DHARMSINH DESAI UNIVERSITY

MCA SEM -2

DATA STRUCTURE USING C

THE DOCUMENTATION OF THE WORKING OF GIT AND GITHUB

NAME : GUJARATI POOJAN P.

ROLL NO: MA004

STUDENT ID : 18MAPOG013

E-MAIL : poojangujarati20@gmail.com

SUBMITTED TO,

PROF. Himanshu Purohit

Before moving further about working of git and github. what is git and github?

Git :

Now, git is nothing but is a free and open source distributed version control system.which is designed to handle everything from small to very large project with speed and efficiency

- Git is easy to learn and has tiny footprint with lightning fast performance
- Git is released under the GNU General Public License version 2.0, which is an open source license.
- **Git is vey fast**. With Git, nearly all operations are performed locally, giving it a huge speed advantage on centralized systems that constantly have to communicate with a server somewhere.
- Git was built to work on the Linux kernel, meaning that it has had to effectively handle large repositories from day one.
- Git is written in C, reducing the overhead of runtimes associated with higher-level languages.
- Speed and performance has been a primary design goal of the Git from the start.

another feature of git is that git has integrity:

Now what does it means? Everything in Git is checksummed before it is stored and is then referred to by that checksum.

This means it’s impossible to change the contents of any file or directory without Git knowing about it. This functionality is built into Git at the lowest levels and is integral to its philosophy.

You can’t lose information in transit or get file corruption without Git being able to detect it.

The another phase is that git is only adds data :

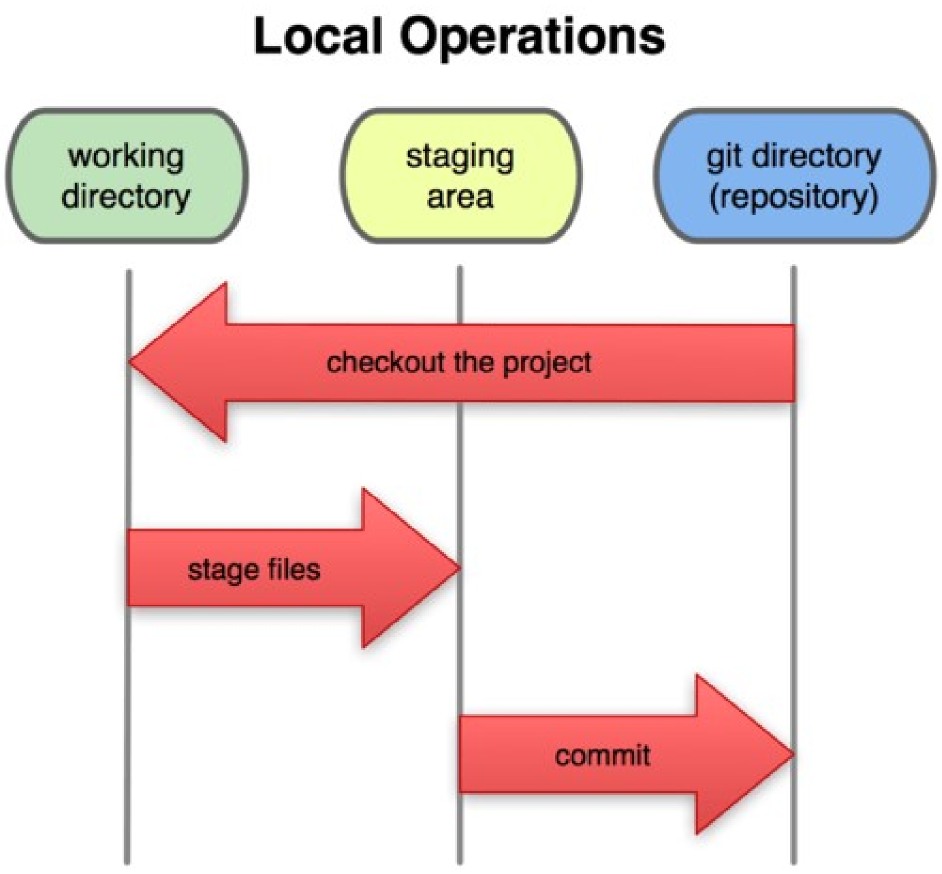
When you do actions in Git, nearly all of them only *add* data to the Git database.It is hard to get the system to do anything that is not undoable or to make it erase data in any way.

As with any VCS, you can lose or mess up changes you haven't committed yet, but after you commit a snapshot into Git, it is very difficult to lose, especially if you regularly push your database to another repository.

Now, moving further about the core part of git is that the three states.

if you want the rest of your learning process to go smoothly. Git has three main states that your files can reside in: *committed*, *modified*, and *staged*:

- Committed means that the data is safely stored in your local database.
- Modified means that you have changed the file but have not committed it to your database yet.
- Staged means that you have marked a modified file in its current version to go into your next commit snapshot.



This leads us to the three main sections of a Git project: the Git directory, the working tree, and the staging area.

- The Git directory is where Git stores the metadata and object database for your project. This is the most important part of Git, and it is what is copied when you *clone* a repository from another computer.

- The working tree is a single checkout of one version of the project. These files are pulled out of the compressed database in the Git directory and placed on disk for you to use or modify.

- The staging area is a file, generally contained in your Git directory, that stores information about what will go into your next commit.

The basic git workflows goes on like as below:

1. You modify files in your working tree.
2. You selectively stage just those changes you want to be part of your next commit, which adds *only*those changes to the staging area.
3. You do a commit, which takes the files as they are in the staging area and stores that snapshot permanently to your Git directory.

If any particular version of a file in the git directory, It is considered as committed. if it has been modified and was added to the staging area, it is staged. And if it was changed since it was checked out but has not been staged ,it is modified.

Github:

Before moving further about working of github what is github?

- GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.
- Github has certain essential like *repositories*, *branches*, *commits*, and *Pull Requests*.
- You can create your own repositories in github.

Certain steps regarding to create repositories in github which briefly explain as below :

Step 1: to create repositories

- A **repository** is usually used to organize a single project.
- Repositories can contain folders and files, images, videos, spreadsheets, and data sets – anything your project needs.
- We recommend including a *README*, or a file with information about your project.
- GitHub makes it easy to add one at the same time you create your new repository.
- *It also offers other common options such as a license file.*
- Your any repository can be a place where you store ideas, resources, or even share and discuss things with others.

Hence from the above briefs we can say that its works in simple way and very useful for backup.

Now, to create new repository certain steps has follow.

1. In the upper right corner, next to your avatar or identicon, click + and then select **New repository**.
2. Name your repository (i.e. datastructure).
3. Write a short description.
4. Select **Initialize this repository with a README**.

Step 2: to create new branch

- **Branching** is the way to work on different versions of a repository at one time.
- By default your repository has one branch named master which is considered to be the definitive branch.
- We use branches to experiment and make edits before committing them to master.
- When you create a branch off the master branch, you’re making a copy, or snapshot, of master as it was at that point in time.
- If someone else made changes to the master branch while you were working on your branch, you could pull in those updates.

Now , how to create a branch

1. Go to your new repository (i.e. datastructure)
2. Click the drop down at the top of the file list that says **branch: master**.
3. Type a branch name, readme-edits, into the newbranch text box.
4. Select the blue **Create branch** box or hit “Enter” on your keyboard.

step 3: make and commit changes

- On GitHub, saved changes are called *commits*.
- Each commit has an associated *commit message*, which is a description explaining why a particular change was made.
- Commit messages capture the history of your changes, so other contributors can understand what you’ve done and why.

Make and commit changes

1. Click the README.md file.
2. Click the pencil icon in the upper right corner of the file view to edit.
3. In the editor, write a bit about yourself.
4. Write a commit message that describes your changes.
5. Click **Commit changes** button.

Now, after a commit changes occurs another phase is that open the pull request.

Step 4: Open a pull request

Pull request are the core or we can say that the heart of the collobation of github.

When you open a *pull request*, you’re proposing your changes and requesting that someone review and pull in your contribution and merge them into their branch.

- Pull requests show *diffs*, or differences, of the content from both branches.
- The changes, additions, and subtractions are shown in green and red.
- As soon as you make a commit, you can open a pull request and start a discussion, even before the code is finished.

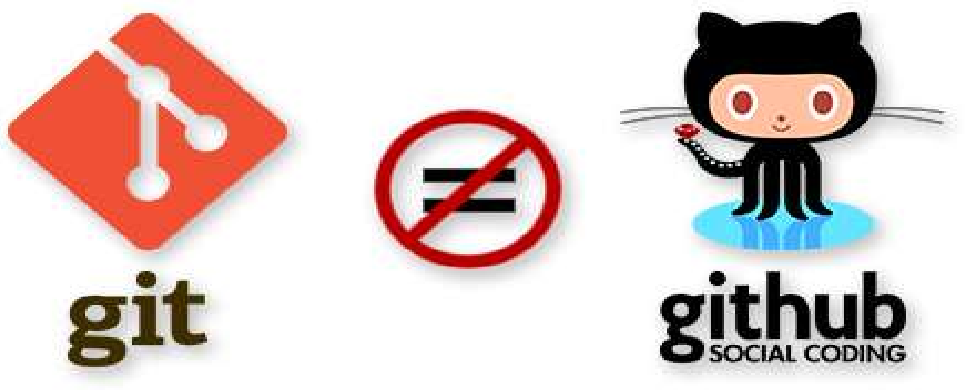
- You can even open pull requests in your own repository and merge them yourself.
- It’s a great way to learn the GitHub flow before working on larger projects.

What is difference between git and github ?

major or we can say that core difference between git and github is that Git is a distributed version control tool that can manage a development project's source code history, while GitHub is a cloud based platform built around the Git tool.

Git is a tool a developer installs locally on their computer, while GitHub is an online service that stores code pushed to it from computers running the Git tool.

Another key difference between git and github is that The key difference between Git and GitHub is that Git is an open-source tool developers install locally to manage source code, while GitHub is an online service to which developers who use Git can connect and upload or download resources.



One way to examine the differences between GitHub and Git is to look at their competitors.

Git competes with centralized and distributed version control tools such as Subversion, Mercurial, ClearCase and IBM's Rational Team Concert.

On the other hand, GitHub competes with cloud-based SaaS and PaaS offerings, such as GitLab and Atlassian's Bitbucket.

Difference between git and github continue…

- Git is maintained by linux foundation since 2005.
- While github is maintaining by Microsoft since 2018.
- Git has have not user management tool.
- While github has built-in usermanagement tool.

Conclusion

As per the above explanation about working process of git and github we can conclude from that is git is distributed version control system written by the creator of linux. while github and similar services bring all of the benefits of a decentralized VCS to a centralized services.

Bibliography

https://www.quora.com

https://guides.github.com/

https://www.theserverside.com

https://guides.github.com