Assignment – 4

Version Control System setup and usage using GIT

Date : 23-02-2019

Aim : Version Control System setup and usage using GIT. Try the following features

1. Creating a repository
2. Checking out a repository
3. Adding content to the repository
4. Committing the data to a repository
5. Updating the local copy
6. Comparing different revisions
7. Revert
8. Conflicts and a conflict Resolution

1. Creating a repository

By using GitHub - GitHub is a code hosting platform for version control and collaboration

#Create a new repository on GitHub.

Steps

(a) In github profile, on the upper right corner click '+' and then select New repository.

(b) Write a name for repository, write a short description about repository and click create repository.

2. Checking out a repository

Checkout is the act of switching between different versions of a target entity. 'git checkout' command lets us navigate between the branches created by git branch

#To change into a different branch

Command : git checkout <brach_name>

3. Adding content to the repository

To add content to remote repository which is located in GitHub, we need to first initialize our local repository with git and upload our files to remote repository

Steps

(a) Create a new directory in local computer with the same name of the remote repository

Command : mkdir <directory_name>

(b) Initialize local directory with git

To do this go to created directory and initialize with git

Change the current working directory to your local project.

Command : cd <local_directory_name>

#To initialize local directory as Git repository

Command : git init (This creates a new subdirectory named '.git' that contains necessary repository files)

(c) Add the files in your new local repository

Command : git add . (. is used to add all files)

4. Committing the data to a repository

Stores the current contents of the index in a new commit along with a log message from the user describing the changes.

#To commit files that are added in local repository

Command : git commit -m “commit message”

5. Updating the local copy

The git push command is used to upload local repository content to a remote repository. Pushing is how you transfer commits from your local repository to a remote repository

Steps

(a) To add a new remote, use the git remote add on the directory where repository is stored at.

Command : git remote add origin <url address of remote repository>

(b) To add files in local repository to remote repository

Command : git push origin master (this will add files local directly to master brach)

6. Comparing different revisions

Comparing different version allow us to see changes we have made on same file. For comparing revisions we need to get the hash value

Steps

(a) Check the changes we have made

Command : git log (this will return list of diffenent files we have made changes recently with a hash value for corresponding changes)

(b) Comparing diffrent versions

Command : git diff <hash_of_change1> <hash_of_change2>

7. Revert

Revert some existing commits. Revert command rollback changes we have committed. It create new commit from a specified commit by inverting it. Hence adds a new commit history to the project, but it doesn’t modify the existing one.

Command : git revert HEAD~3

This will revert the changes specified by the 4th last commit in HEAD and create a new commit with the reverted changes.

8. Conflicts and a conflict Resolution

If we changed file in remote repository directly we will use a command pull, to pull in the remote changes to local computer. But if we forgot to pull the changes there will be conflict if we tries to push our files after changing local file.

Command : git push origin <brach name>

If the content of a file in GitHub website and local computer doesn’t match, there will be conflict. To fix the conflict pull the changes first.

Command : git pull origin <branch name>

Then we will result in conflict error message like “merge conflict in <brach-name>

And we can see conflict markers which shows us where the conflict occured. And there will be different version of the file seperated by equel (=) sign. Head will be most recent commit and the other will be the changes made on remote repository.

We can select which version to choose. Delete all other version and conflict marker. Then add the files and commit message and push it to required branch.

Command : git add <filename>
git commit -m “commit message”
git push origin <branch name>

And go to the pull requesr tab in remote repository and click “Merge Pull Request” and confirm the merge. It will be resulted with a message “Pull request successfully merged and closed”. This is how to fix conflicts.