SIM Global Education

Topic 10
Version control systems

Learning Outcomes

After completing this topic and the recommended reading, you should be able to:

- Access pre-existing git repositories using command line tools.
- Create a new git repository and add files to it.
- Explain the purpose of git branching, use it and justify its use in different contexts.

1. Version Control

- ***Version Control*** is a class of systems responsible for managing changes to computer programs, documents, large websites, or other collections of information.

Releases / Versions

- Single version, latest release
- Multiple versions, multiple releases
- Multiple developers, asynchronous updates
- Open-source vs. closed-source software
- Examples:
    - iOS; MacOS; Windows OS; Linux; etc

2. Version Control Systems

- ***Version Control Systems*** (VCS) are software tools that help software teams manage changes to source code over time.
    - Undertakes the tedious task of keeping track of the changes to all project's files and who made them
    - Allows users to recover any previous version at any given time

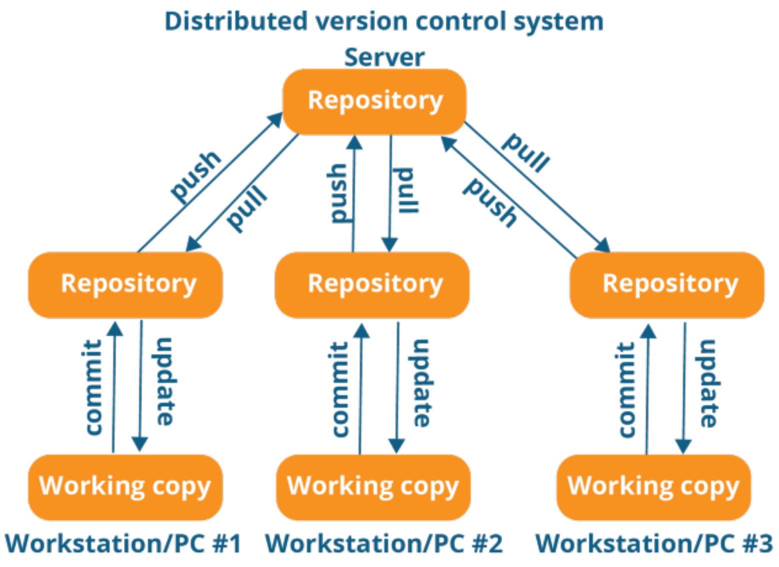
Examples

- Source Code Control System (SCCS)
- Apache Subversion (SVN)
- Concurrent Versions System (CVS)
- Bitkeeper
- Git

Git

- Originally created by *Linus Torvalds*, in 2005, for Linux kernel development
- Source Control Management System for Linux kernel code
- Distributed version control system (DVCS)
    - All project files and their histories are present both remotely and in the computers of all developers contributing to the project
        - vs. Central Server (SVN/CVS), updates clash
    - Developers can work offline and asynchronously without a constant connection to a central repository
        - Branching & Merging locally
    - Collaboration:

    - Commit separately (without disturbing others)
    - Merging good branch
- Reliable
- Good performance
- Content Management
  - SCM:
    - Source Code Management
    - Software Configuration Management
    - Source Control Management
  - What comes out is what goes in



[Source: *https://medium.com/@sahoosunilkumar/how-does-git-works-5cc8444ea383*]

3. Git Repositories

- ***Git repository*** contains the collection of the files and directories, as well as the history of changes made to those files.
  - A local directory holding a project's files and folders that is not under version control is turned into a Git repository
  - A remote Git repository is cloned into your computer from elsewhere

Remote Repositories / Git Servers

- Other features (vs. local)
  - Version tracking
  - Issue tracking
  - Users or key stores
  - Clone / Backup
  - Community

Examples

- GitHub
  - Public
  - https://github.com
- GitLab
  - Own server
  - https://about.gitlab.com
- Bitbucket
  - https://bitbucket.org

4. Basic Git Operations

Installing Git

- Go to “Git”, click at “Download for Mac” or “Download for Windows”
    - https://git-scm.com/

Setting up Git Credentials

- git --version
    - *# show the git version*
- git config --global user.name “Handsome Koh”
- git config --global user.email “chkoh005@mymail.sim.edu.sg”

Setting up Git Repository

- cd ~/directory/gitKOH/
    - *# change to gitKOH directory*
- git init
    - *# initialise empty Git repository in gitKOH*
    - *# ~/directory/gitKOH/.git/ created*

Basic Git Commands

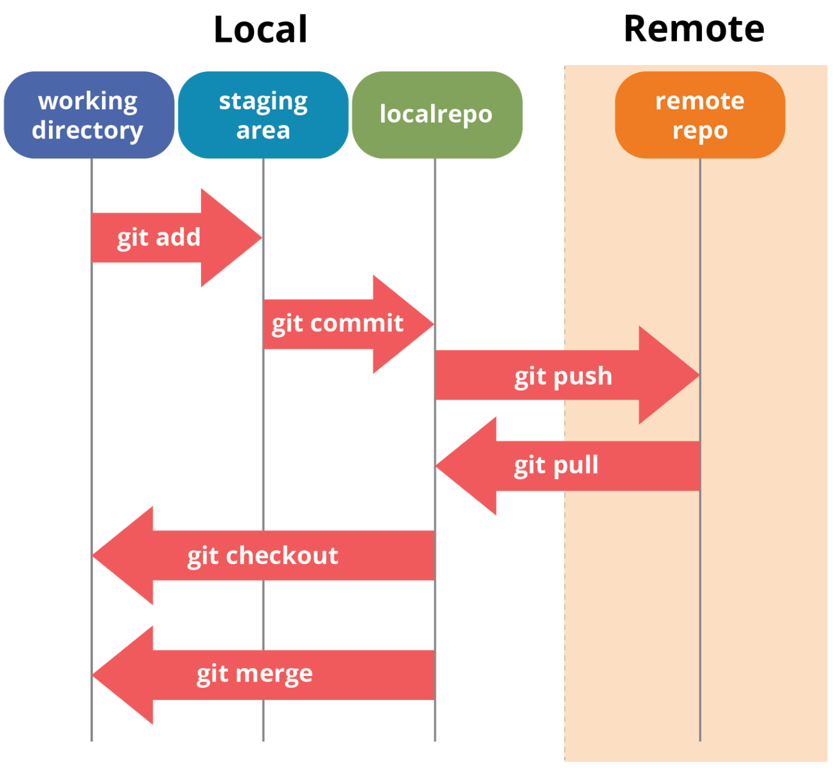
- git status
    - *# check repository current status*
    - *# branching; committing; staging*
- git add example.py
    - *# git add <filename>*
    - *# stage / adding example.py to repository for version control*
- git rm --cached <filename>

    - *# un-stage / removing file from repository*
- git commit -m "Handsome Version 1"
    - *# captures a snapshot or milestone along the timeline of a Git project*
    - *# commits are created to capture the project's currently staged changes*
    - *# confirm a staged files*
- git commit --amend
    - *# un-commit / edit previous commit*
- git log
    - *# show complete log of all changes*
    - *# list the commits done so far*
- git branch
    - *# checking the branch currently at*
    - *# a pointer to a snapshot of changes*
    - *# git stores a branch as a reference to a commit, instead of copying files from directory to directory.*
- git branch first-branch
    - *# create a new branch name first-branch*
- git checkout first-branch
    - *# switch to branch first-branch*
- git checkout master
    - *# switch to master branch*
- git merge first-branch
    - *# merge first-branch into master*
- git reset

Interact with Remote Repository

- git clone <url>
  - *# clone from a repository*
- git push
  - *# push to the repository*
- git pull
- git fetch

- Branch and Fix (some codes)
  - git checkout -b my_fix
- Commit and Merge (the changes)
  - git commit -a -m "commit for my_fix"
  - git checkout master
  - git merge my_fix



[Source: *https://medium.com/@sahoosunilkumar/how-does-git-works-5cc8444ea383*]

5. Practice Quiz

- Work on *Practice Quiz 10* posted on Canvas.

Useful Resources

-
    - http://