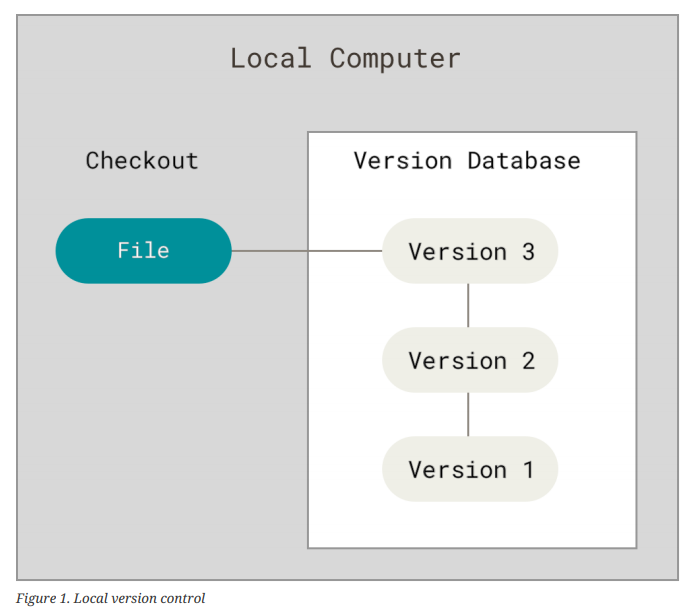
Git Architecture

# Getting Started

## About 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.

## Local Version Control Systems

RCS or Revision Control System

## Centralized Version Control Systems

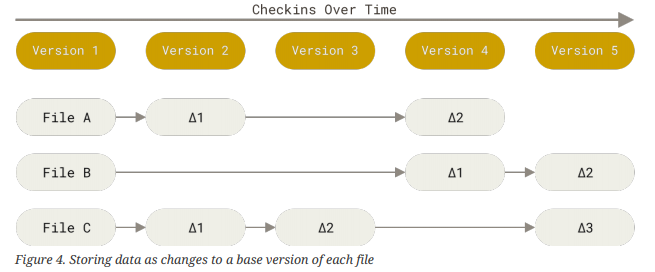
However, this setup also has some serious downsides.

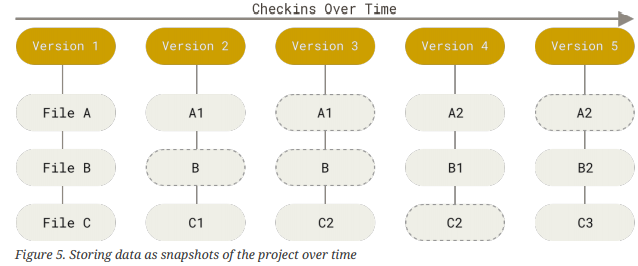
* The most obvious is the single point of failure that the centralized server represents. If that server goes down for an hour, then during that hour nobody can collaborate at all or save versioned changes to anything they’re working on.
* If the hard disk the central database is on becomes corrupted, and proper backups haven’t been kept, you lose absolutely everything — the entire history of the project except whatever single snapshots people happen to have on their local machines.

## Distributed Version Control Systems

### Snapshots, Not Differences

The major differences between Git and any other VCS is the way Git thinks about its data. Conceptually, most other systems store information as a list of file-based changes.

These other systems think of the information they store as a set of files and the changes made to each file over time. (delta-based version control)

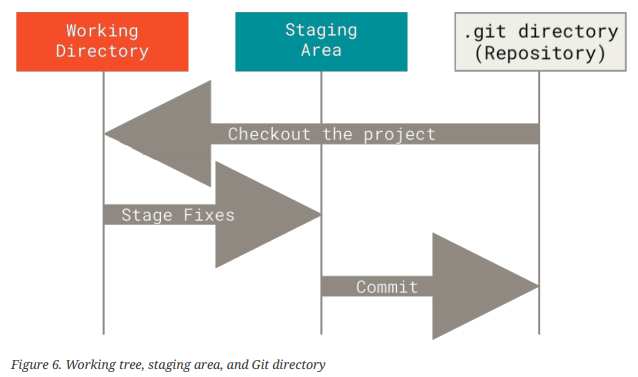
Git doesn’t think of or store its data this way. Instead, Git thinks of its data more like a series of snapshots of a miniature filesystem. With Git, every time you commit, or save the state of your project, Git basically takes a picture of what all your files look like at that moment and stores a reference to that snapshot. (stream of snapshots)

### The three states

The three main states that your files can reside in:

* *Modified* - that you have changed the file but have not committed it to your database yet.

The working tree is a single checkout of one version of the project. These files are pulled out of the compressed database in the Git directory and placed on disk for you to use or modify.

* *Staged* - means that you have marked a modified file in its current version to go into your next commit snapshot.  
  The staging area is a file, generally contained in your Git directory, that stores information about what will go into your next commit. Its technical name in Git parlance is the “index”, but the phrase “staging area” works just as well.
* *Committed* - means that the data is safely stored in your local database

# Git Basics

git rm <filename>

git log –stat