Git

Linux Kernel Community:-

→ global group of developers, ethusiasts who collaborate to develop and improve the Linux kernel.

→ relies on a distributed, collabrative model.

Process:-

Dev submit codechange (patches) —via emails→ Subsystem maintainers.

Subsystem maintainers → Linus Torvalas (for final inclusion)

→ This proces needs a robust version control system.

Bitkeeper:-

→ proprietary (ಪ್ರೊಪ್ರೈಟರಿ), distributed version control system (DVCS).

→ developed by BitMover Inc.

→ designed to manage large, distributed software development projects efficiently.

→ Better than CVS and Subversion.

How Linux uses this:-

Linus Torvalds adopted Bitkeeper in 2002 for Linux Kernel Development. Bitmover provided a free of charge license for the open source community to use it.

The conflict and the breakup

① License restriction:-

developers using the tool for open source projects were not allowed to ~~do contribution on linux kernel development or to~~ work on competing other VCS. ie do not use other shovel or not design any other shovel (VCS).

② Reverse engg Trigger :-

In 2005 Andrew Tridgell ~~the~~ reverse engg--ed the Bitkeeper network protocols to create a functional open source client for it.

As there is a clear violation, BitMover CEO revoked the free license for Linux kernel dev community.

③ The Aftermath:-

→ Results in sudden and unexpected loss of their primary VCS.

→ Linus decided to write his own distributed, fast, open-source VCS.

→ Result was git.

Subversion:- Centralized VCS.

=> single central repo stores all versions of projectfiles.

=> Dependent on server to commit changes.

How it works:

-> Devs "check out" or get a copy of specific files they need from the central server to local machine.

↓

Then they work on those files → To share changes with team.

↓

They must commit their changes back to the main server.

Example

In Library - a book is taken. and write some notes(changes) on it and then return only the official copy for others. to see yours changes. While the book is checked out; no one can modify the specific part.

Drawbacks:-

=> Single point of failure

=> Req internet connection.

=> Slower operations.

Eg:- Subversion (SVN) and CVS.

Distributed VCS:-

→Every developer has a complete copy (or clone) of the entire project repo.

⇒including its full ~~repository~~ history on their ~~sing~~ single local machine.

⇒There is no single master resource.

How it works:-

Devs clone → They work and commit to local repo first

↓

To share changes.

push the local commits to Github.

Analogy:-

→A group of people has their own identical copy of entire book.

→They can make (changes) notes in their book anytime anywhere

→When they meetup. they can swap notes (patches) and merge their changes.

Eg: Git, Mercurial.