

CSE 374: Programming Concepts and Tools

Eric Mullen
Spring 2017
Lecture 17: git

Administrivia

- Homework 5 out now, due next Thurs
 - Start **TODAY**, or **SATURDAY** at the latest
- Homework 6 with partners, survey out this afternoon
 - Choice due next Wed

Where we are

- Last lecture: all about make
 - Tool for multi-file projects
- This lecture: all about git
 - Tool for multi-person, multi-platform, multi-month projects
- In general: talking about Tools

Tools

- make and git are not C specific
- You can use them with whatever language you want
- Be careful with git and large files (stick to mainly text files for sanity)

Version Control

- There are plenty of version control systems
- Terminology and commands are not standard, but concepts are
- In use today: svn and perforce (central repository), and git and mercurial (distributed repository)

If that doesn't fix it, git.txt contains the phone number of a friend of mine who understands git. Just wait through a few minutes of 'It's really pretty simple, just think of branches as...' and eventually you'll learn the commands that will fix everything.

THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.

COOL. HOW DO WE USE IT?

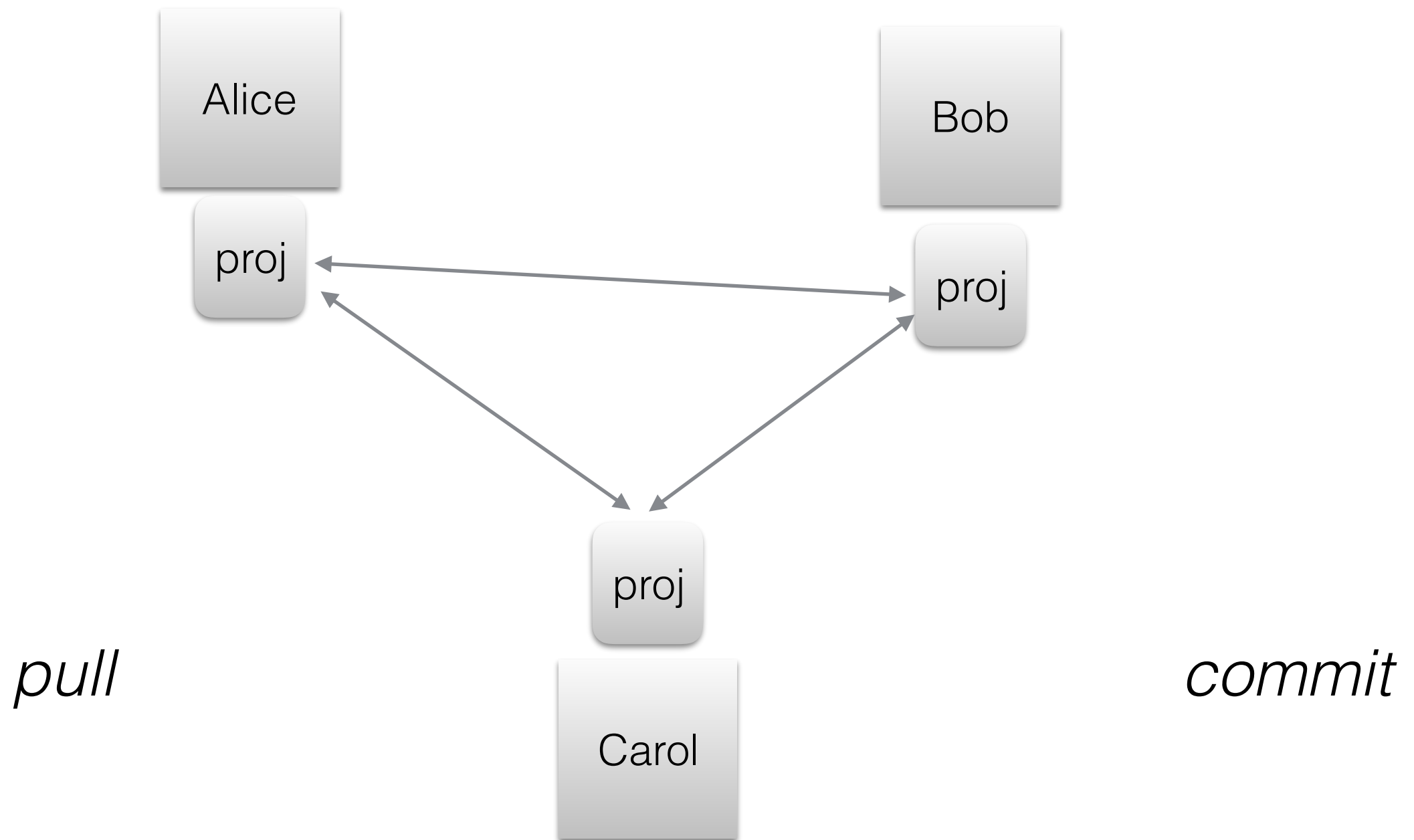
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.



git history

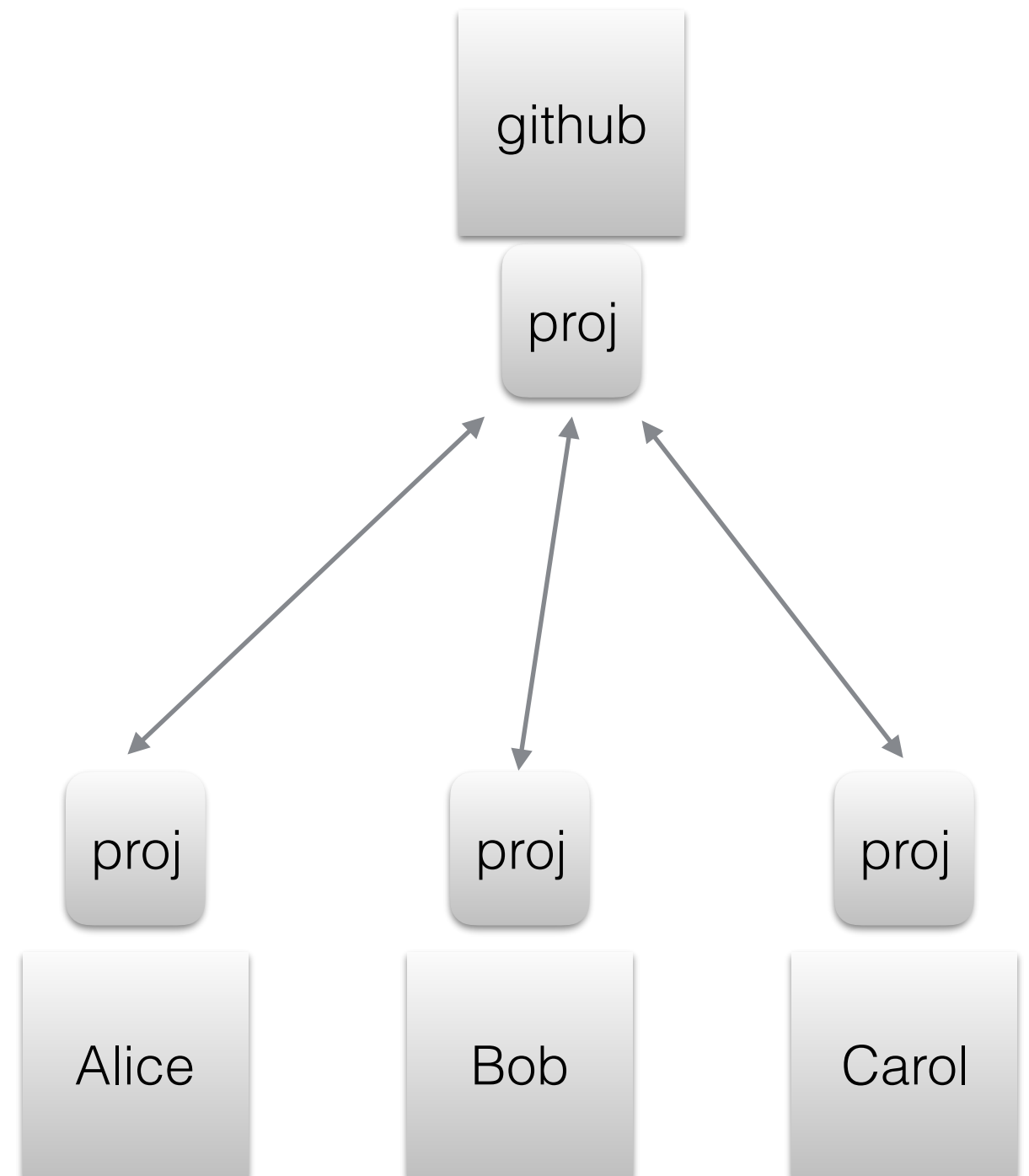
- First release in 2005
- Built by Linus Torvalds
- Allegedly stands for “global information tracker” or “g*dd@mn idiotic truckload (of sh*t)” depending on if it’s working or not
- Many version control systems came before, git is very prevalent in the world now

git basics



git basics (central repo)

- Users have a shared repository (called *origin* in git literature)
- Each user *clones* the repository
- Users *commit* changes to their local repository
- Users *push* changes to origin
- Other users *pull* changes from origin



Using git

- Memorize a few common cases, look up the rest
- Create
 - a new repository: rare
 - a new branch: more common (not used in 374)
 - a new commit: all the time (used in 374)
- Push to repo
 - regularly (push early and often, very effective backup)
- Other operations rarely

Common Commands

- edit a file foo.c (use your favorite editor)

1. add file to list of things to commit

```
git add foo.c
```

2. commit all added changes

```
git commit -m "commit message here"
```

3. push changes to central repository

```
git push
```

Conflicts

- Suppose someone else pushed first
 - You now have to integrate their changes into yours
 1. use `git pull` to get their changes
 2. fix any merge conflicts
 3. `git add` all merged files
 4. `git push`

File rename/move/delete

- `git rm` works just like `rm`, but also tracks deletion from repository
- `git mv` works just like `mv`, but also tracks deletion from repository
- No special command to make new file, just make it locally then `git add`

Demo



Be careful

- There are many other git commands, and a simple google search will turn them all up
- There are many ways to get your repository in an incomprehensible state
- Come to TAs or office hours before running commands you find on the internet
 - Especially if they contain `branch`, `rebase`, or other scary words

Enjoy the weekend!

- High 50s and sunny, almost no chance of rain all weekend
- Send the staff a picture of you working on 374 in a cool place, best pictures featured at the end of future lectures