

Lecture 2 (1-14-26)

- in order to use ssh, you at least need to know the other machines location
 - aka the **domain name**: `student10.cse.nd.edu`
- when you connect to student10 it creates a **shell** (bash)
 - this shell is not running locally, its running on student machine
- anything you type on the new shell, it is run on the remote machine
- all student machines use **NFS** (network file system)
 - this is how you see the same files no matter what student you log into
- if you are off campus, the student machines won't work
 - this is because of a **firewall**
 - this firewall blocks traffic
 - so you need a **vpn** to access the machines off campus
 - this allows you to go through the firewall

GIT

- **git is a program**
- **github is a hosting platform**
- Git is a **DVCS** (distributed version control system)
 - git is a way to maintain different versions of the file
 - git is a **journal** (records transactions)
 - git is a time machine
 - git is a share space

GIT : ANALOGY

- think of it as a **checkbook**
- every time you commit, it's a transaction you write down
- in git you **can go back to a previous transaction** and remove any subsequent changes
- `git log` to see the "checkbook"
 - this will return a list of commits with id numbers
 - if you do `git show id#` , it will show you the changes made

GIT : MODEL

- **Upstream** - this is the original pbui repo
- Then I created a copy of the repo via forking

- we call this the **origin**
- **Both of these repos are in the cloud**
- to get this copy from the cloud, I need to **clone** to get it to my student machine
 - you only need to **clone** a repo **once per machine I want to use it on** (might be on exam)
- We **checkout** the project files from our local mirror to populate the working directory. (this is usually automatically in clone)
- We use `git add` to add files to the **staging area**
- We **commit** changes to permanently record the staging areas to the "checkbook"
- We **push** to send local changes to the github cloud
- **REVISIT GIT PULL (FETCH AND MERGE)**

GIT : BRANCHES

- There is a master/main branch
- We will create a new branch for every assignment (ex reading 01)
 - all the work for reading01 should be on the reading01 branch
 - **Commands:**
 - `git switch master`
 - `git pull --rebase` -> takes new commits and appends them
 - `git switch -c reading01` -> creating branch reading01 and switching to it
- `git branch` will show you the branches you have
-
- **you will write reading quiz answers in a json file**
- `git diff` tells you the things that have been changed since the last commit
- `git restore` **reverses** changes from the staging area