Git & GitHub

Academia de Software Rumos 2024

Content

What is Git & GitHub

Basic Functions

Let’s see some examples

Introduction

- Had code that worked, made a bunch of changes and saved it, which broke the code, and now you just want the working version back…

Introduction

- Had code that worked, made a bunch of changes and saved it, which broke the code, and now you just want the working version back…
- Accidentally deleted a critical file, hundreds of lines of code gone…

Introduction

- Had code that worked, made a bunch of changes and saved it, which broke the code, and now you just want the working version back…
- Accidentally deleted a critical file, hundreds of lines of code gone…
- Somehow messed up the structure/contents of your code base, and want to just “undo” the crazy action you just did!

Introduction

- Hard drive crash!!!! Everything’s gone, the day before deadline!

Introduction

- Hard drive crash!!!! Everything’s gone, the day before deadline!
- Whose computer stores the "official" copy of the project?

Introduction

- Hard drive crash!!!! Everything’s gone, the day before deadline!
- Whose computer stores the "official" copy of the project?
- Will we be able to read/write each other's changes?

Introduction

- Hard drive crash!!!! Everything’s gone, the day before deadline!
- Whose computer stores the "official" copy of the project?
- Will we be able to read/write each other's changes?
- Let’s just email changed files back and forth! Yay!

Introduction

- Hard drive crash!!!! Everything’s gone, the day before deadline!
- Whose computer stores the "official" copy of the project?
- Will we be able to read/write each other's changes?
- Let’s just email changed files back and forth! Yay!
- What happens if we both try to edit the same file?

Introduction

- What happens if we make a mistake and corrupt an important file?

Introduction

- What happens if we make a mistake and corrupt an important file?
- How do I know what code each teammate is working on?

Introduction

- What happens if we make a mistake and corrupt an important file?
- How do I know what code each teammate is working on?
- How do we know who added this code piece?

Introduction

- What happens if we make a mistake and corrupt an important file?
- How do I know what code each teammate is working on?
- How do we know who added this code piece?

Version Control System

Version control System is all about managing multiple versions of documents, programs, web sites, etc.

- Almost all “real” projects use some kind of version control
- Essential for team projects, but also very useful for individual projects

Version Control System

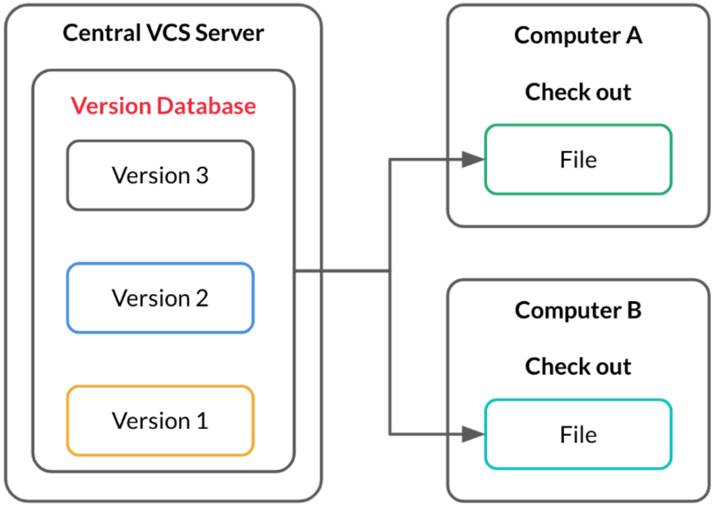
Some well-known version control systems are CVS, Subversion, Mercurial, and Git

- CVS and Subversion use a “central” repository; users “check out” files, work on them, and “check them in”
- Mercurial and Git treat all repositories as equal

Git

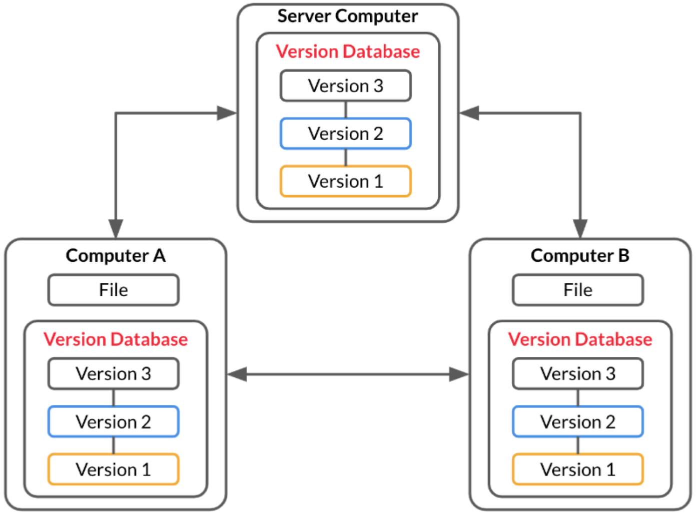
Why Git?

Central Version Control System (CVCS)



Why Git?

Distributed Version Control System (DVCS)

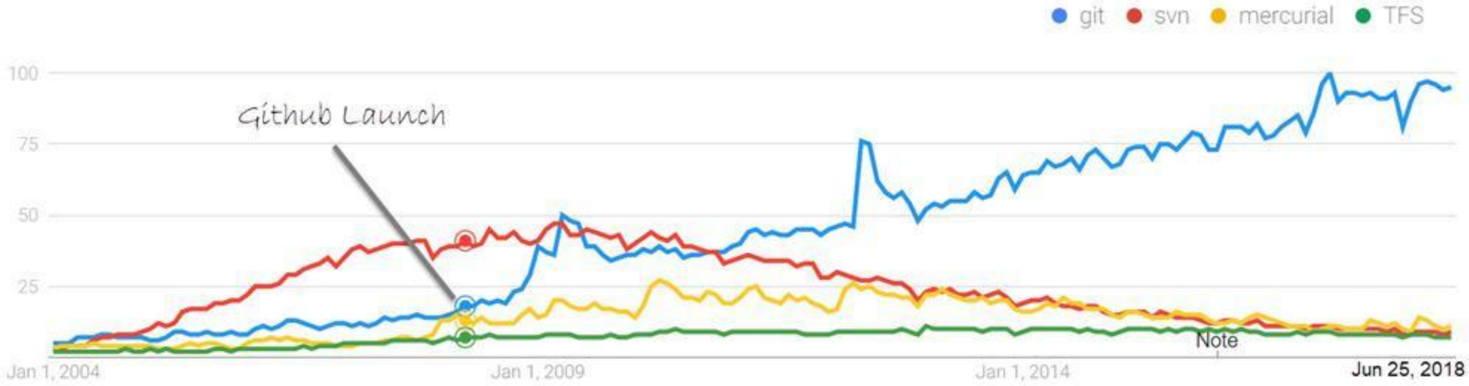


Why Git?

As shown in the following diagram, the entire database of different versions is present on the local machines of developers as well as the central server. Even if the central server crashes, this database is available on all the local machines as well.

Why Git?

Git came into existence because of a controversy between Linux developers and BitKeeper. It was developed by Linus Torvald who was the main developer of the Linux kernel, used by Linux distributions.



What is Git?

- Git is a widely used Version Control System (VCS) that lets you keep track of all the modifications you make to your code.
- This means that if a new feature is causing any errors, you can easily roll back to a previous version. https://git-scm.com/

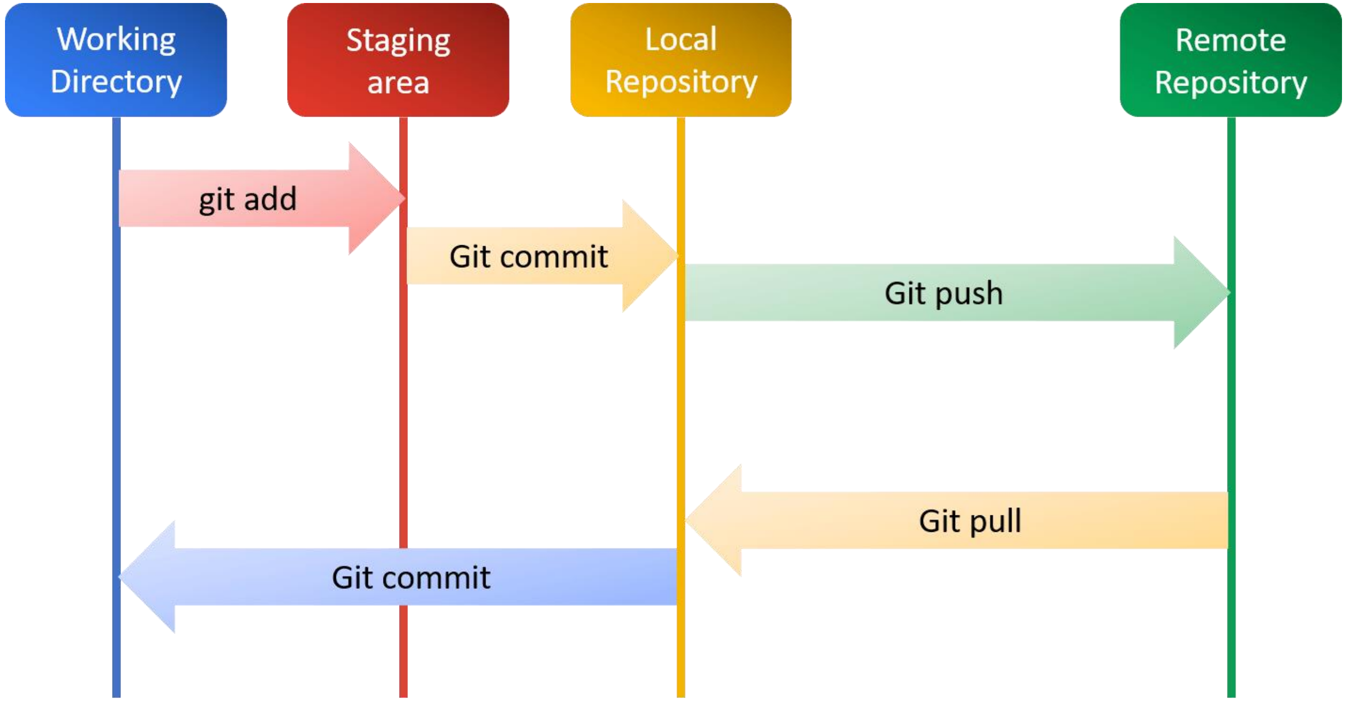
What is Git?

- Since it is written in the C language, speed and performance are ingrained in Git right from its inception.
- Besides this, Git also provides a lot of buffers before actually saving any changes to the project.

What is GitHub?

- GitHub is a widely used platform for version control that uses Git at its core. It lets you host the remote version of your project from where all the collaborators can have access to it.
- https://github.com/

Process flow



Git comands

# Clone Repository

- git clone https://github.com/????/??????.git

Git comands

# Git Status

- git status

Git comands

# Create branch

- git branch test1

Git comands

# List branches

- git branch --list

Git comands

# Checkout branch

- git checkout test1

Git comands

# Pushing to Remote

- git add <file>
- git add –A
- git commit -m "commit message"
- git push -u origin main

Git comands

# Pulling from Remote

- git pull
- git pull <remote>

Git comands

# Revert

- git log --oneline
- git revert e33bb1c

Git comands

# Merge

- git checkout dev
- git fetch
- git merge <branch-name>

Git comands

https://www.freecodecamp.org/news/10-important-git-commands-that-every-developer-should-know/

Contactos

- Sérgio Fontes:
  - sergio.c.fontes@gmail.com
  - https://www.linkedin.com/in/sergiofontes/