STU SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

Introduction to Git

Dr.Sc. Oleksii Yehorchenkov

Department of Spatial Planning

Funded by the European Union NextGenerationEU

What is version control?

> Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.

https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control

What is version control?

It allows you to:

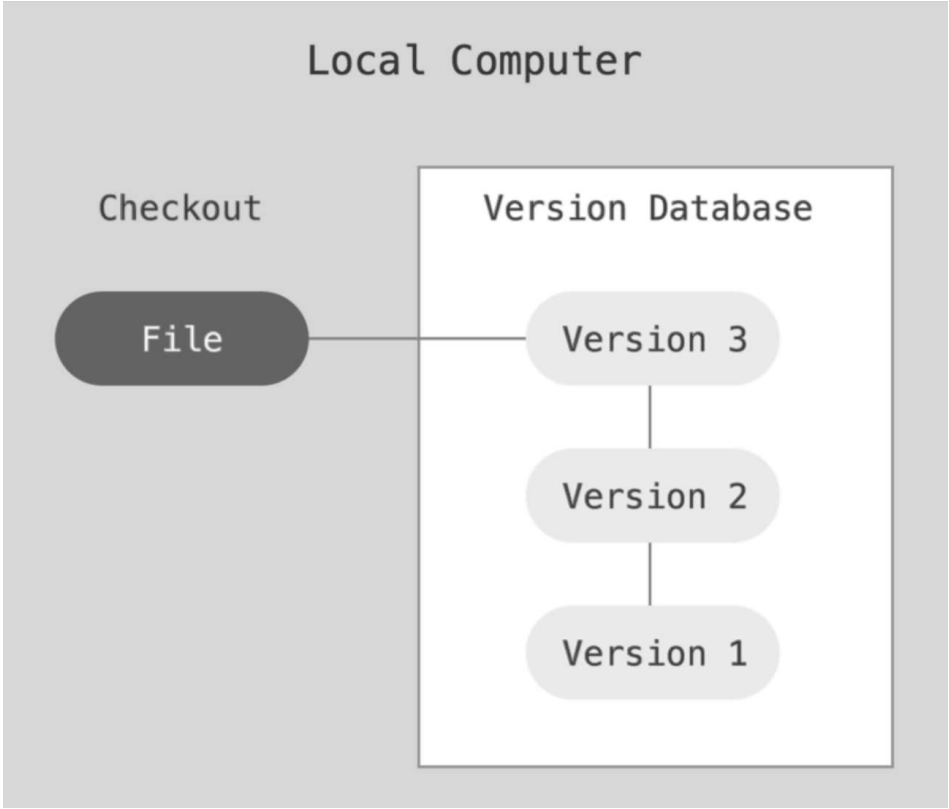
- revert selected files back to a previous state,
- revert the entire project back to a previous state,
- compare changes over time,
- see who last modified something that might be causing a problem,
- who introduced an issue and when, and more.

Using a VCS also generally means that if you screw things up or lose files, you can easily recover. In addition, you get all this for very little overhead.

Local Version Control Systems

Many people's version-control method of choice is to copy files into another directory (perhaps a time-stamped directory, if they're clever). This approach is very common because it is so simple, but it is also incredibly error prone. It is easy to forget which directory you're in and accidentally write to the wrong file or copy over files you don't mean to.

To deal with this issue, programmers long ago developed local VCSs that had a simple database that kept all the changes to files under revision control.

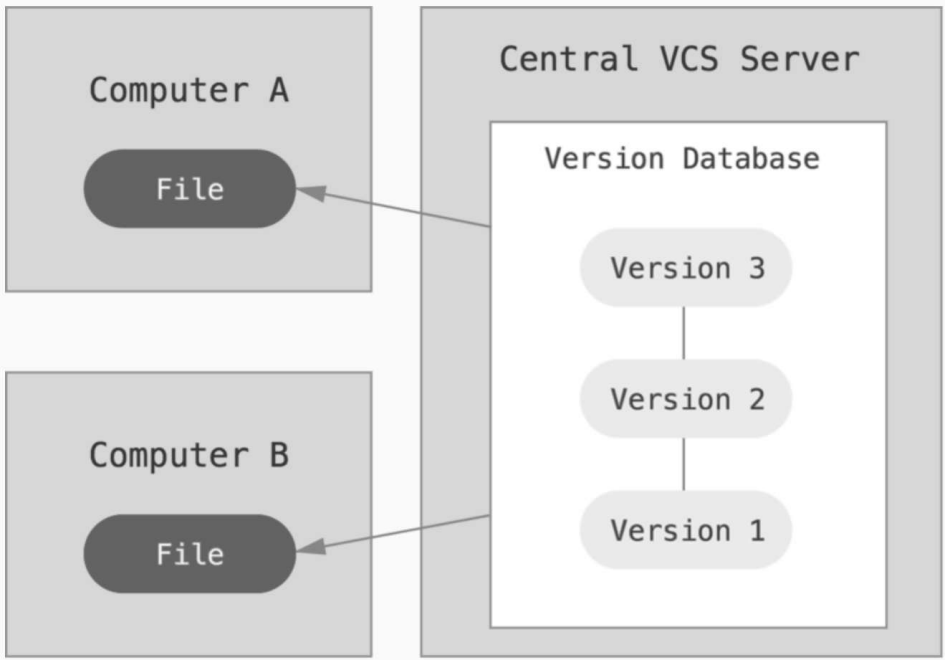


Centralized Version Control Systems

Clients have a single central copy of a project on a server.

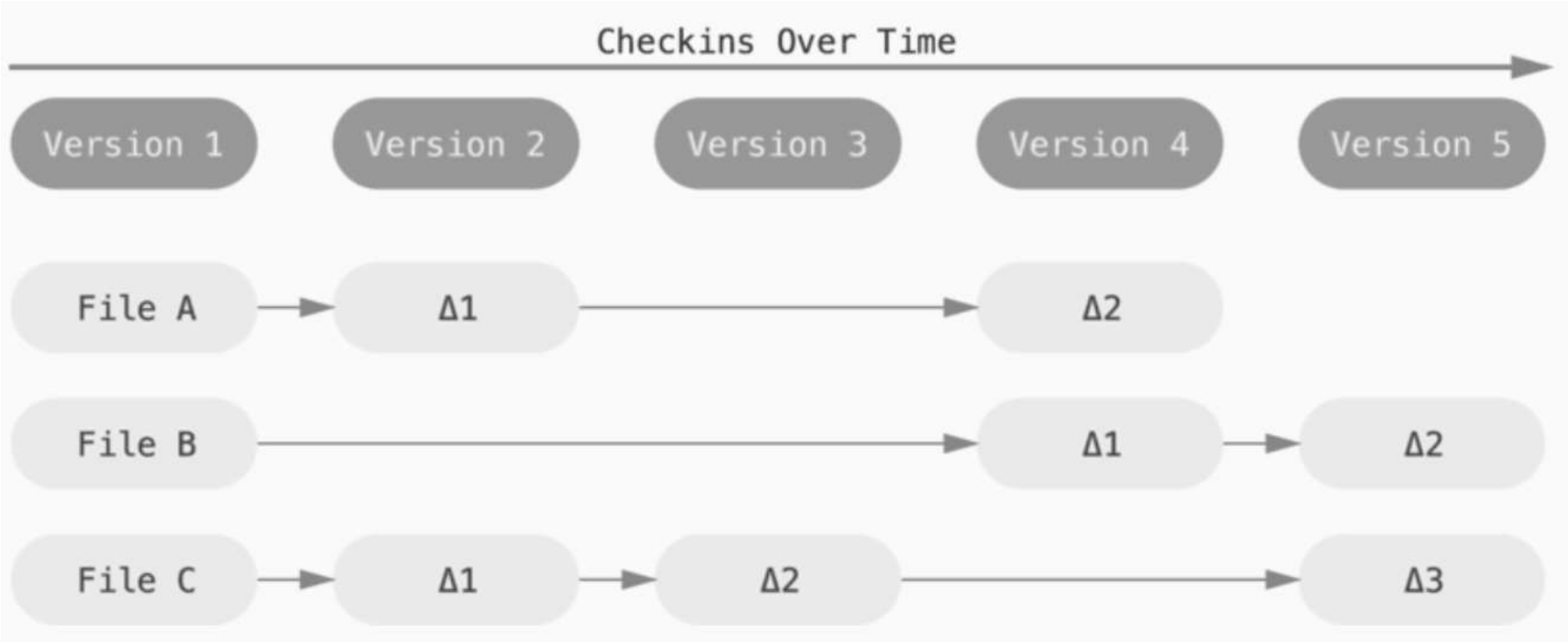
The problem is if the hard disk the central database is on becomes corrupted, and proper backups haven’t been kept, programmer loses absolutely everything — the entire history of the project except whatever single snapshots people happen to have on their local machines.

Examples of such system are SVN (Subversion), CVS, Perforce



Centralized Version Control Systems

Delta-based version control

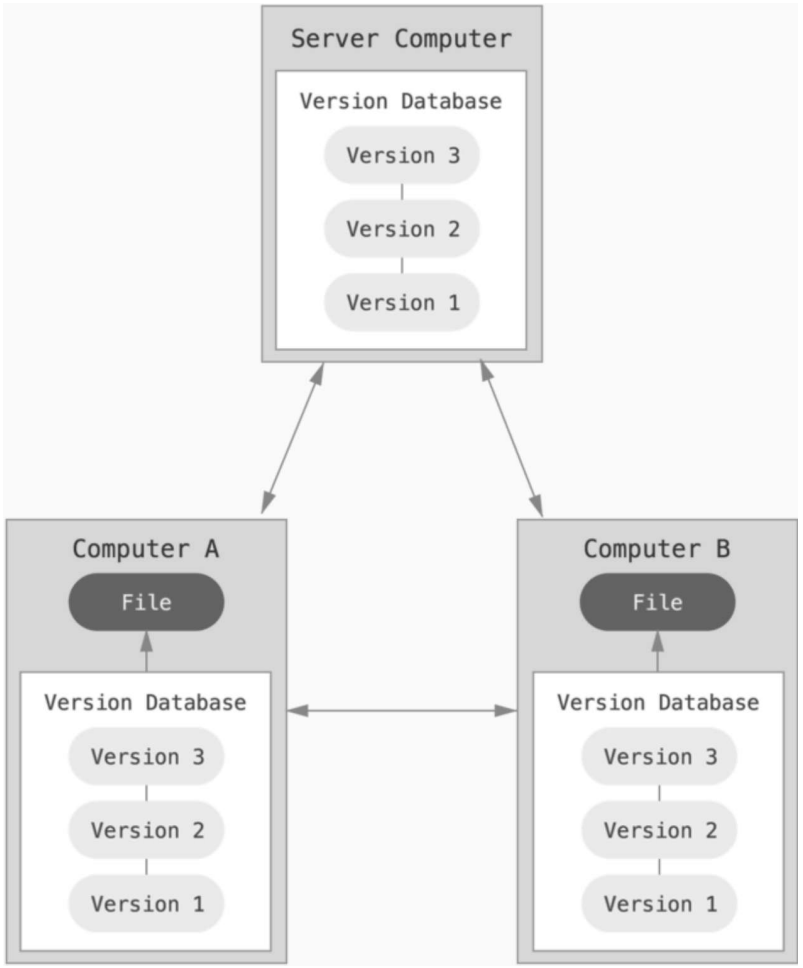


Distributed Version Control Systems

With distributed version control systems (DVCS), you don't rely on a central server to store all the versions of a project's files. Instead, **you clone a copy of a repository locally** so you have the full history of the project.

Thus, if any server dies, and these systems were collaborating via that server, any of the client repositories can be copied back up to the server to restore it. In DVCS **most operations are executed locally**!

Two common distributed version control systems are Git and Mercurial.

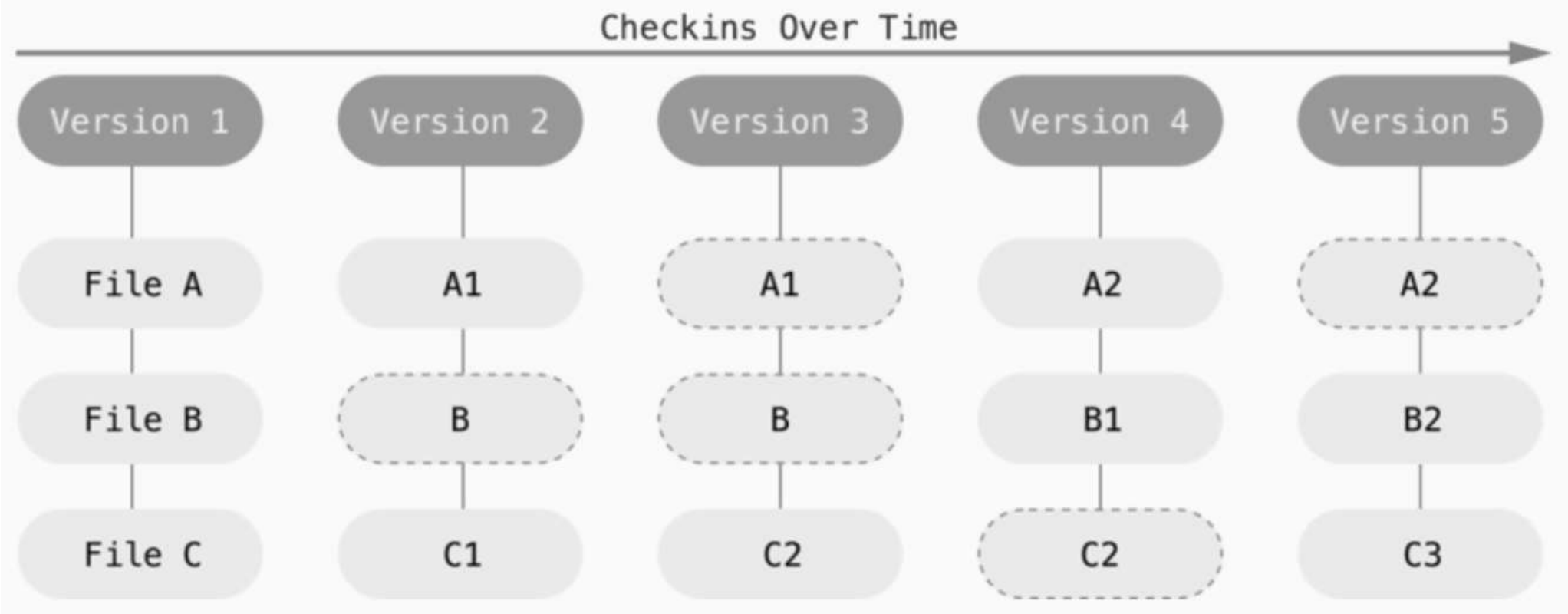


What is Git?

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency

What is Git?

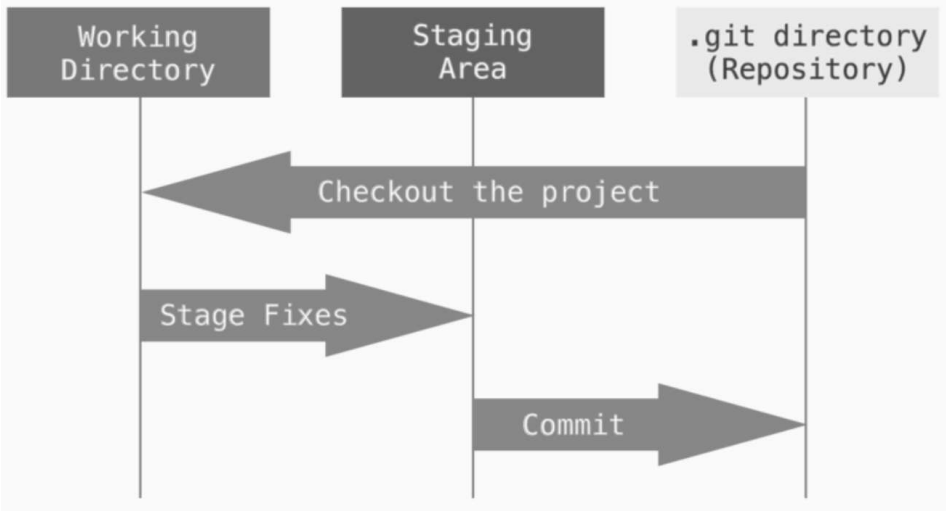
Git thinks about data more like stream of snapshots



Why Git?

- Speed;
- Simple design;
- Fully distributed;
- Strong support for non-linear development (thousands of parallel branches);
- Able to handle large projects like the Linux kernel efficiently (speed and data size).

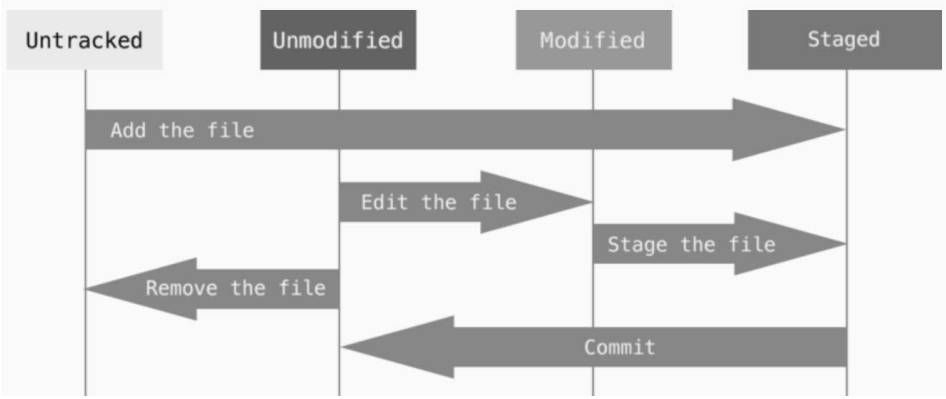
Recording changes to the git repository



- Create new file or edit existing file or files in the Working Directory
- Stage changes (add to index)
- Commit (only added files from staged area)

You can’t commit changes that are not staged

Recording changes to the git repository



- Each file in your working directory can be in one of two states: **tracked** or **untracked**;
- Tracked files can be **unmodified**, **modified**.

What is GitHub?

> GitHub, Inc. is a platform and cloud-based service for software development and version control using Git, allowing developers to store and manage their code. It provides the distributed version control of Git plus access control, bug tracking, software feature requests, task management, continuous integration, and wikis for every project.

https://en.wikipedia.org/wiki/GitHub

What is GitHub?

- Allows users to “push” and “pull” their local repositories to and from remote repositories on the web;
- Provides users a user-friendly interface for managing their repositories;
- Users’ repositories are backed up on the GitHub server in case somethings happens to the local copies;
- Social aspect allows users to follow one another and share projects;
- Users can sign up and host a public code repository for free, which makes GitHub especially popular with open-source projects.
- User-friendly interface allows users manage not only software projects but many other projects like writing books.

As of January 2023, GitHub reported having over 100 million developers and more than 372 million repositories, including at least 28 million public repositories. It is the world’s largest source code host as of June 2023.