1-The process of monitoring and controlling changes to software code is known as version control, commonly referred to as source control. Software technologies called version control systems assist software development teams in tracking changes to source code over time.

2-Software technologies called version control systems assist software development teams in tracking changes to source code over time. Version control solutions enable software teams to operate more swiftly and intelligently as development environments have increased.

3-Version control systems come in two types: distributed and centralised.

4-Differences between distributed version control and other version control systems

Users can access a repository from different places using a distributed version control system (DVCS).

Developers who must work on projects from numerous machines or who must communicate with other developers remotely frequently employ b-DVCSs.

Version control that is centralised

All users of a centralised version control system (CVCS) collaborate on the same central repository.

When a group of developers wants to exchange code and keep track of changes, centralised version control systems are frequently utilised.

5- Git is a DevOps tool used for source code management.

6- Features of Git

Tracks history

Free and open source

Supports non-linear development

Creates backups

Scalable

Supports collaboration

Branching is easier

Distributed development

7-

a-gitinit-Initializes a new Git repository. If you want to place a project under revision control, this is the first command you need to learn.

b-git commit-Takes the staged snapshot and commits it to the project history. Combined with git add, this defines the basic workflow for all Git users.

C-git pull-Pulling is the automated version of git fetch. It downloads a branch from a remote repository, then immediately merges it into the current branch. This is the Git equivalent of svn update.

8- No git and Git hub are not same.Below is the reason

The functionality of Git and GitHub differs significantly. They both offer source code management (SCM), which facilitates merging and sharing of code, but that's about where their similarities end. Imagine Git as a single computer and GitHub as a network of many linked computers, each with a distinct function in the process of getting there.

9-git –version

10- git add

11-the distinction between the commands git status and git log

Git status: The git status command shows the current status of the staging area and the working directory.

A useful tool for reviewing and reading a repository's history is called log-Git log.

12-git init

14-It's untrue because if you neglect to run git add, all the newly produced files would be gone from your project. It would appear that I should be able to commit the whole directory, together with any and all new files, in a single command without having to specifically tell the computer to include all new files in the commit in a prior command.

13-our files may be in one of three stages in Git: edited, staged, or committed.

Modified refers to the fact that you have made changes to the file without yet committing them to your database.

command- git add

Staged refers to the designation of a changed file in its present state for inclusion in your subsequent commit snapshot.

command-git commit -m”your message”

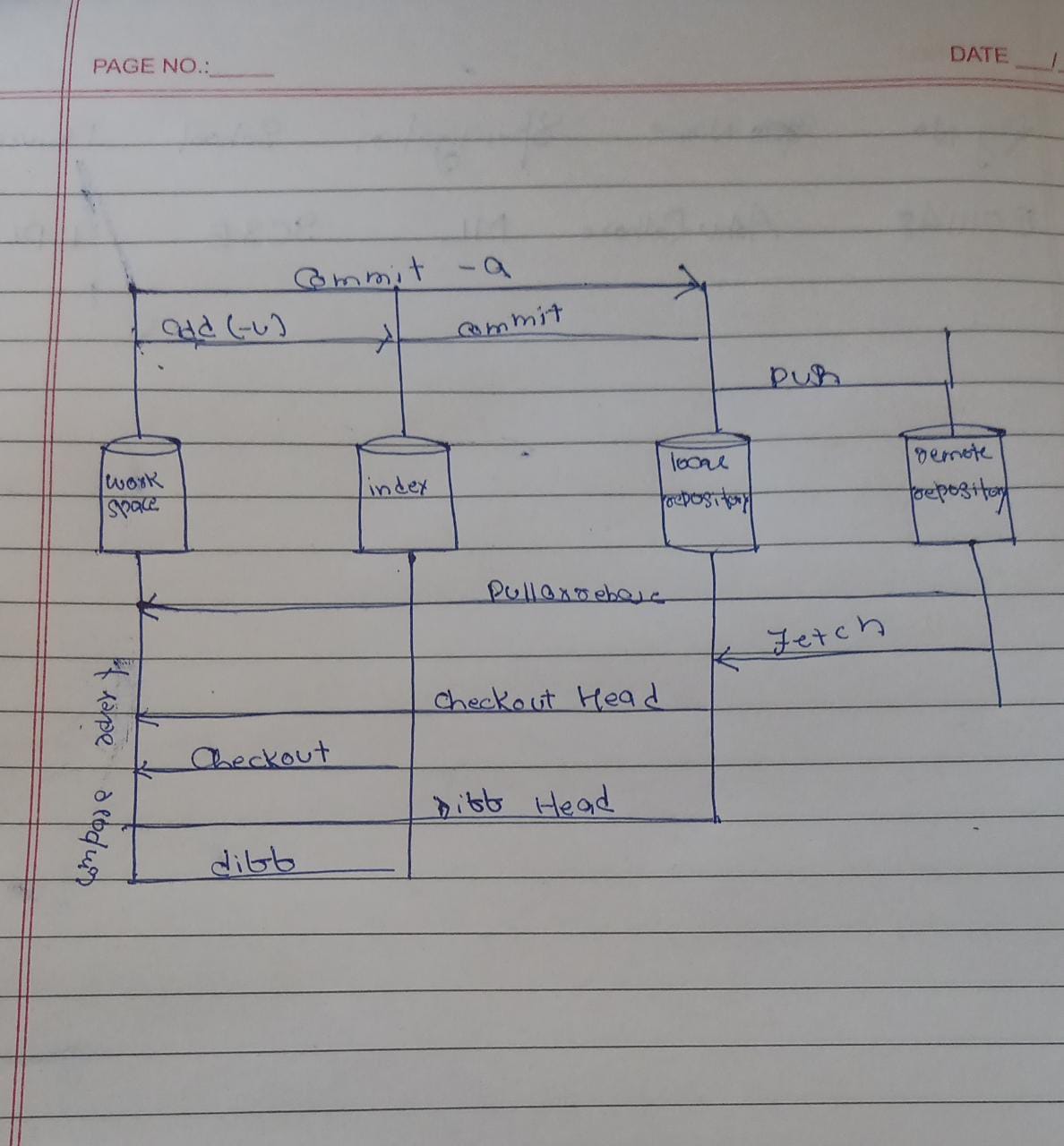
Data that has been committed to storage in your local database is kept there securely.

command- git commit

15-git commit

16- git commot -m”New email".

17-

Github repository or a remote server maintained by your business are examples of remote repositories. just when you initially pull a code and push the modifications once you are finished.

local repository: The code base is built here in the local repository when you clone a remote Git repository or create a new repository. Your commitments will all be made here initially.

index: Staging Area, a location in between your local repository and your working copy of the code. It helps you organise the files you wish to monitor and commit.

workspace: The location where you create, modify, and remove files. Your local computer is where this code is located.

18-An separate line of development is represented by a branch. For the edit stagecommit procedure, branches behave as an abstraction. They may be viewed as a means to ask for a fresh working directory, staging area, and project history.

19- git branch "new-email"

20- git checkout "new-email"

21-Using the "git switch" command, you may switch branches on Git by providing the name of the branch you wish to go to.

You must provide the "-c" option when switching to a branch if it does not already exist. Otherwise, an error message will be displayed.

22-The git init command creates a new Git repository.

A fork makes a whole separate copy of the Git repository.

The target repository is shared by all the developers who have previously contributed to the source repository when a Git repository is cloned. The codebase's prior contributors will keep pushing their modifications and downloading updates from the cloned repository. Anyone who clones a repository has the ability to update their copy of the codebase to reflect any changes made by other developers.

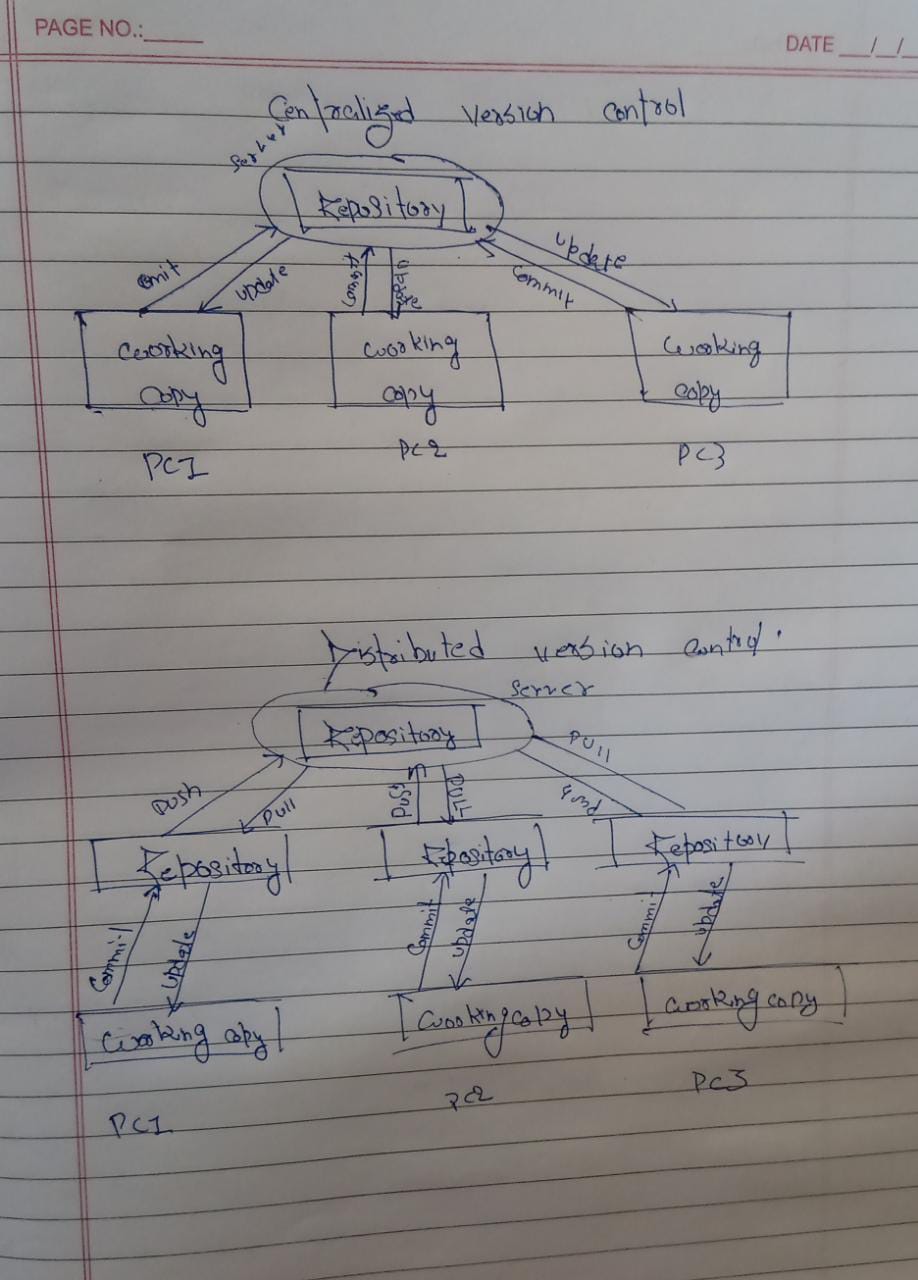
A Git fork procedure will produce a whole new copy of the target repository, as contrast to a clone. The codebase that is forked will be entirely under the control of the person who performed the fork. The cloned Git repository won't be known to developers that contributed to the original source. Unless the developer who executed the fork procedure grants access to them, previous contributors will not have any way to contribute to or synchronise with the Git fork.

We can clone a repository by using git clone <file url>

We can fork a repository by visiting the github then in folder and the right corner have fork option

24-The git push command is used to upload local repository content to a remote repository.

Command for git push— git push <file name>

25- 

Centralized version control architecture -

Centralized version control systems only have one repository worldwide, and each user must commit in order for their changes to be reflected in the repository. By updating, you give others a chance to view your changes.

To make your modifications apparent to others, you must do two things, specifically:

You commit

Theiy updates

Distributed version control architecture-

Multiple repositories are present in distributed version control systems. Each user has a separate working copy and repository. You cannot share your modifications with others by just committing them. This is due to the fact that commit just makes those changes accessible in your local repository; to make them visible in the main repository, you must push them. Similar to this, unless you first pull other people's changes into your repository, you do not receive other people's changes when you update.

There are 4 elements needed to make your modifications apparent to others:

You commit

You push, they pull, and they update