Reset Command

Reset command is use to remove files from working directory, staging area and local repository. There are three types of modes according to which reset works

Format: git reset <mode> <ref no>

1. Mixed mode (--mixed): In this type of mode files are removed from local repo as well as staging area.

Command: git reset -- mixed <ref of previous commit>

Example:

In local repo

File 3(145876) (head)

File 2(458796 ) ------🡪 git reset –- mixed 458796 ; ------🡪 file2(458796)

File 1(789665) file1(789665)

1. Soft mode(--soft): only remove file from local repo and files in working area and staging area remain the same.

Command: git reset -– soft <ref of previous commit>

1. Hard mode: remove file from everywhere.

Command: git reset --hard <ref of previous commit>.

Git Branching

When we create a project there are several files and several number of commits. And by default, it all goes to the master branch (main branch).

If there are multiple developers working on different features, on that case if the developer wants to use the source code which is committed in the master branch and want to develop that feature above that.

Then the dev creates a new branch and till the date when dev creates a branch all the files which are present in the commit will be carried to new branch and change will not be been shown to the master branch.

Example: if we are working on the project in c++ in master branch we create all the classes and the variables and declare all the functions. And the other team mates will get that code and do their roles without disturbing the master branch by creating the branches.

Ever branch is independent master or branch 1, branch 2 etc.

1. To view available branches:

Command: git branch

\* Shows which is active branch or on which branch we are now.

1. Create new branch:

Command: git branch <branch name>

1. Switch to branch:

Command: git checkout <branch name>

Merging:

After performing all the process in the branch it’s time to merge the branches.

There are two types of merges:

1. Fast forward merge: in this case after creating the branch is we do not do any change or edits in the master branch then the process is quite simple and fast so this type of merge is called fast forward merge.

F1(MC)------>F2(MC)------------------->---------------------->F1+f2+f3+f4

| |

|branching |

|\_\_\_\_\_\_\_\_\_\_\_\_\_>F3(FC)---------->F4(FC)

Where MC represent Master commit and FC represent Feature branch commit.

1. Three-way merge: in this type of merge when there are changes in master and feature branch after the branch created then this type of merge case collision so there is a automatically a new commit is created.

Command:

To merge the branches, we have to first switch to master branch then we can perform merge.

git merge <branch name>

To delete the branch after merge: git branch -d <branch name>.