Version Control System

Agenda

Version Control Systems

Git basics

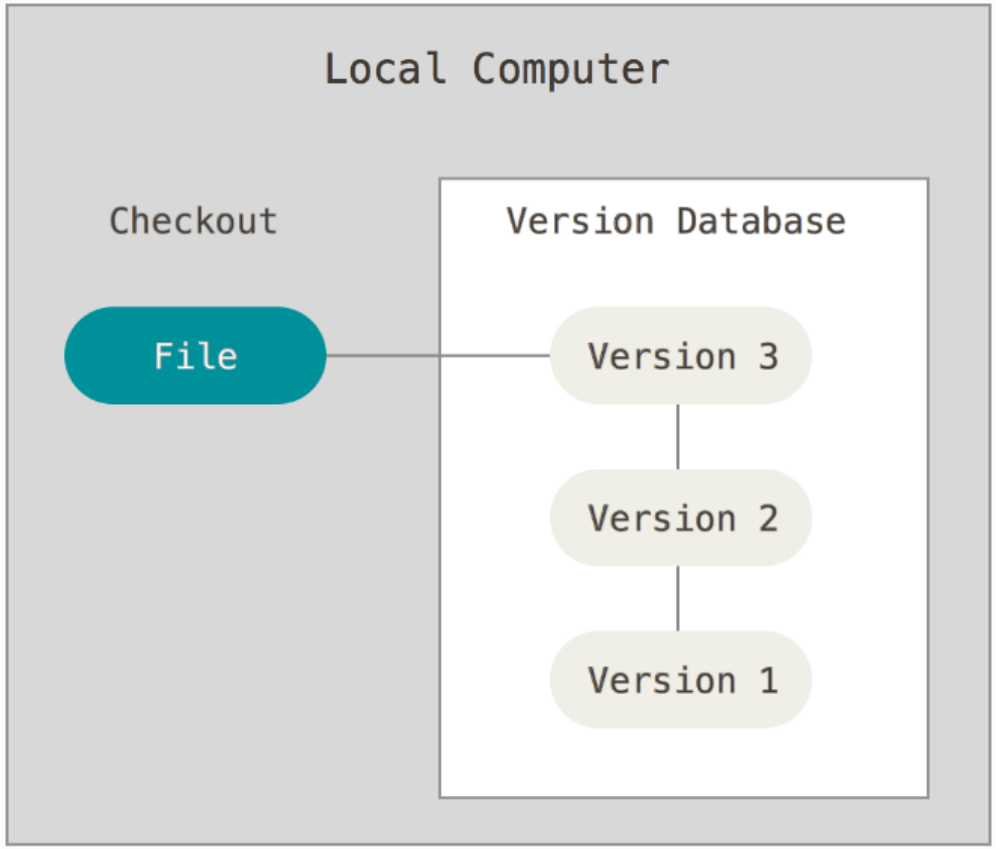
Creating repository

Recording changes

Working with remotes

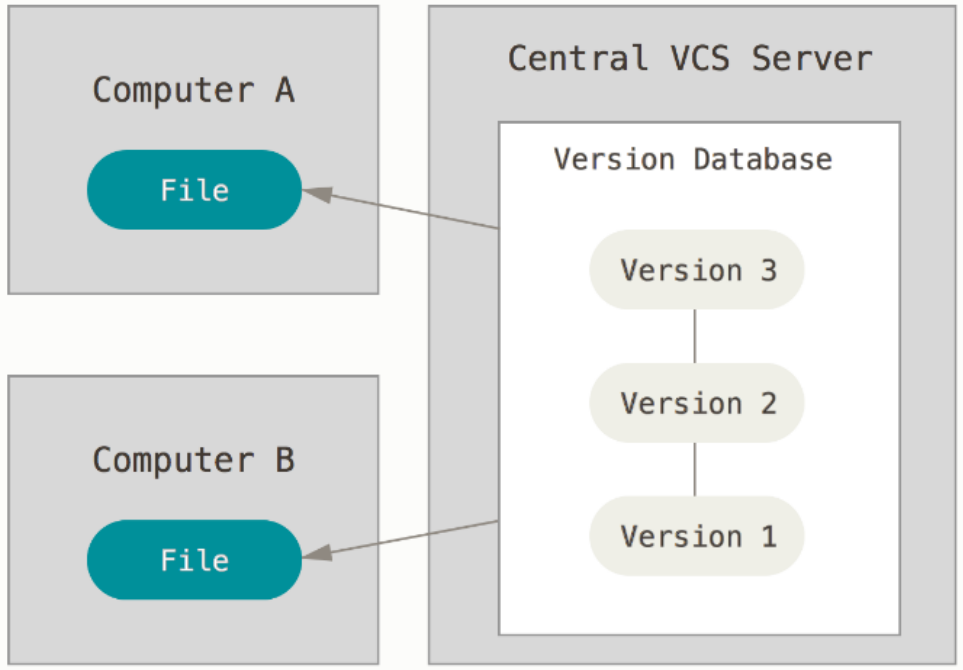
Branching out

Version Control System (VCS)

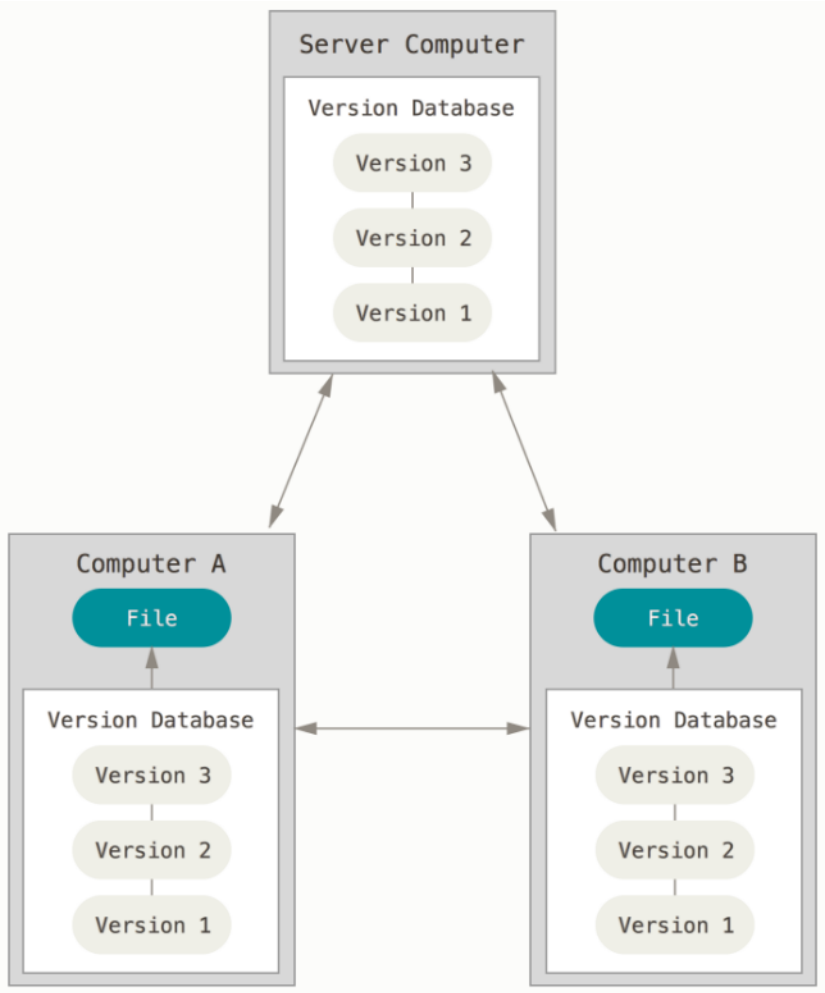


Source: https://git-scm.com/doc

Centralised vs Distributed VCS



Source: https://git-scm.com/doc



Git

Distributed Version Control System

Speed

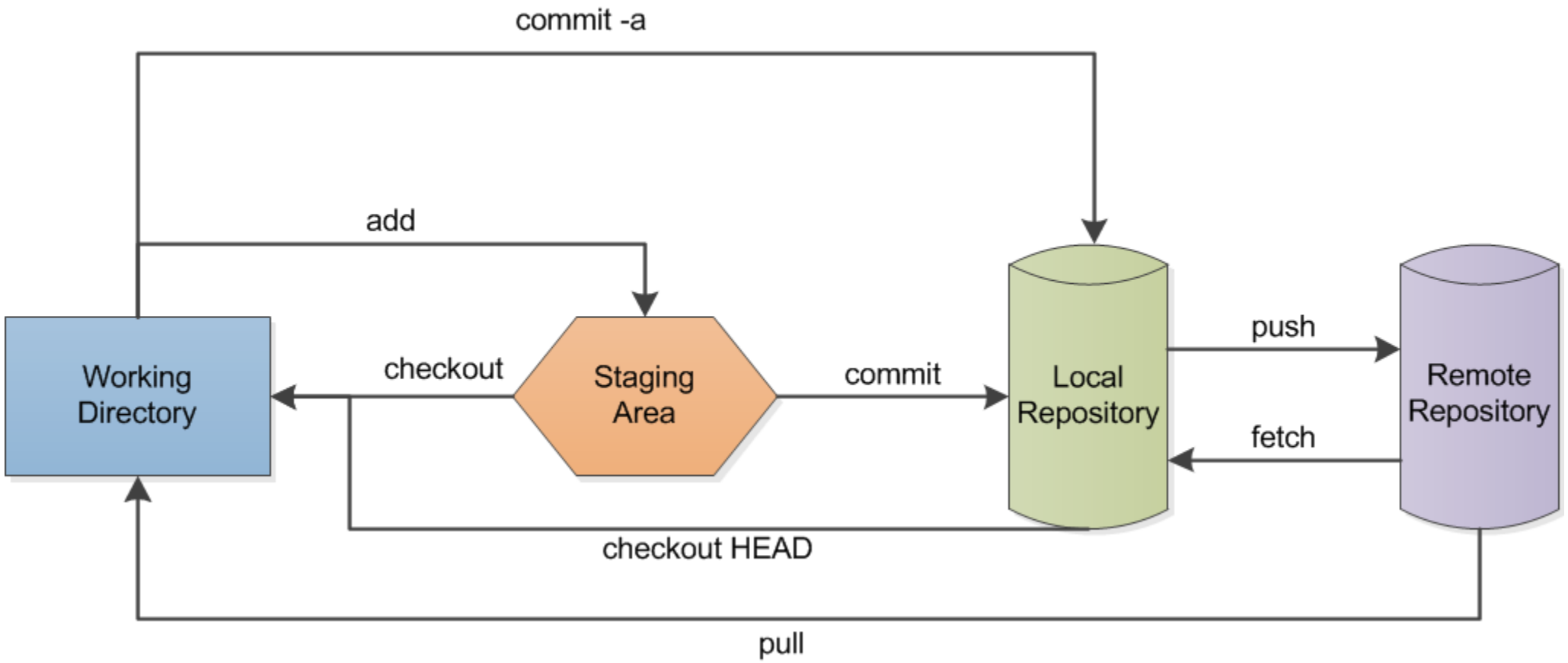
Support for non-linear dev.

Fully distributed

Handle large projects (Linux)

Linus Torvalds ☺

Git data flow



Remote repositories

Git Local Repository

github SOCIAL CODING

Atlassian Bitbucket

GitLab

Tools

Command line

GUI

https://git-scm.com/

Initialising repository

```
cd <folder>     // going to the project folder

git init        // initialising local repository
```

Cloning existing repository

```
git clone <url>      // getting a copy of existing repo

cd <folder>          // going to git repo
```

Setting user identity

```
git config --local user.name ”your name and surname”

git config --local user.email ”your email address”


git config --local --list     // checking repo config
```

Recording changes

```
git status                  // tracking info

git add [files]             // tracking new files

git add [files]             // staging modified files

git commit –m ”message”     // committing changes

git log                     // viewing commit history

git log --oneline           // viewing commit history
```

Working with remotes

```
git remote -v                          // showing remotes

git remote add <name> <url>    // adding remote

git remote show <name>             // inspecting remote


git push [remote-name] [branch-name]

git fetch [remote-name]

git pull [remote-name]
```

Branching out

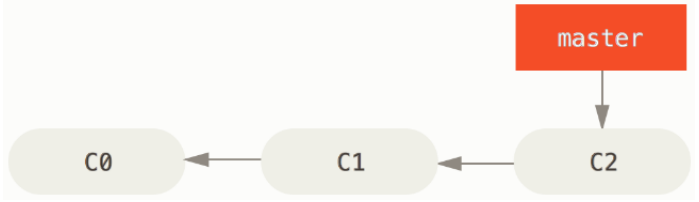
Working on features

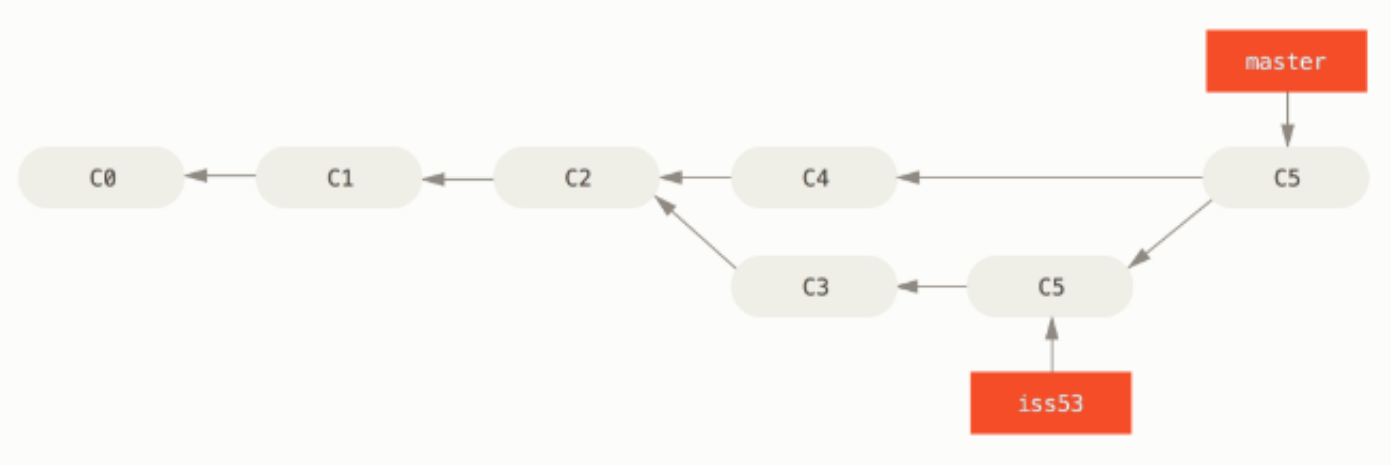
Working on bugs

Trying out new ideas

Creating history of work

Making separate commits



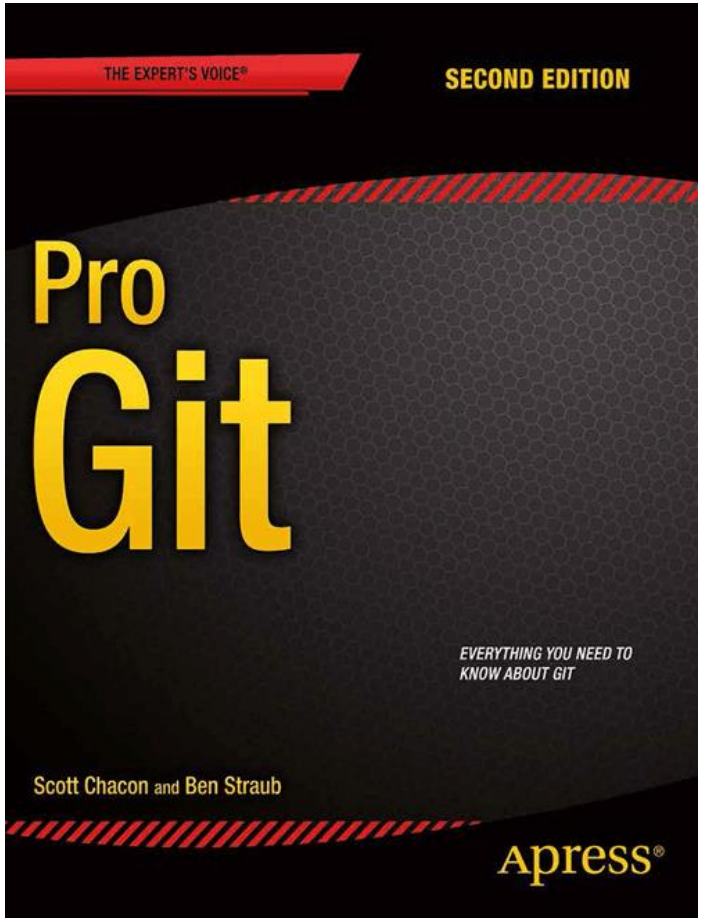


Source: https://git-scm.com/doc

Working on branches

```
git branch                          // listing branches

git branch <branch-name>            // creating new branch

git checkout <branch-name>          // switching branches

git commit …                        // storing branch changes

git merge <branch-name>             // merging branch

git branch –d <branch-name>         // deleting branch

git push …                          // pushing changes
```

Documentation



Book EN - PL translation

CCA Non Commercial Share

PDF, Mobi, ePub

Screencasts

Amazon – printed version

https://git-scm.com/doc

References

Simple guide

http://rogerdudler.github.io/git-guide/

Crash course

https://www.youtube.com/watch?v=SWYqp7iY_Tc

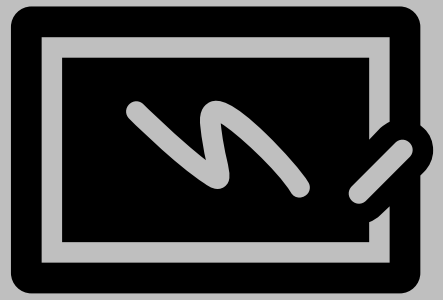
Understanding the Github Flow

https://guides.github.com/introduction/flow/

Become a git guru

https://www.atlassian.com/git/tutorials

To do



Create github account

1. **On the github, create your personal account (if you do not have yet)**

   https://github.com/

Create repository from scratch

1. On the github, create an empty MyFirstProject repository. Familiarise yourself with the Quick setup info (MyFirstProject repository code tab).
2. On the computer desktop, create MyFirstProject folder with three empty files: a.txt, b.txt, and c.txt. Initialise a git repository in the MyFirstProject folder. Then for the repository, set user identity.
3. Start tracking files, and then show repository status.
4. In MyFirstProject, add an empty file author.txt and start tracking the file. Show repository status.
5. In the first line of the file, add your name and surname, and then show repository status.
6. Add files to the staging area. Then show repository status.
7. Commit changes . Use ‘My first commit’ message. Then show commit logs. Show repository status.
8. Add remote and push changes from local to remote repository (github).
9. On the github, check out the MyFirstProject repository contents (should be the same as local repo).
10. On the computer desktop, remove the MyFirstProject folder.

Clone existing remote repository

1. On the computer desktop, clone the MyFirstProject remote repository into a local folder.
2. Complete author.txt with your email.
3. Commit changes. Use ‘Added email address’ message.
4. Push changes from local to remote repository (github).
5. On the github, check out the MyFirstProject repository (author.txt content).
6. On the computer desktop, delete MyFirstProject folder.

Use git GUI

Do the ‘Create repository from scratch’ assignment. Instead of command line tool, use git Graphical User Interface (git GUI).

Create and merge branches

1. Clone MyFirstProject.
2. Display branch list. What is the name of the default branch?
3. Add a new branch ‘interests’. Then display branch list (please note that only one branch is active).
4. Switch to ‘interests’ branch.
5. Add new file myinterests.txt. Type two-three sentences about your hobby.
6. Commit changes.
7. Switch between ‘master’ and ‘interests’ branches. Can you see the differences in the working directory?
8. Merge ‘interests’ branch into ‘master’.
9. Check the differences in the working directory.
10. Remove ‘interests’ branch.
11. Push changes from local to remote repository (github) and check the remote repo content.

Collaborate on project

1. **On your local computer, create a new repository Books.**
2. **In the repo, add a new file books.txt with your three favourite book titles, on separate lines.**
3. **Commit changes and push the repo to the github. Make sure to create an empty remote repo before.**
4. **On the github, add your colleague as the repository collaborator.**
5. **Ask your collaborator to**
   - clone the repository
   - add bookstores.txt with two names and URLs of the Internet bookstores
   - commit changes and push them to the github
6. **On your local computer, add a new file itbooks.txt with three titles of information technology books.**
7. **Commit changes and try to push the repo to the github. Why is that not possible?**
8. **Fetch the repo from the github, merge it with your local one, and then push the repo to the github.**
9. **Check the final repo content on the github.**

Resolve conflicts

1. **Create a completely new git repository with at least three files, one of them should be authors.txt.**
2. **Add your colleague as a collaborator.**
3. **With your collaborator, complete the authors.txt with your name and surname in the same time. Then you and your collaborator try pushing changes on the github.**
   - How to compare and display differences between the two versions of authors.txt?
   - How to resolve the conflict?
   - Try to find a solution in the Pro Git book, git manual or on the Internet.

Undo changes

1. Create a folder with three files. You can copy and paste any existing files.
2. Initialise a local git repository in the created folder.
3. Start tracking the files, and then show repository status.
4. Commit changes . Use ‘Repo with 3 files’ message. Then show commit logs.
5. Add to the folder a file.
6. Start tracking this file and commit changes . Use ‘Repo with 4 files’ message. Then show commit logs.
7. Add to the folder a file.
8. Start tracking this file and commit changes . Use ‘Repo with 5 files’ message. Then show commit logs.
9. Roll back last two changes (adding two files).

Git-it tutorial

1. **Familiarise yourself with the Git-it tutorial**
   - https://try.github.io/
2. **Download and unzip the Git-it app**
   - You can find the app in the section: What to Install | Git-it (releases)
3. **Move on to the Get started section and face all the challenges contained in the Git-it app**