit file added to .gitlet/commits
- a tree corresponding to this commit is created in .gitlet/trees
- dog.txt is hashed and stored in .gitlet/files
- clear state.out

```
$ java Gitlet add dog.txt
```

- check to see if **any changes have been made (using hash values)** or file has not been previously added
- in this case, no changes have been made, print the following error message: File has not been modified since the last commit.

```
$ java Gitlet commit “added dog yet again”
```

- check if files exist in state.out, in this case, no (because dog.txt had been cleared from state.out in replaced dog.txt… commit)