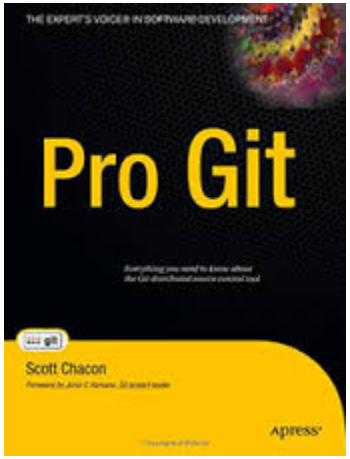
Disclaimer: This presentation is based on Pro Git



Git Introduction and Practices

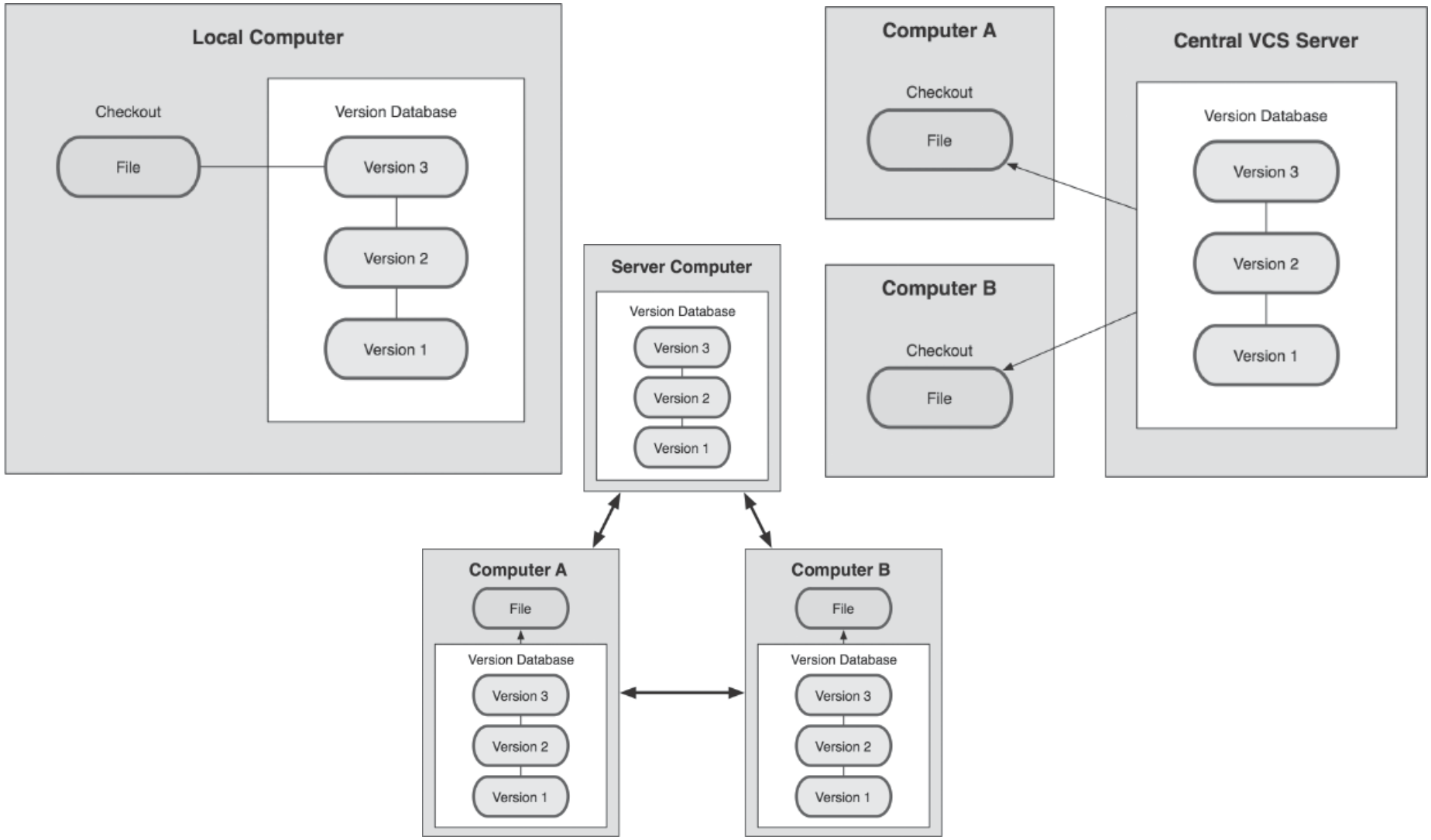
cmput 301 lab 3

Haiming Wang

Contents

1. VCS introduction
2. Git knowledge
    a. how does git work, file system+communication protocol
    b. local commands
    c. branching
    d. work with remote server
    e. workflow
3. GitHub

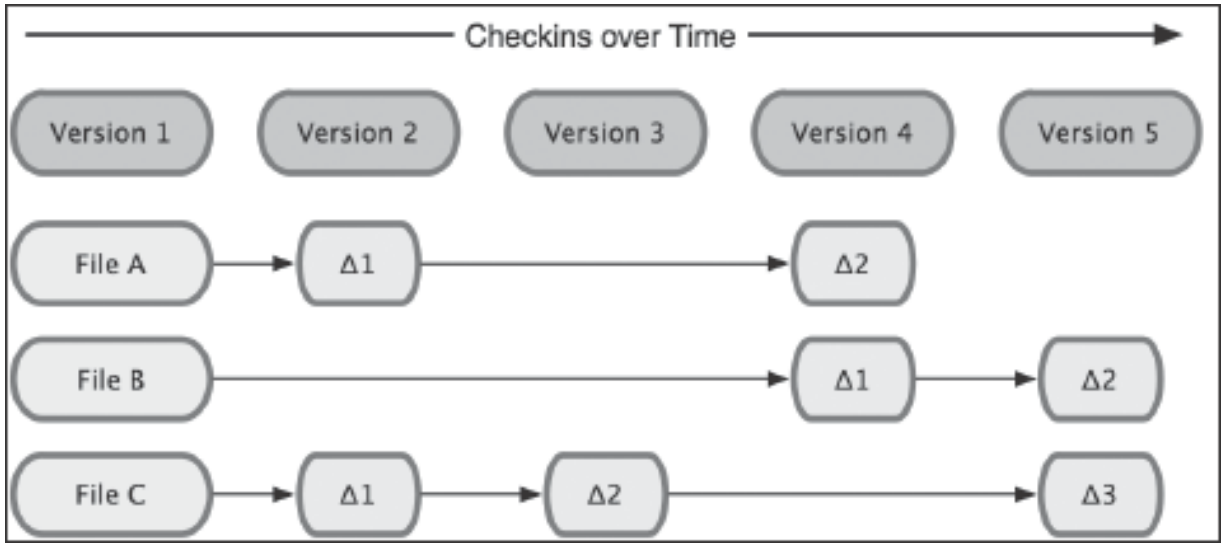
Version Control System(VCS)

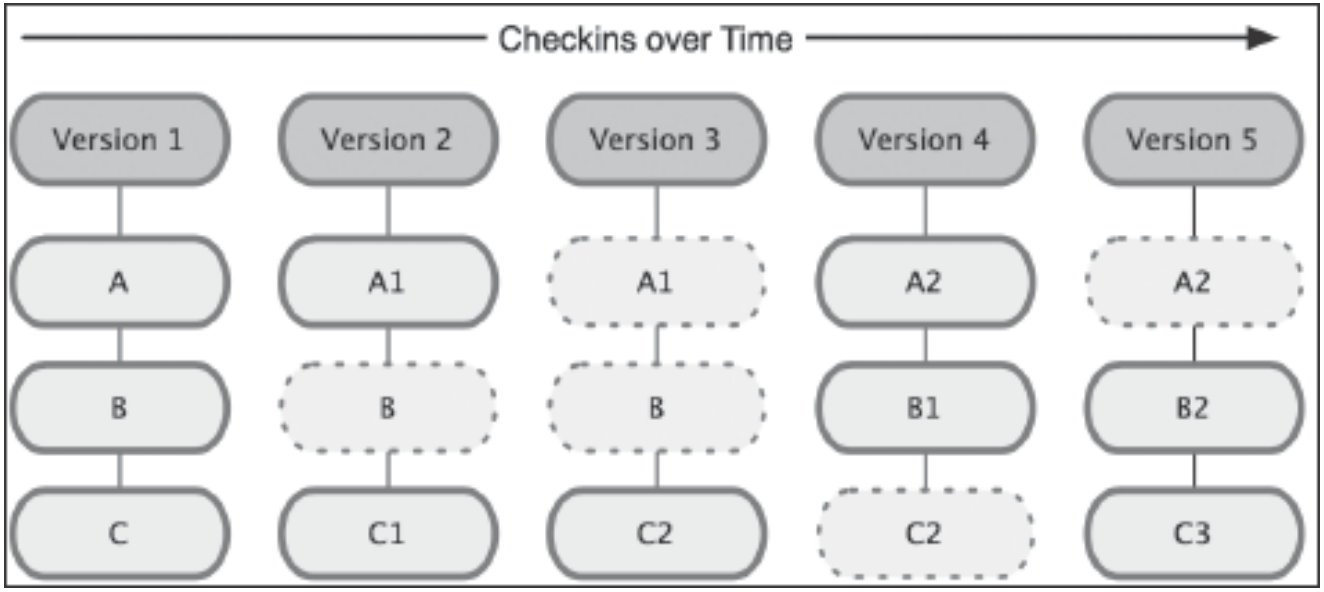


Git Basics

- Design Goal

other system ->





← git

Git Setup

1. $ git config --global user.name "301git"

   $ git config --global user.email haiming1@ualberta.ca

2. $ git config --list
   - multiple keys(e.g. user.names)
   - $ git config user.name

3. $ git config --global core.editor vim

   $ git config --global merge.tool vimdiff

4. git help <verb>

Get a Git repository

- Cloning an Existing Repository

- Initializing a Repository in an Existing Directory

Recording Changes to the Repository

- Check status
- Add a new file
- Staging Modified Files
  - modify after staging?
- Ignoring files:
  - .gitignore

```
# a comment - this is ignored
*.a       # no .a files
!lib.a    # but do track lib.a, even though you're ignoring .a files above
/TODO     # only ignore the root TODO file, not subdir/TODO
build/    # ignore all files in the build/ directory
doc/*.txt # ignore doc/notes.txt, but not doc/server/arch.txt
```

- git diff
- git commit -am
- git rm (--cached)
- git mv

Undoing Things

- Changing Your Last Commit
  - $ git commit -m 'initial commit'
  - $ git add forgotten_file
  - $ git commit --amend
- unstage
  - git reset HEAD <file>
- unmodify
  - git checkout -- <file> (Dangerous)

Viewing the Commit History

- git log
  - -n print the n most recent ones
  - -p print the diff
  - --pretty = oneline, short, full, and fuller, format:"%h - %an, %ar : %s"
  - --author = “301git” # filter
  - --graph # to show branching and merging history
  - --all # all branches, another example, gitk
  - --color #

Work with Remote

- Sign up GitHub account
- Set up a repository
- Add remote
  - git remote add [shortname] [url]
- Show remote
  - git remote -v
- Fetch data
  - git fetch shortname
  - git pull
- Push data

Branching

- what a branch is:
  - http://git-scm.com/book/en/Git-Branching-What-a-Branch-Is
- git branch <new branch name>
- git checkout <branch name>
- git checkout -b <new branch name>
- git merge <branch name> / rebase
- merge conflict:
  - edited the same area(status -> edit -> add -> submit)
  - one branch removed file, while another modified
    - edit
    - remove

Branching with remote

- git push <remote> <list of branches>
- git fetch <remote>
- git pull <remote> <branch>
- git checkout -b <local branch name> <remote name>/<remote branch name>

Workflow

- Branch workflow

http://git-scm.com/book/en/Git-Branching-Branching-Workflows

- Small team workflow

http://git-scm.com/book/en/Distributed-Git-Contributing-to-a-Project

GitHub

- SSH, HTTP: Protocol difference
  - ssh: faster, secure, authenticated, easy to set up
  - http: provide read access
- Add ssh public key to github
  - ssh-keygen -t dsa
  - ssh-agent
  - ssh-add
  - copy *.pub to github
- Add collaborator to allow them write access
  - write access enabled
- Fork a project