

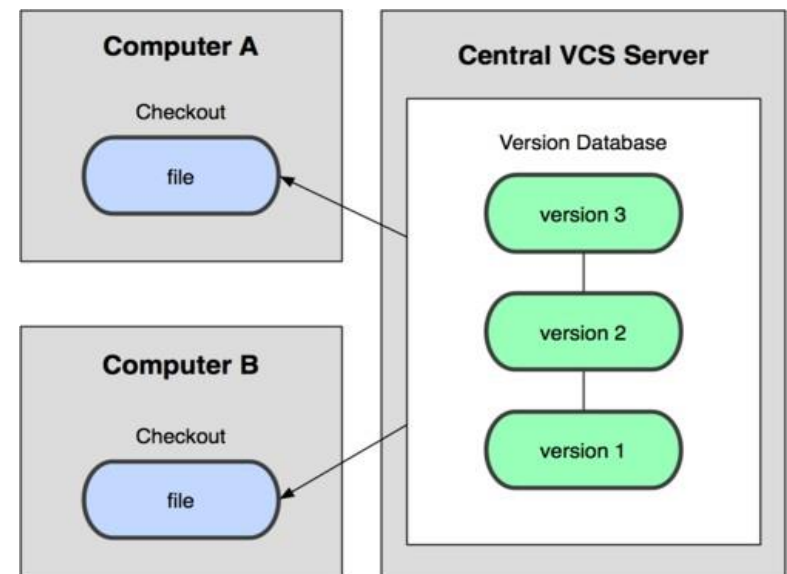
# About Git

- Created by Linus Torvalds, creator of Linux, in 2005
  - Came out of Linux development community
  - Designed to do version control on Linux kernel
- Goals of Git:
  - Speed
  - Support for non-linear development (thousands of parallel branches)
  - Fully distributed
  - Able to handle large projects efficiently
  - *(A "git" is a cranky old man. Linus meant himself.)*



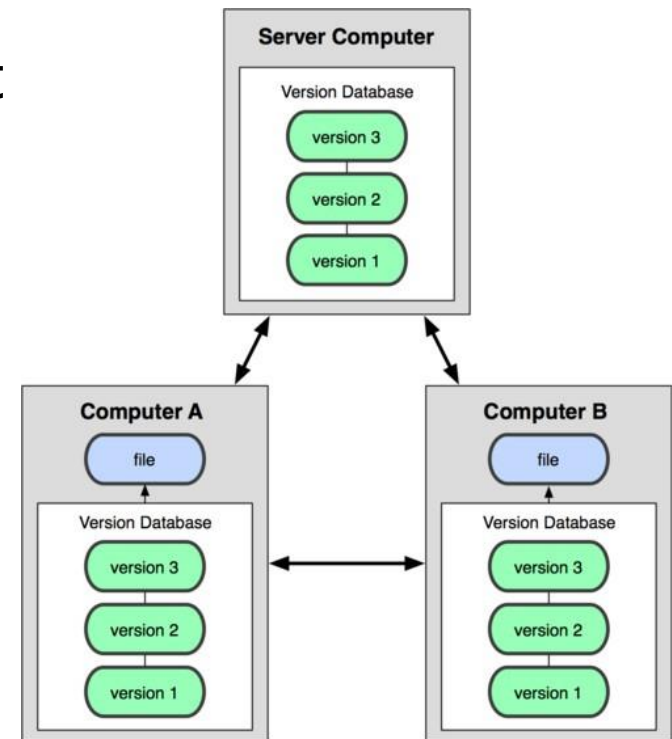
# Centralized VCS

- In Subversion, CVS, Perforce, etc.  
A central server repository (repo) holds the "official copy" of the code
  - the server maintains the sole version history of the repo
- You make "checkouts" of it to your local copy
  - you make local modifications
  - your changes are not versioned
- When you're done, you "check in" back to the server
  - your checkin increments the repo's version



# Distributed VCS (Git)

- In git, mercurial, etc., you don't "checkout" from a central repo
  - you "clone" it and "pull" changes from it
- Your local repo is a complete copy of everything on the remote server
  - yours is "just as good" as theirs
- Many operations are local:
  - check in/out from *local* repo
  - commit changes to *local* repo
  - local repo keeps version history
- When you're ready, you can "push" changes back to server



# Initial Git configuration

- Set the name and email for Git to use when you commit:
  - `git config --global user.name "Akshay Deokar"`
  - `git config --global user.email aks@gmail.com`
  - You can call `git config -list` to verify these are set.
- Set the editor that is used for writing commit messages:
  - `git config --global core.editor nano`
    - (it is vim by default)

# Creating a Git repo

*Two common scenarios: (only do one of these)*

- 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"`

- To **clone a remote repo** to your current directory:

- `git clone url localDirectoryName`

- This will create the given local directory, containing a working copy of the files from the repo, and a `.git` directory (used to hold the staging area and your actual local repo)

# Git commands

command	description
git clone <i>url [dir]</i>	copy a Git repository so you can add to it
git add <i>file</i>	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 [ <i>command</i> ]	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	

# Viewing/undoing changes

- To view status of files in working directory and staging area:
  - `git status` or `git status -s` (short version)
- To see what is modified but unstaged:
  - `git diff`
- To see a list of staged changes:
  - `git diff --cached`
- To see a log of all changes in your local repo:
  - `git log` or `git log --oneline` (shorter version)
    - 1677b2d Edited first line of readme
    - 258efa7 Added line to readme
    - 0e52da7 Initial commit
  - `git log -5` (to show only the 5 most recent updates), etc.

# 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 branchname`
- To merge changes from a branch into the local master:
  - `git checkout master`
  - `git merge branchname`



# Interaction remote repo

- **Push** your local changes to the remote repo.
- **Pull** from remote repo to get most recent changes.
  - (fix conflicts if necessary, add/commit them to your local repo)
- To fetch the most recent updates from the remote repo into your local repo, and put them into your working directory:
  - `git pull origin master`
- To put your changes from your local repo in the remote repo:
  - `git push origin master`

# GitHub

- [GitHub.com](https://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.
- *Question:* Do I always have to use GitHub to use Git?
  - *Answer:* No! You can use Git locally for your own purposes.
  - Or you or someone else could set up a server to share files.
  - Or you could share a repo with users on the same file system, as long everyone has the needed file permissions).