**Version Control System**

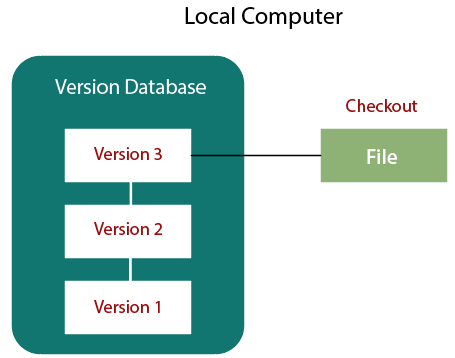
A version control system is a software that tracks changes to a file or set of files over time so that you can review and use specific versions later. It also allows you to work together with other programmers. The version control system is a collection of software tools that help a team to manage changes in a source code. It is very helpful in software development. It also allows software teams to preserve efficiency and agility. It gives the benefits of simultaneously working, branching and merging and traceability.

* **A repository:** It can be thought as a database of changes. It contains all the edits and previous and current versions of the project.
* **Copy of Work (called as checkout):** It is the personal copy of all the files in a project. You can edit to this copy, without affecting the work of others and you can finally commit your changes to a repository when you are done making your changes.

Types of Version Control System

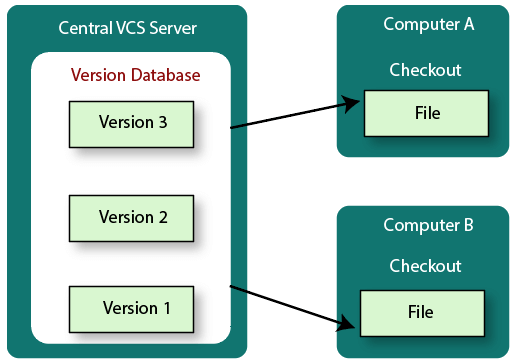
* Localized version Control System
* Centralized version control systems
* Distributed version control systems

### Localized Version Control Systems



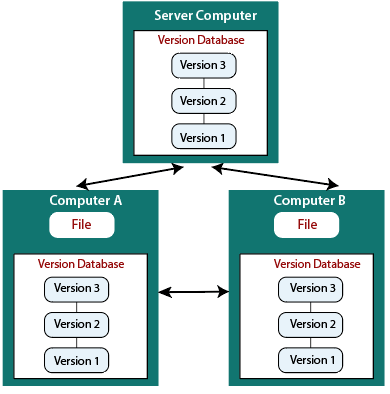
* This VCS is error prone.
* To reduce errors, programmers developed simple database which keeps all changes to the file under revision control.
* It keeps local copies of the file.
* It doesn’t allow developers to collaborate with each other.

1. Centralized Version Control System



* It has a single server that contains versioned files and some clients to checkout from a central place.
* Everyone on the system has information about the work what others are doing on the project.
* Administrators have control over other developers.
* Centralized Version Control System uses a central server to store all the database and team collaboration.
* Due to single point failure, which means the failure of the central server, developers do not prefer it.

1. Distributed Version Control System



* The user has a local copy of a repository. So, the clients don't just check out the latest snapshot of the files even they can fully mirror the repository.
* The local repository contains all the files and metadata present in the main repository.
* It allows automatic management branching and merging
* It enhances the ability to work offline and does not rely on a single location for backups.
* If any server stops and other systems were collaborating via it, then any of the client repositories could be restored by that server.
* Every checkout is a full backup of all the data.

GIT:

* Git is a [free and open source](https://git-scm.com/about/free-and-open-source) distributed version control system designed to handle everything from small to very large projects with speed and efficiency.
* A Git repository contains the history of a collection of files starting from a certain directory.
* The process of copying an existing Git repository via the Git tooling is called cloning. After cloning a repository the user has the complete repository with its history on his local machine. Git also supports the creation of new repositories targeting the usage on a server. If you clone a Git repository, by default, Git assumes that you want to work in this repository as a user.
* A local repository provides at least one collection of files which originate from a certain version of the repository. This collection of files is called the working tree.
* A file in the working tree of a Git repository can have different states. These states are the following:

1. untracked: the file is not tracked by the Git repository. This means that the file never staged nor committed.
2. tracked: committed and not staged
3. staged: staged to be included in the next commit
4. dirty / modified: the file has changed but the change is not staged