

Git and SVN

Gyeongsang National University

Seongjin Lee

2017-1 Data Structure and Algorithm

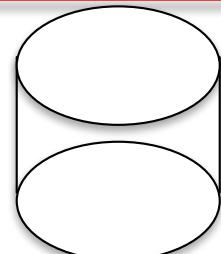
The history

Generation	Networking	Operations	Concurrency	Examples
First	NONE	One file at a time	Locks	RCS, SCCS
Second	Centralized	Multi-file	Merge before commit	CVS, Subversion, SourceSafe, Team Foundation Sever
Third	Distributed	Change sets	Commit before merge	Bazaar, Git, Mercurial

- Forty years of history of version control tools
 - Steady movement toward more concurrency
 - 1st gen. used locks to manage concurrency – one person at a time
 - 2nd gen. are more permissive about simultaneous modification – merge before commit
 - 3rd gen. merge and commits are separated

Basic Terminology

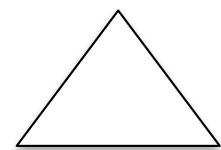
- Repository – official place to store the work
 - Keeps track of tree of files and directories
 - More importantly it contains history
 - **create** operation makes a new repository



repository

Repository = filesystem x time

- Checkout – create a working copy of existing repository
 - Records timestamp on the working file
 - Records the version number of the repository file (to note the start)
 - Keeps complete copy of the retrieved file



Working copy

WorkCycleFromStart:

make a working copy from repository

WorkCycle:

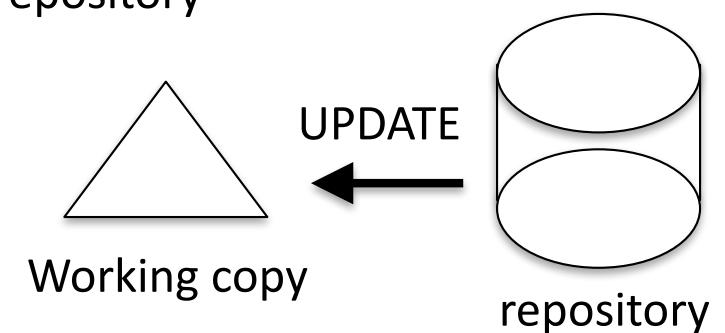
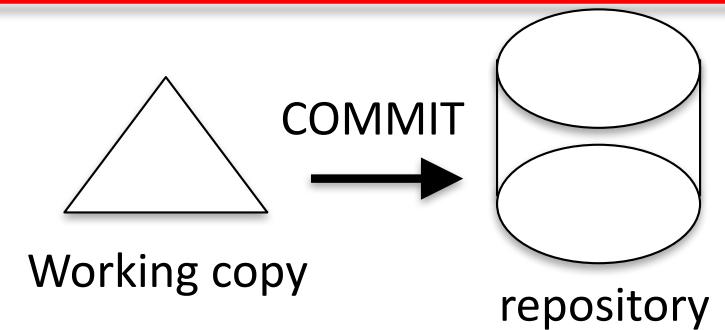
modify working copy

update the repository

GOTO WorkCycle

Basic Terminology

- Commit – apply modification in the working copy to the repository as a new change set
 - Several others modify the working copy and add an operations to a pending changeset list
 - Pending changeset – a place where changes wait to be committed
 - Commit operation takes the pending changeset and makes it to create a new version of the tree in the repository
 - Operations are atomic – all or nothing
- Update – update the working copy with respect to the repository
 - Make working copy up-to-date
 - Apply changes from the repository, merge them with any changes on the working copy



Basic Terminology

- ADD – add a file or directory for version control
 - After add they become part of the pending changeset
- EDIT – modify a file
 - Edit operation does not involve the VCS
- DELETE – delete a file or directory
 - Remove a file or directory from the repository
 - Immediately delete the working copy of the file, but they are left in pending changeset
 - File / directory in the repository is not really deleted; just making a new tree w/o them
- RENAME – rename a file or directory
 - Some of the earlier tools had no support for it; so, should check how your VCS works
- MOVE – move a file or directory
 - Move file or directory from one place in the tree to another
 - Operation is added to the pending changeset

Basic Terminology

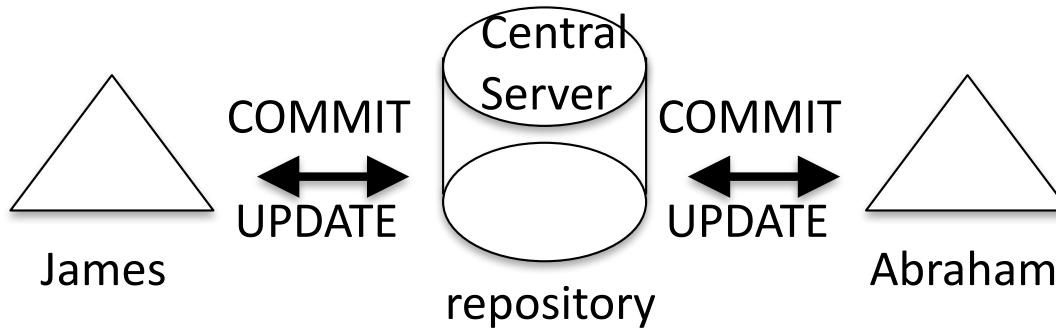
- STATUS – list the modifications that have been made to the working copy
 - It shows the list of pending changeset
- DIFF – shows the details of the modifications that have been made to the working copy
 - Status for list and diff for what exactly have been changed
 - How it prints out the differences is VCS dependent
- REVERT – undo modifications that have been made to the working copy
 - Throw away all your pending changeset and the return the working copy to the way it was just after the checkout
- LOG – show the history of changes to the repository
 - Keeps track of every version
 - Who; When; What

Basic Terminology

- TAG – associate a meaningful name with a specific version in the repository
 - To mark a specific instant in the history of the repository with meaningful name
- BRANCH – create another line of development
 - To fork off into two different directories
- MERGE – apply changes from one branch to another
 - Used branch to enable the development to diverge, merge is to converge again
- RESOLVE – handle conflicts resulting from a merge
 - VERY IMPORTANT
- LOCK – prevent other people from modifying a file
 - Not all have this feature

2nd Generation Background: SVN

- Centralized version control system
- \$ mkdir projectA
- \$ svnadmin create projectA/HAM
- \$ svnserver -d --root=/Users/James/projectA



- Checkout, add, status, commit
 - \$ svn checkout http://----
 - \$ svn add YOURFILE
 - \$ svn status
 - \$ svn commit -m "LOG CONTENT"
 - \$ svn log
 - \$ svn diff

2nd Generation Background: SVN

- Merge changes to the working copy
 - \$ svn update

Select: (p) postpone, (df) diff-full, (e) edit, (r) resolved,
(mc) mine-conflict, (tc) theirs-conflict,
(s) show all options:

- Postpone – deal with the conflict later
- Resolved – mark it as solved
 - \$ svn resolve --accept=working
- Mine-conflict – use my version as new
- Theirs-conflict – use the repository as new

<<<<<< .mine

Some text is introduced in this line
This is what James wrote

=====

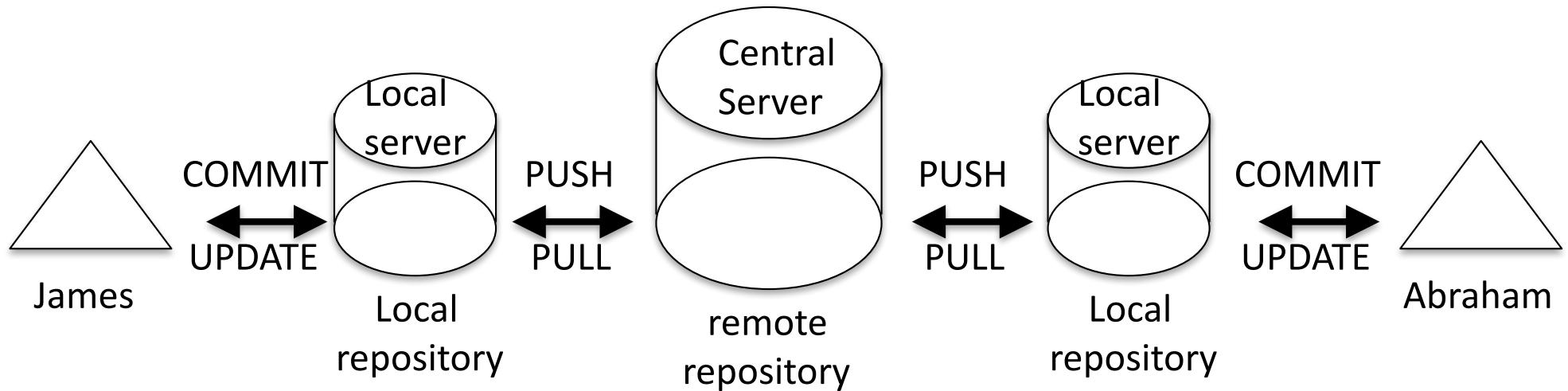
Some different contents in this line
This is what Abraham wrote
>>>>> .r4

Summary of operations in SVN

- Create \$ svnadmin create
- Checkout \$ svn checkout
- Commit \$ svn commit
- Update \$ svn update
- Add \$ svn add
- Delete \$ svn delete
- Rename \$ svn move
- Move \$ svn move
- Status \$ svn status
- Diff \$ svn diff
- Revert \$ svn revert
- Log \$ svn log
- Tag \$ svn copy
- Branch \$ svn copy
- Merge \$ svn merge
- Resolve \$ svn resolve
- Lock \$ svn lock

3rd Generation Background: GIT

- Distributed or decentralized version control system



Synchronizing the local and the remote

- PUSH – copy changesets from a local repository instance to a remote one
- PULL – copy changesets from a remote repository instance to a local one
- Note that not all changes on the local is same as that on the remote

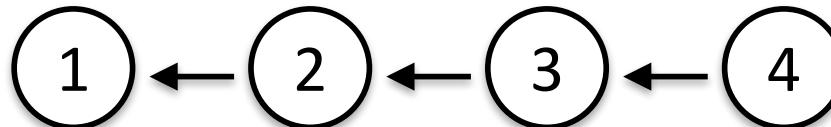
3rd Generation Background: GIT

- Directed Acyclic Graph (DAGs)
 - Ability to push and pull changesets between multiple instances of the same repository comes from a design model called DAG
 - Consists of Node, directed edge, root node, leaf node
 - Node – represents one revision of the entire repository tree
 - Directed edge – shows relationship between nodes

3rd Generation Background: GIT

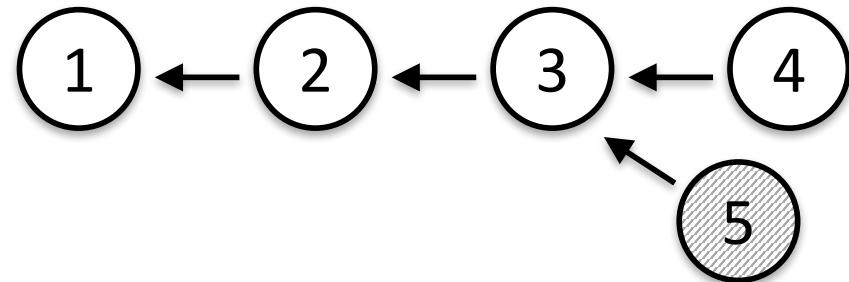
- Repository history as a line

1. Fork latest version
2. Modify
3. Check back in



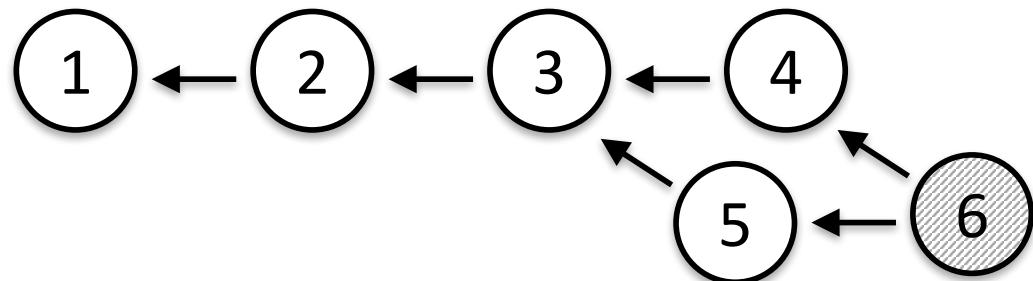
- Somebody else does the same

- Before I make change to version 3 somebody else already made version 4



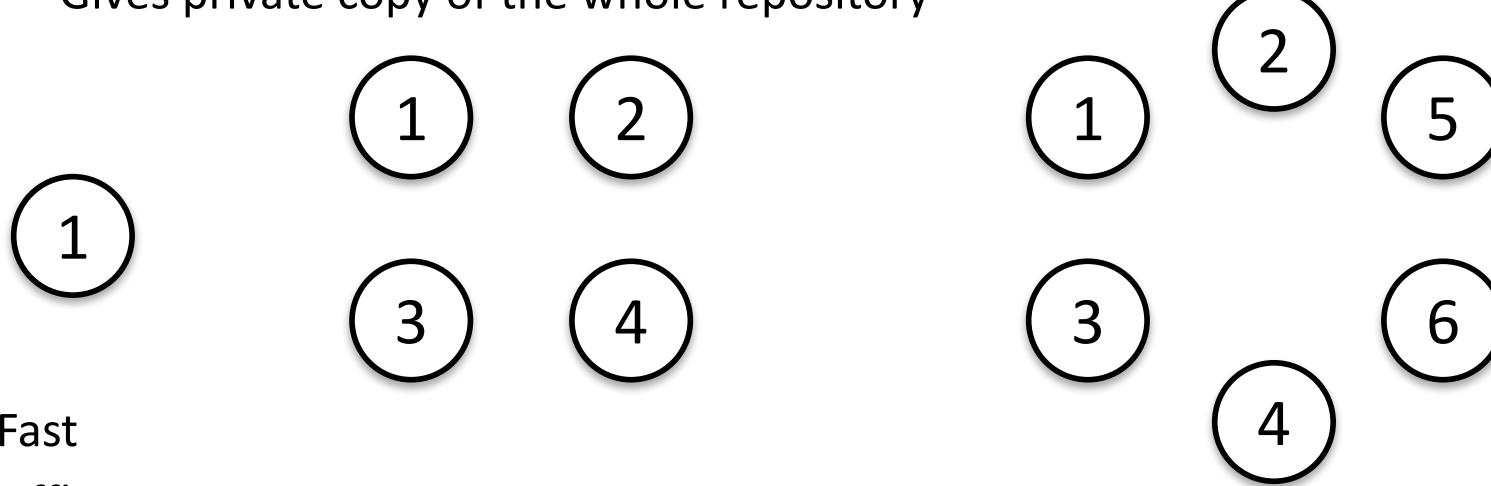
- REMEMBER to commit before merge

- Benefit of having DAG model
 - Everything is not linear
 - Flexible and expressive

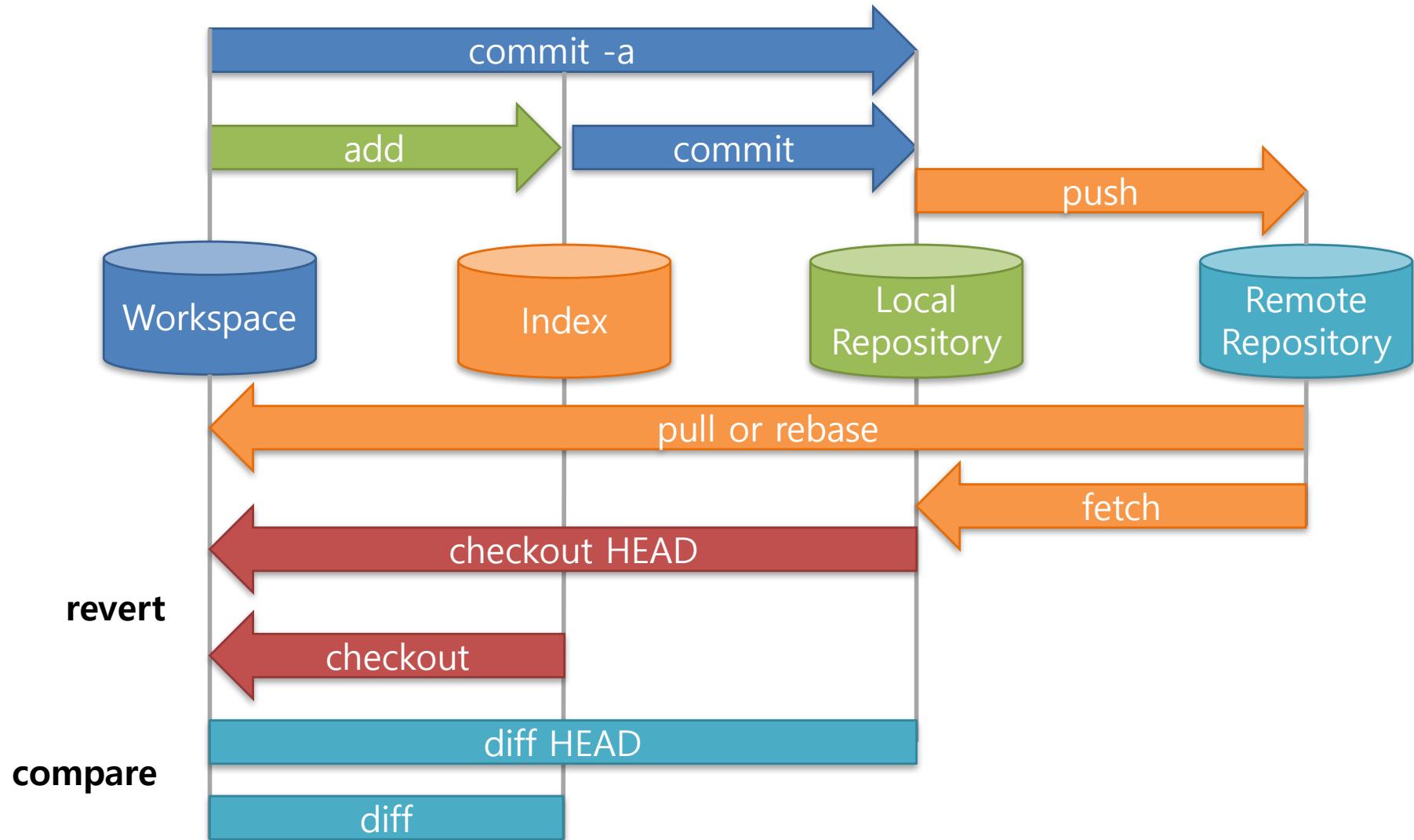


3rd Generation Background: GIT

- Advantages of DVCS
 - Private workspace
 - Gives private copy of the whole repository
 - Fast
 - Offline
 - Scale out, and scale up

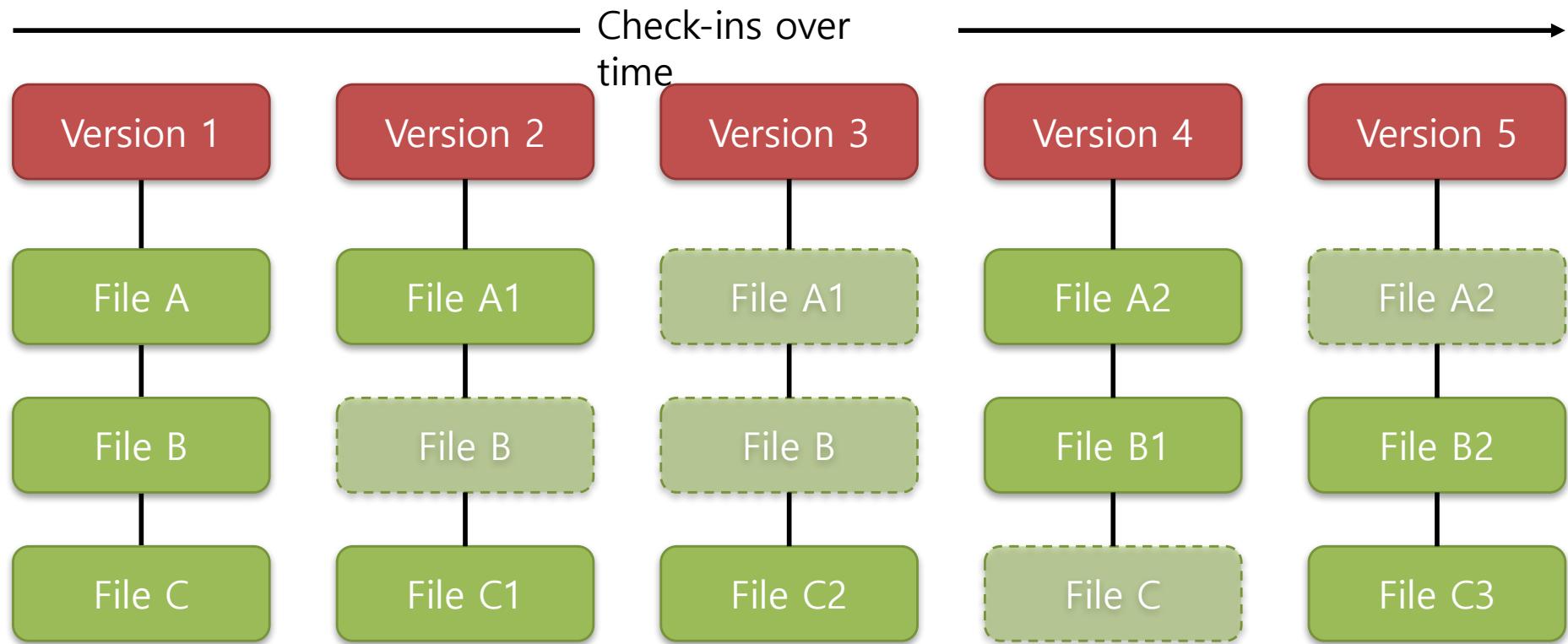


Flow of Working with Git



File management in Git

- It keeps the delta of the object



Basics with Git

- IF you have a git server already
- \$ mkdir project_with_git
- \$ cd project_with_git
- \$ git init --bare project_with_git

- Using Github.com
- \$ vi ~/.gitconfig

```
[user]
    name = YOURID
    email = YOUR-EMAIL
```

- \$ git clone ADDRESS
- \$ git push OR \$ git push -all
- \$ git pull
- \$ git log
- \$ git status -s
- \$ git diff

Basics with Git

- \$ git commit -a -m “fixed and revised the text”

- Or use \ to comment many lines

- \$ git push

- \$ git pull

- \$ git show -c // for showing changes

- Conflicts

<<<<<< HEAD

Some text is introduced in this line

This is what James wrote

=====

Some different contents in this line

This is what Abraham wrote

>>>>> b30hf32hfaohf8dhaf8a

Summary of operations in Git

- Create \$ git init
- Checkout
- Commit \$ git commit -a
- Update \$ git checkout
- Add \$ git add
- Edit \$ git add
- Delete \$ git rm
- Rename \$ git mv
- Move \$ git mv
- Status \$ git status
- Diff \$ git diff
- Revert \$ git checkout
- Log \$ git log
- Tag \$ git tag
- Branch \$ git branch
- Merge \$ git merge
- Clone \$ git clone
- Push \$ git push
- Pull \$ git fetch

Installing the version control systems

GIT

- For linux \$ apt-get install git
- For mac <http://git-scm.com/download/mac>
 - Or link is http://sourceforge.net/projects/git-osx-installer/files/git-2.5.3-intel-universal-mavericks.dmg/download?use_mirror=autoselect

SVN

- For linux \$ apt-get install subversion libapache2-svn
- For mac - Build using source code or check if it has one
- Check version
 - \$ git --version
 - \$ svn --version