Computer System Organization
Recitation
[Spring 2018]
CSCI-UA 0201-006

R1: Introduction

What to submit today?

- Read CSO_CHEAT_SHEET.md and modify all “fegin” to your GitHub user name.

- The document contains some important logistics for the class and instructions of GIT.

- The deadline is 11:59 PM 3/9.

- Not submitting the document will result in the overall recitation point deduction.

Logistics

1. For all future labs and recitations, the scores will be post to NYU classes. We will post on Piazza when the scores are ready.
    1. If you have any question, you must contact us before the deadline specified in the post.
2. All the deadlines are 11:59 PM on the due date. Any late submission will not be considered.
    1. For example, if you submit at 0:06 AM for recitation, that submit will be ignored.
3. All the future recitation deadline will be on Tuesday nights instead of Monday.

Logistics

4. You must follow all the instructions we post on Piazza.
    1. For example, we have posted how to use grace days for labs. In the future, any submission with incorrect format will be ignored.

5. When doing labs, you have to always follow the instructions. Don't change the output format…

6. When doing labs, you should only modify the files which the instructions ask you to modify.

7. Plagiarism:
    1. For game of life, you can look up how to read a file in C language on the Internet.
    2. But the logic of game of life must be entirely implemented by you.

Logistics

7. When doing labs, I should only modify the files which the instructions ask us to modify. Even I really have to modify some files for testing purpose, I should never commit them.

Git

- CSO_CHEAT_SHEET.md teaches you how to solve some git issues you may have in this class. Read it and follow the instructions before asking.

- Please type the command “sync” after you finish your work.

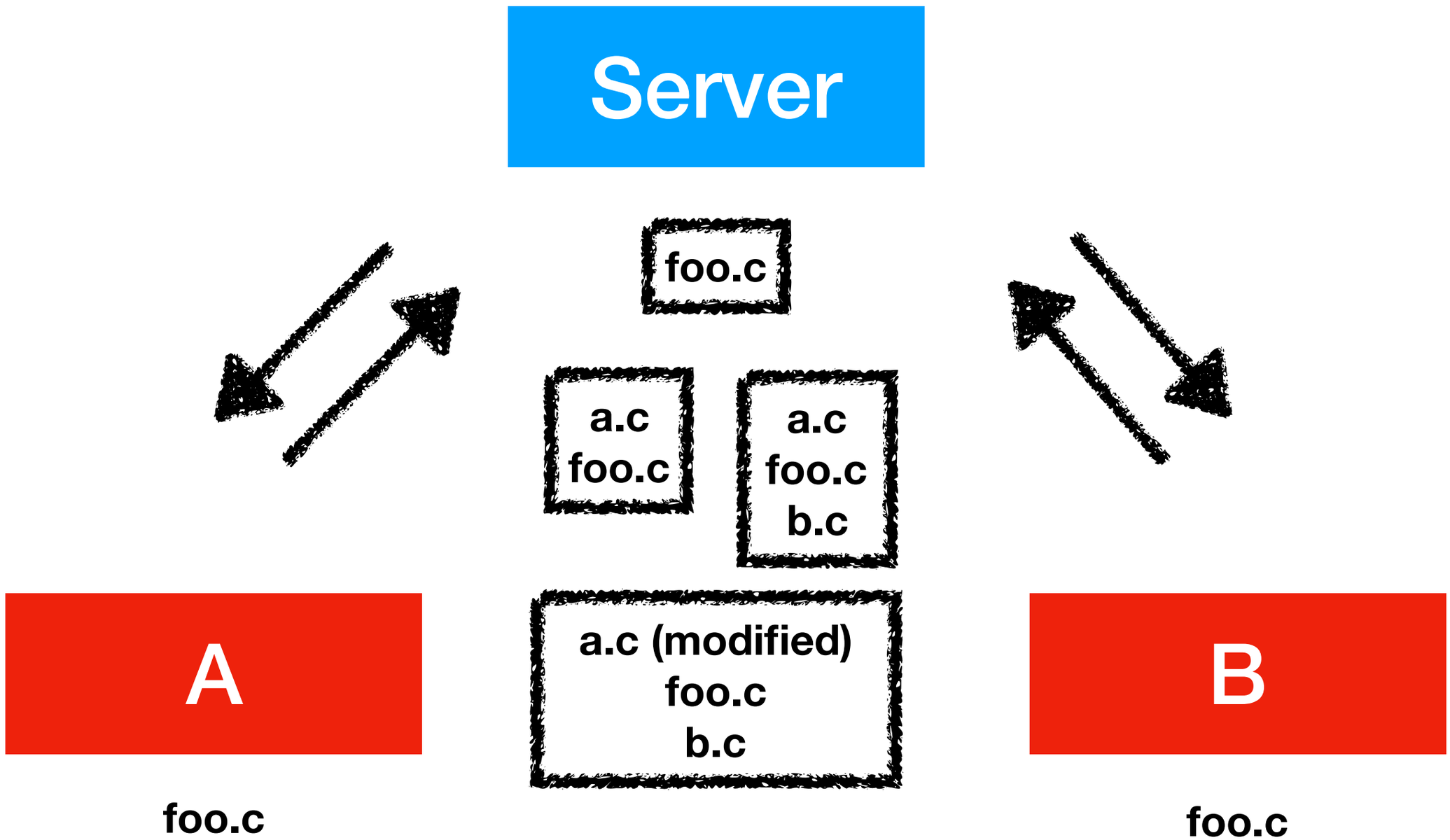
Version control system

- What?
  - Manages changes to documents, source files and other collections of information.
- Why?
  - Do you remember which source file you added/modified last week? Probably not.
  - Have you ever developed a project with other people? Coordinating programmers is hard.
- How?
  - CVS, SVN and GIT

Server/client version control system

- What?
  - A kind of version control system that puts all tracking metadata on a server. Clients can fetch/upload source files and information from the server.
- Why?
  - Strait-forward and easy to maintain.
  - Save space.

Version control system



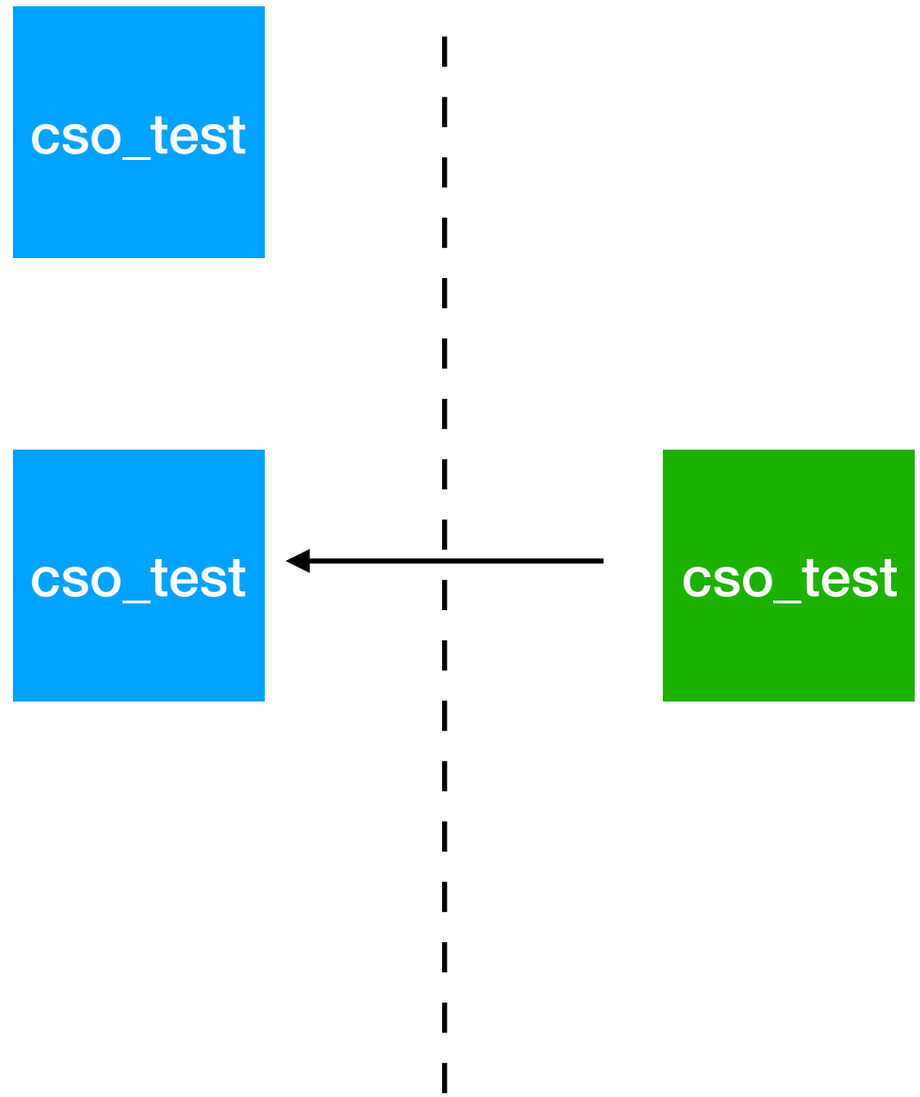
Distributed version control system

- What?
  - There is no “server”. Every client owns a complete repository locally (local repository) and can sync(push/pull) with any other remote repositories.
- Why?
  - There are hundreds or more projects and thousands or more developers in Linux community.
    - Coordinating the development using one single server is difficult.
  - Can work without network.

Git — initialization

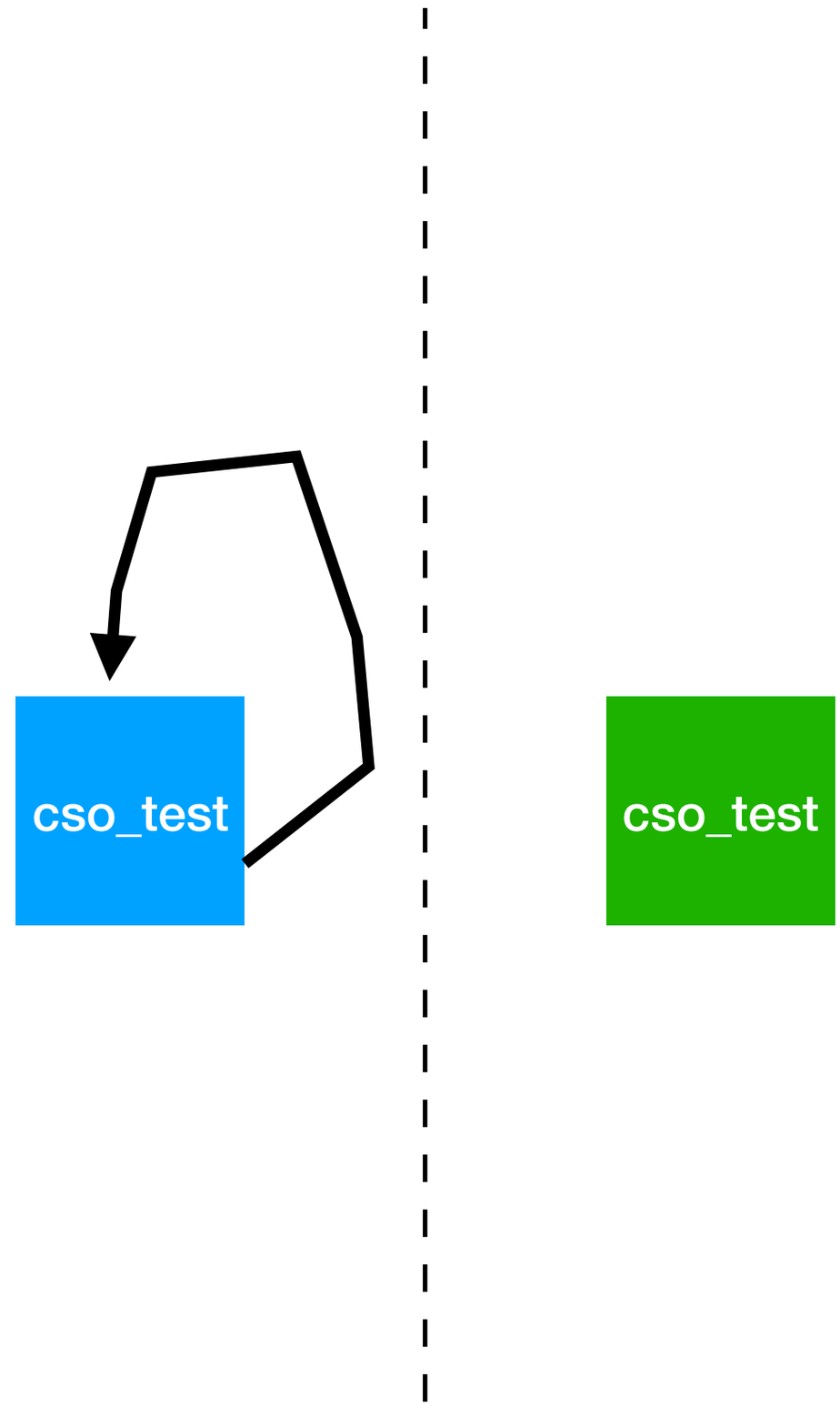
- git init

- git clone
  - git remote -v



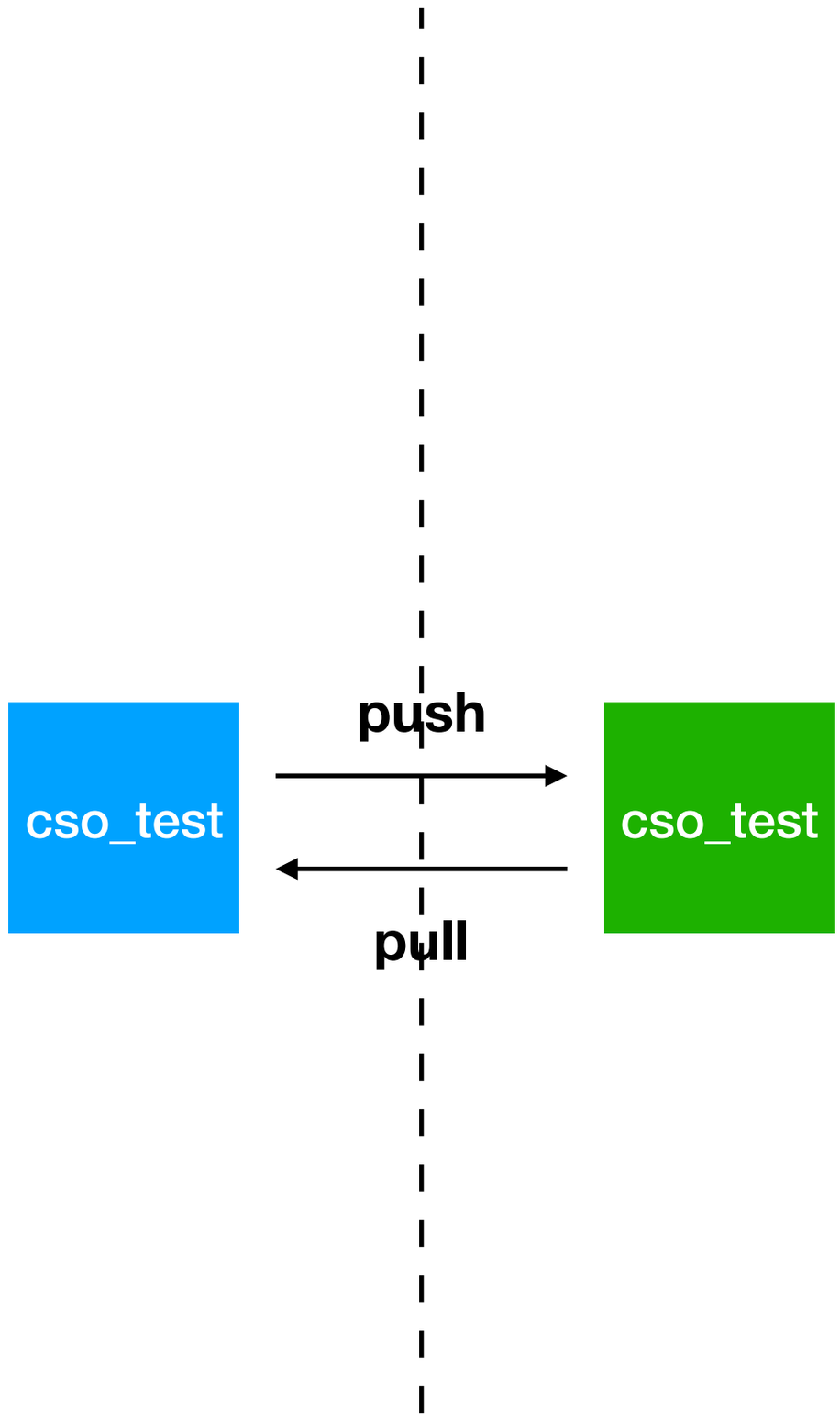
Git — commit

- git commit -m “comment”
- git add FILES
- git rm FILES
- git log

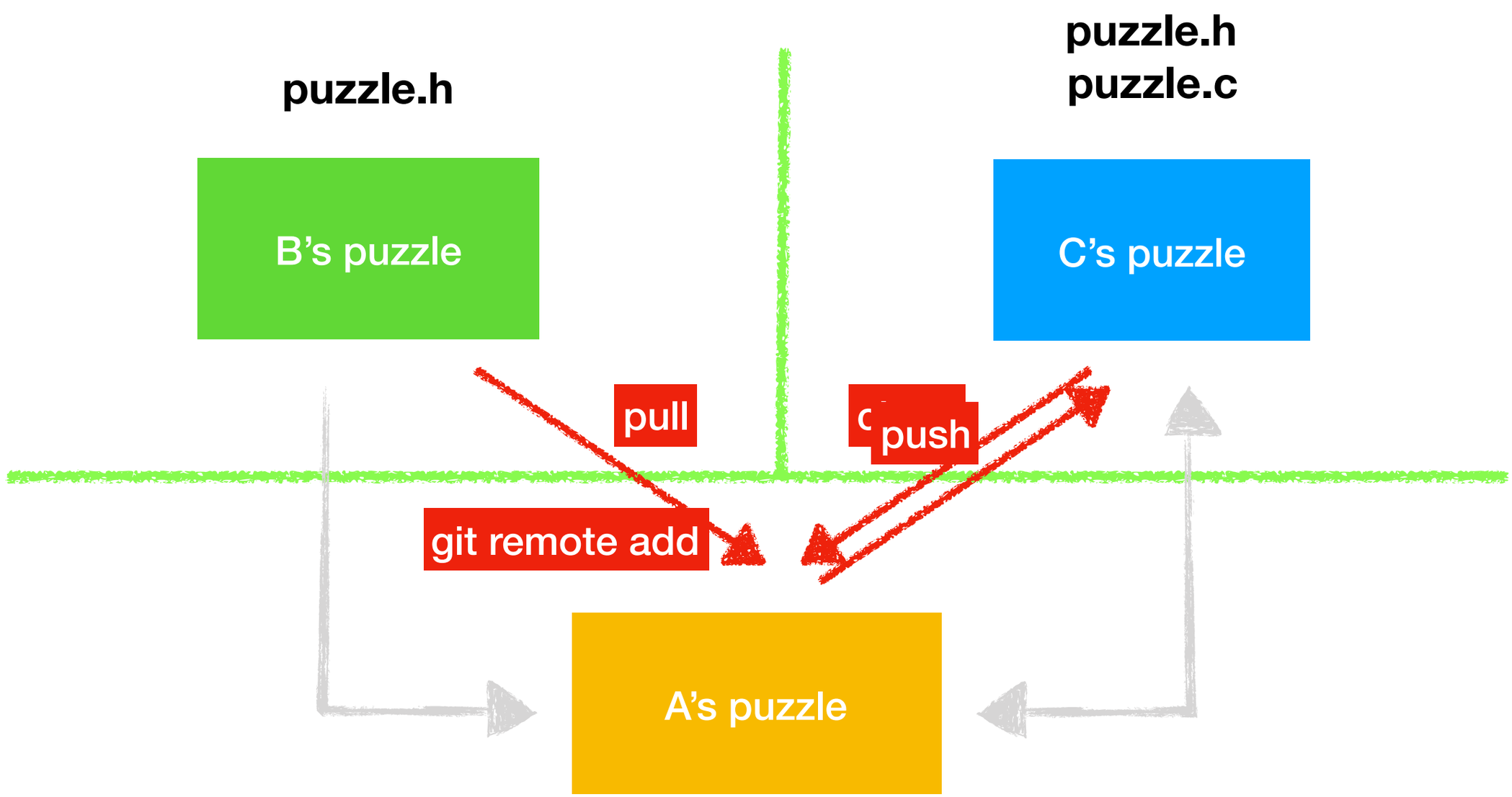


Git — remote

- git remote -v
- git push
  - git push origin master
- git pull
  - git pull origin master



Git status for our recitation and labs

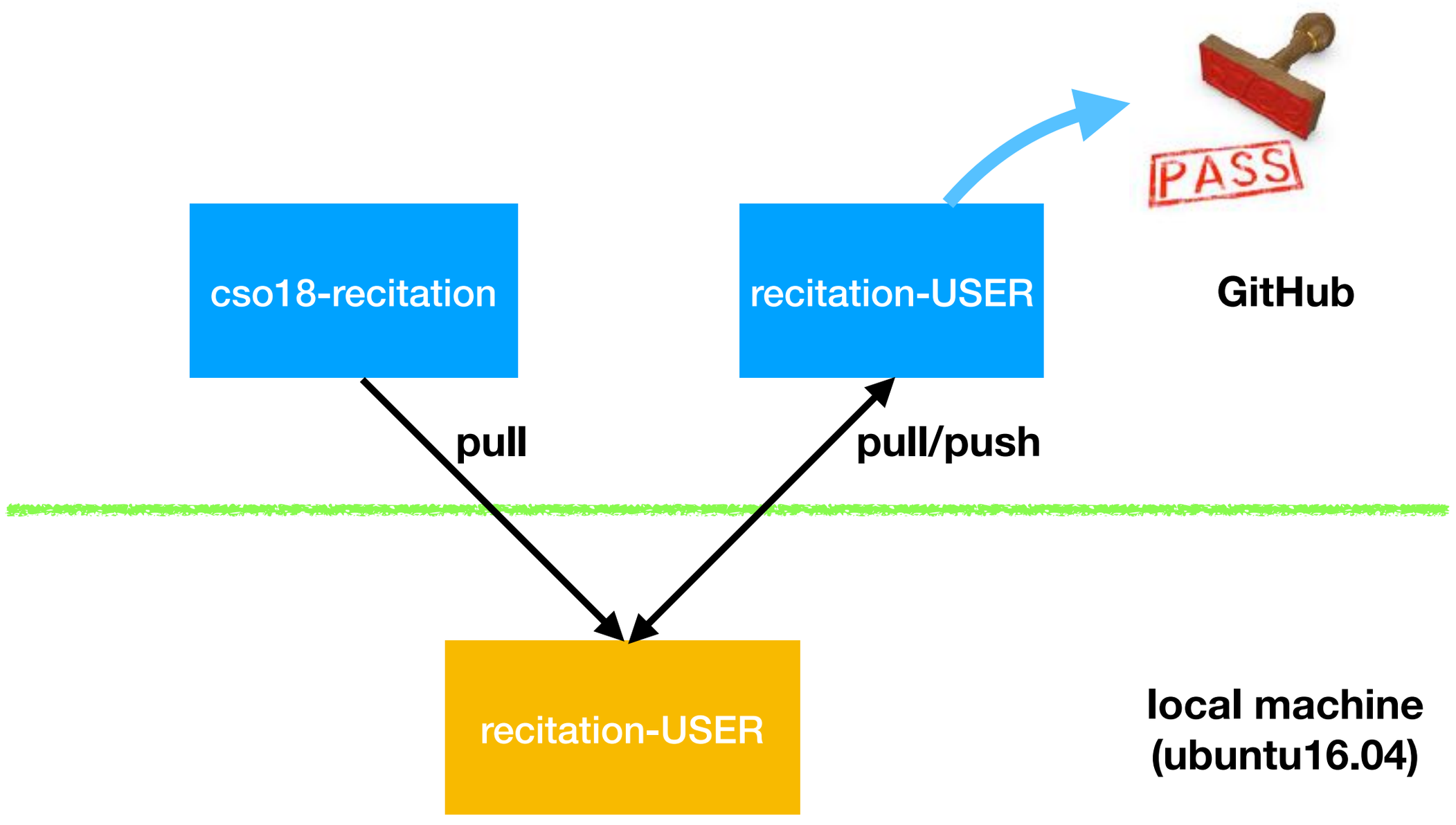


puzzle.h

puzzle.c

- https://github.com/nyu-cso18/cso18-recitation

Git status for our recitation and labs



Quiz Discussion Part1

1. 2, 4

2. 1, 2

3. 1, 4

4. 4

5. 3

Quiz Discussion Part2

1. 2

2. 1

3. 1

4. 3

5. 1, 5

Quiz Discussion Part3

1.
   - *s != ‘\0’
   - s++;
2. 3
3. Line 22: char**fields = malloc(sizeof(char*)*3);
4. Line 8: *s = ‘\0’;
5. 2 & 0

Quiz Discussion Part4

1.

- unsigned int*
- unsigned

2. 1

3. if array[i] >= result { result = array[i]; }

4. 7