git

VERSION CONTROL SYSTEM

Version Control System

– We will be known forever by the tracks we leave

– Automatic backup

– Sharing on multiple computers

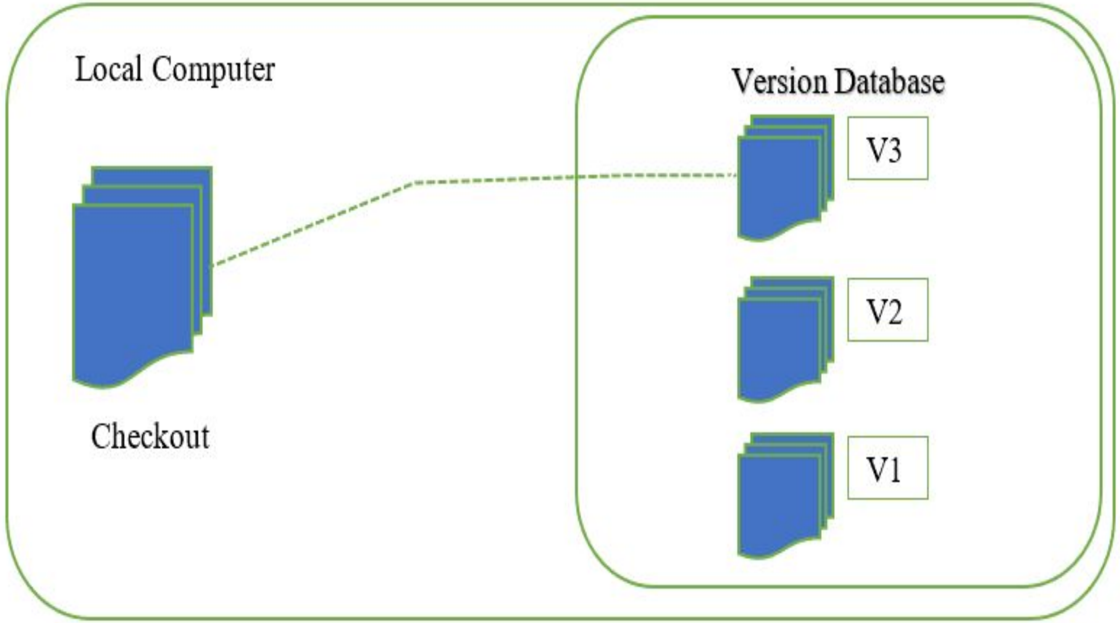
– Version control and branching

– Logging where be changed

TYPES OF VCS

- Local Version Control System
- Centralized Version Control System
- Distributed Version Control System

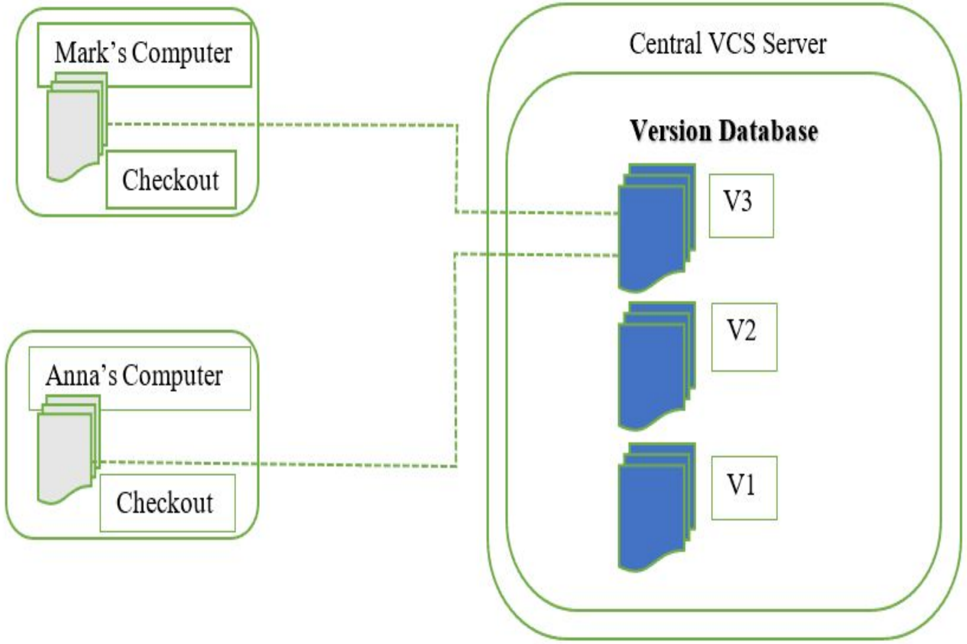
Local Version Control System



A local version control system is a local database located on your local computer, in which every file change is stored as a patch. Every patch set contains only the changes made to the file since its last version. In order to see what the file looked like at any given moment, it is necessary to add up all the relevant patches to the file in order until that given moment.

Centralized Version Control System

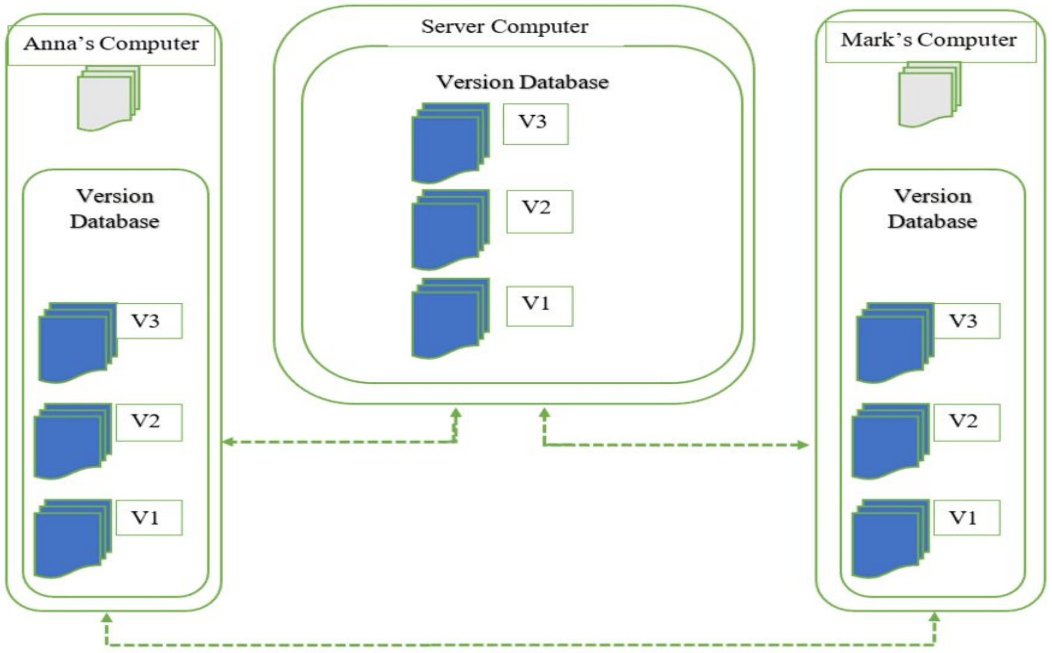
•



A centralized version control system has a single server that contains all the file versions. This enables multiple clients to simultaneously access files on the server, pull them to their local computer or push them onto the server from their local computer. This way, everyone usually knows what everyone else on the project is doing. Administrators have control over who can do what.

Distributed Version Control System

•



Distributed version control systems, clients don't just check out the latest snapshot of the files from the server, they fully mirror the repository, including its full history. Thus, everyone collaborating on a project owns a local copy of the whole project, i.e. owns their own local database with their own complete history. With this model, if the server becomes unavailable or dies, any of the client repositories can send a copy of the project's version to any other client or back onto the server when it becomes available. It is enough that one client contains a correct copy which can then easily be further distributed.

Basic Git workflow

- **Modify** files in your working directory.
- **Stage** files, adding snapshots of them to your staging area.
- **Commit**, which takes the files in the staging area and stores that snapshot permanently to your Git directory.

COMMANDS IN GIT

Set the name and email for Git to use when you commit:

- git config --global user.name "Bugs Bunny"
- git config --global user.email bugs@gmail.com

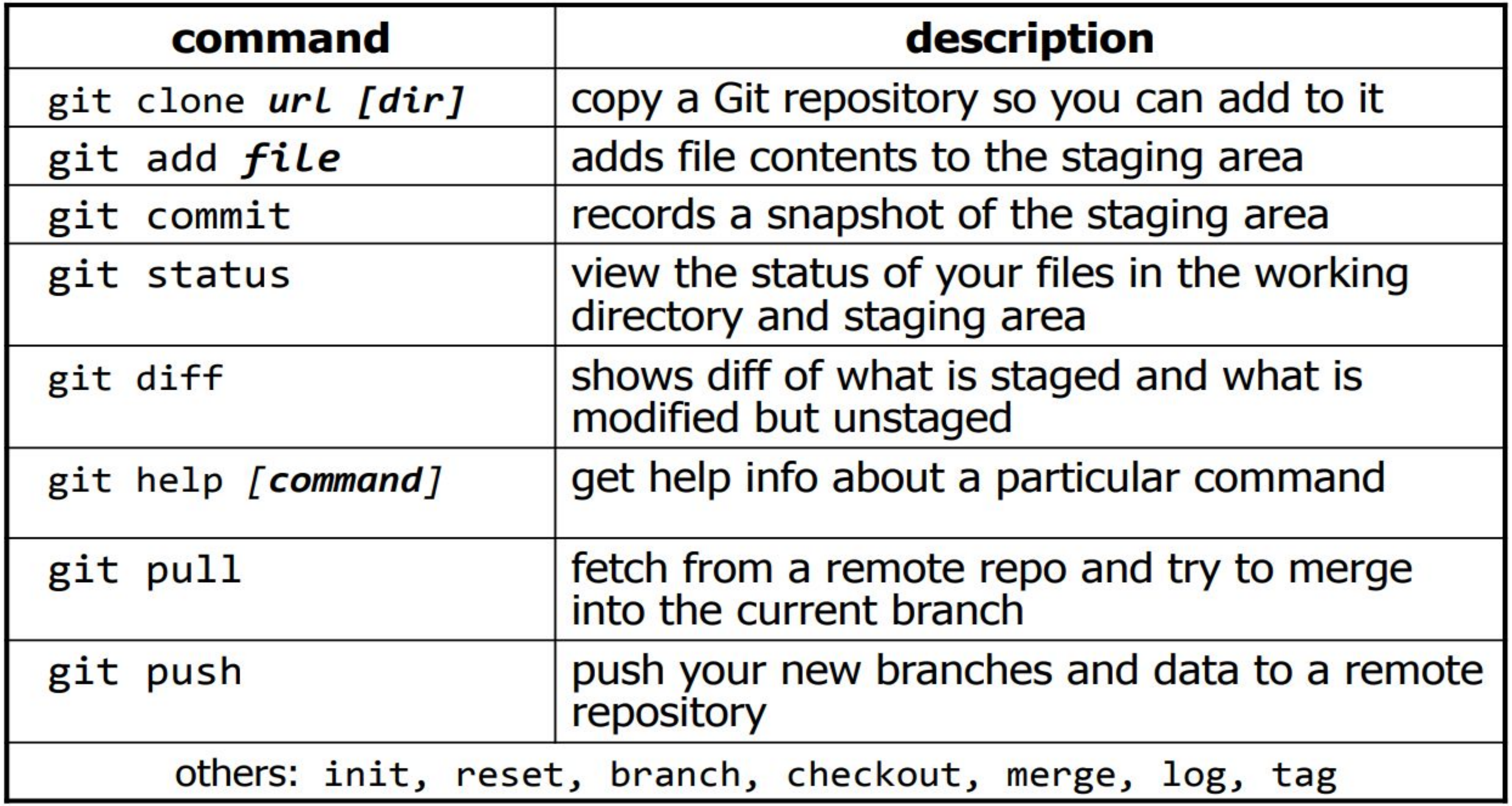
To create a new local Git repo in your current directory:

– git init

• This will create a .git directory in your current directory.

• Then you can commit files in that directory into the repo.

- git add filename
- git commit –m "commit message"



command	description
`git clone` ***url [dir]***	copy a Git repository so you can add to it
`git add` ***file***	adds file contents to the staging area
`git commit`	records a snapshot of the staging area
`git status`	view the status of your files in the working directory and staging area
`git diff`	shows diff of what is staged and what is modified but unstaged
`git help` ***[command]***	get help info about a particular command
`git pull`	fetch from a remote repo and try to merge into the current branch
`git push`	push your new branches and data to a remote repository
others: `init`, `reset`, `branch`, `checkout`, `merge`, `log`, `tag`	

BRANCHING AND MERGING

Git uses branching heavily to switch between multiple tasks.

• To create a new local branch:

– git branch_name

• To list all local branches: (* = current branch)

– git branch

• To switch to a given local branch:

– git checkout branch_name

• To merge changes from a branch into the local master:

– git checkout master

– git merge branch_name

Merge conflicts

• The conflicting file will contain <<< and >>> sections to indicate where Git was unable to resolve a conflict:

```
<<<<<<<< HEAD:index.html
todo: message here
========
thanks for visiting our site
>>>>>>>> SpecialBranch:index.html
```

• Find all such sections, and edit them to the proper state (whichever of the two versions is newer / better / more correct).

GITHUB

- GitHub.com is a site for online storage of Git repositories.

– You can create a remote repo there and push code to it.

– Many open source projects use it, such as the Linux kernel.

– You can get free space for open source projects, or you can pay for private projects.

THE END