

CS2104 Problem Solving in Computer Science

*Margaret Ellis,
Naren Ramakrishnan,
Sehrish Basir Nizamani*

Command Line and Git Skills



Objectives

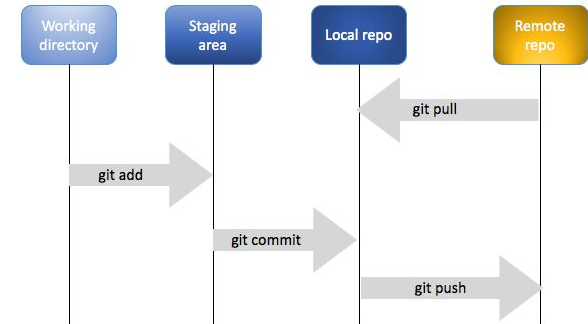
- Use basic git commands
- Collaborate on writing and testing algorithms using Git
- Analyze the use of LLMs for version control

Git Intro

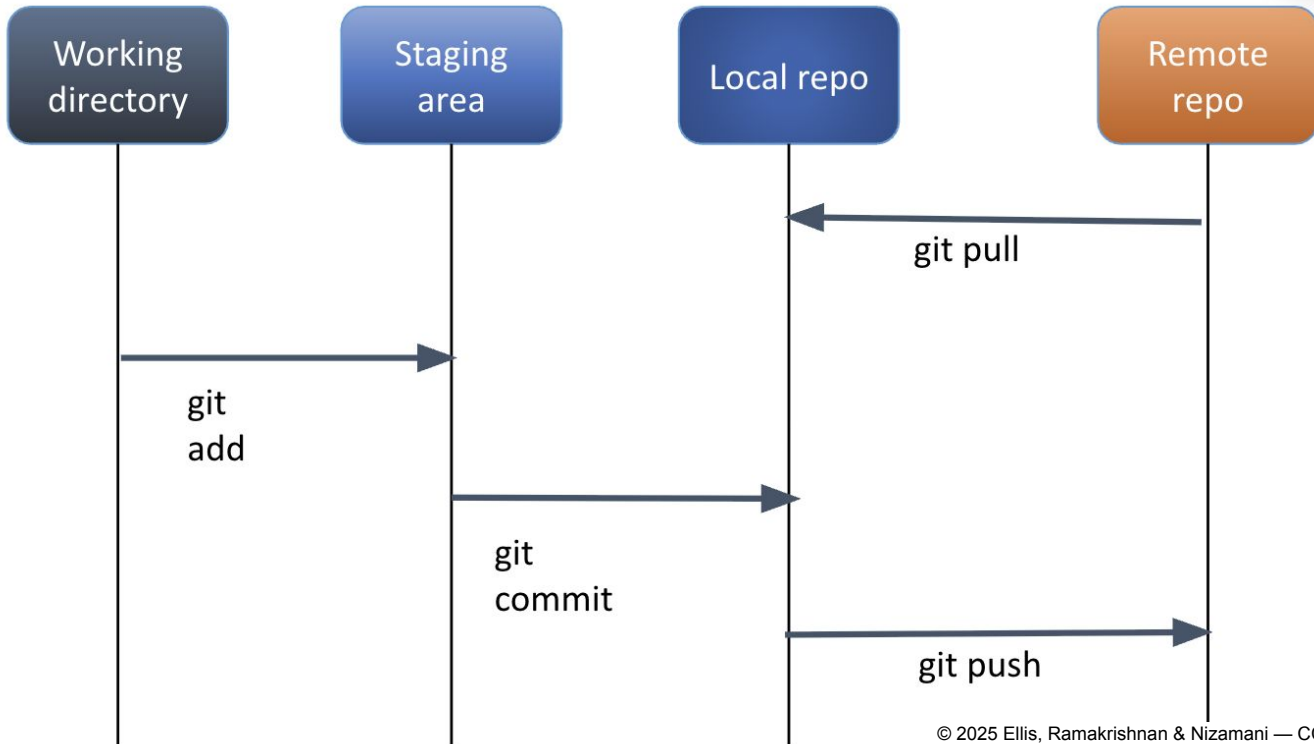
- Version Control (e.g. [gitHub](https://github.com) and [Bitbucket](https://bitbucket.org))
- We will use gitLab git.cs.vt.edu
- Purpose
 - Track and document file versions
 - Collaborate
- Key concepts
 - Local and remote repositories
 - Stay synced by pulling and pushing your project
 - Stage and commit files as your work

Git commands (Demo example)

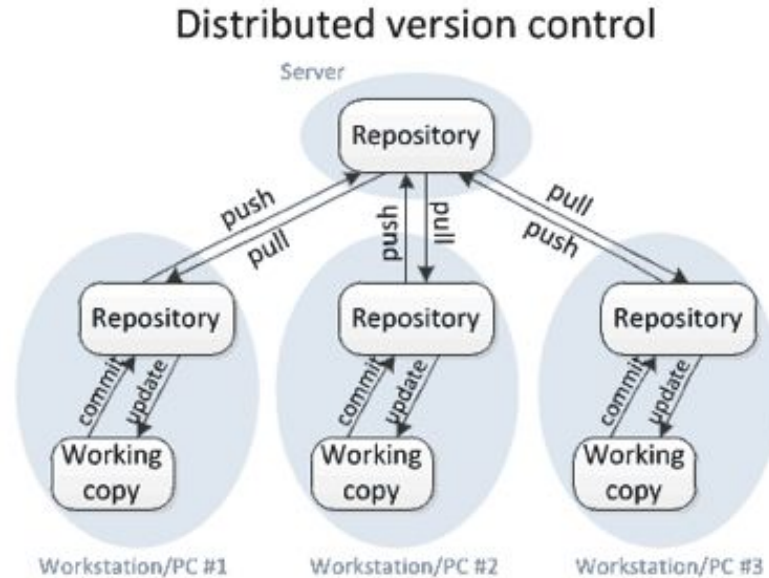
- Start a project
 - Create a remote repository on git.cs.vt.edu
 - New Project
 - Check box to initialize repository
 - Create a local repository in the directory of your choice using
 - `git clone <path to repository>`
 - drop down Code menu, use icon to copy, then paste to terminal (ssh from rlogin)
- To make changes locally to your project in the working directory
 - Pull (so you are up-to-date with the remote repository)
 - `git pull`
 - Do your work and add or remove files to stage them as needed
 - `git add <file name>`
 - When you are satisfied with changes, commit them
 - `git commit -m "Made changes to deal with semester change"`
 - Push changes to the remote repository
 - `git push`



Git



Distributed Version Control



<https://homes.cs.washington.edu/~mernst/advice/version-control.html>

Git: Checking in files

When in
doubt, use
“git status”

- Always **pull** the latest version of the remote repository.
- Then create or edit files, be sure to **add** them so they get staged locally.
- When you have file(s) ready, you **commit** them locally.
- When you are ready to share, you **push** them to the remote repository.

.gitignore file

.gitignore is a hidden file that lists files or file types to not include in the repo

- These files then do not get shared/included in the repo
- For example: log files, configuration files, data files
- Note that a file that starts with . is “hidden”, need ls -a
- Examples of useful templates
<https://github.com/github/gitignore>

#General Files

*.csv

*.txt

*.pdf

#Editor-related files

*.vscode

#macOS-related files

.DS_Store

Git options

- IDEs provide interfaces for using git, get to know them (they may not work how you expect)
- Github Desktop provides a GUI interface for github (used in CS2114)
- LLMs can be used to ask questions about git commands
 - Usually ok for intro tasks
 - Less reliable for advanced tasks
 - Be careful



Git Commands Practice Day



- Form groups
- Work together on the CW

Heads up about git and LLMs

**As you start working with branches and multiple developers,
git gets complicated quickly**

- This is a place where a little knowledge can be a dangerous thing
- The best way to *get good at git is to use git*
- If an error is made, there are mechanisms to roll back but it can quickly become “digging yourself out of a hole”
- ***Hence.... Do not use commands without understanding them (aka don't simply trust a LLM!)***

Git Algorithms and Tests Assignment

- Make your group. It is based on your “**Git Group**” Group number in canvas
- Work together on the CW
- Add the UTA who will be reviewing your work to your Repo: