

Version Control System

April 21, 2023

What is Version Control System? (1/1)

Version control, also known as **source control**, is the practice of tracking and managing changes to software code.

- Version control systems are software tools that help software teams manage changes to source code over time.
- The distributed version control tool is used for source code management.
- It allows multiple developers to work together.
- It supports non-linear development through its thousands of parallel branches.

What is Git (1/1)

- Git is a version control system used for tracking changes in computer files.
- It is generally used for source code management in software development.
- A **repository** contains all of your project's files and each file's revision history. You can discuss and manage your project's work within the repository.
- Git is used to tracking changes in the source code

Use of Merkel tree (1/3)

- A version control system like Git is used to keep track of changes made to files over time.
- When you create a new version of a file, Git makes a copy of the file and stores it as a snapshot, along with a description of the changes you made.

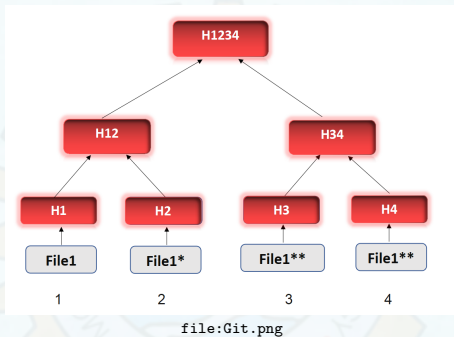
Git uses a data structure called a **Merkle tree** to efficiently keep track of these snapshots.

- A Merkle tree is a tree structure where each leaf node represents a piece of data, in this case, a file snapshot.
- The parent node of each pair of leaf nodes represents the hash of those two leaf nodes concatenated together.

Use of Merkel tree (2/3)

- This process continues recursively up the tree until there is only one root node, which represents the hash of the entire tree.
- Overall, the Merkle tree is a powerful data structure that helps Git efficiently manage changes to files over time.
- It allows Git to store a complete history of changes to files while still keeping storage requirements to a minimum.

Use of Merkel tree (3/3)



Working

- When you make changes to a file, Git creates a new snapshot of the file and calculates its hash.
- Git then adds this new hash as a new leaf node to the Merkle tree, along with a reference to its parent node.
- By using this Merkle tree, Git can efficiently compare versions of files to determine what has changed.
- It does this by comparing the hash of the new version of the file with the hash of the previous version.
- If the hashes are the same, then the contents of the file have not changed.
- If the hashes are different, the contents of the file have changed, and it can use the Merkle tree to efficiently determine which parts of the file have changed.

Benefit of Git (1/4)

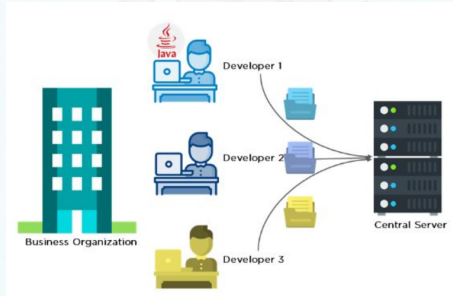


Figure: When there is no VCS like Git

Benefit of Git (2/4)

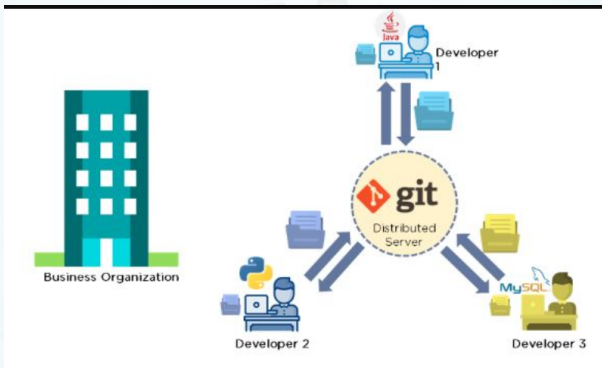


Figure: Scenario after Git

Benefit of Git (3/4)

Now let's look at the scenario after Git:

- Every developer has an entire copy of the code on their local systems
- Any changes made to the source code can be tracked by others
- There is regular communication between the developers