

# Computer Systems Organization Recitation CSCI-UA 0201-007

R01: Introduction & environment setup

Many slides based on Chien-Chin Huang's Spring 2018 CSO recitation and Lingfan Yu's Spring 2019 CSO recitation

# Before we get started...

- Go to <https://nyu-cso.github.io/labs> and start the download for the version of VirtualBox relevant to you AND also start the download for the class image

# Logistics

Important things you should know

# What is this recitation for?

- Review course contents, including but not limited to:
  - Exercises to help you understand course contents better
  - Tutorials of labs
  - Quiz preparation and review
- Make us all suffer by forcing us out of bed at 6am

# Where we release course materials

- Course Website
  - <https://nyu-cso.github.io/index.html>
  - Recitation slides also on the course schedule page
- Piazza
  - It's your responsibility to read Instructor's Note on Piazza
  - You are encouraged to ask questions on Piazza
- GitHub
  - All labs and recitations are released on GitHub
  - You will submit all labs and recitation assignments on GitHub

# How to contact us

- If you have general questions about course contents or labs / recitations
  - Ask on Piazza
  - Come to office hours
- If you want to send us a private message
  - Email cso-staff mailing list at [cso-staff@cs.nyu.edu](mailto:cso-staff@cs.nyu.edu)
    - ▶ Include your name, your GitHub username, and your NYU NetID

# How are we going to proceed?

- For the first two weeks, we will focus on environment setups, usage of basic tools, etc.
  - Today we will cover environment related setups
  - Next recitation will cover programming tools such as gcc, gdb, and make
- Problem driven starting from the third week
  - We will go through recitation exercises together
- Recitation Exercises will be due Thursday evenings
  - An automated script will pull submissions Friday one minute after midnight
  - Please submit early, hours before the deadline

# Academic Integrity

- All work must be your own - do not copy or even look at assignments done by others
  - Don't ask StackOverflow or Chegg for help - if you need it, ask us!
  - Don't hire someone to do your work for you
- We reserve the right to use software plagiarism detection tools such as Moss
- It's not worth the risk, just don't cheat and make me sad

# Getting Started

Important things you must do

# Today's Topics

- Setting up your virtual machine
- Setting up your git repositories
- Basic Unix commands
- Program development
  - Editor (Sublime)
  - Version control (Git)

# Today's Goal

- By the end of today's recitation, you should
  - Have the class virtual machine installed
  - Have GitHub ready for you to submit work
    - An account
    - Recitation repo
    - Lab 1 repo
- Homework/Exercise today
  - Prove you can submit assignments by modifying r01/README.md in your recitation repository

# Setup Github

- Create a GitHub account if you don't have one
- Enroll yourself in the GitHub classroom
  - Create your recitation repository by clicking the link below
    - ▶ <https://classroom.github.com/a/tLJel9JG>
    - ▶ Select your NYU NetID
      - Very important!
      - ▶ Don't select someone else's NetID!
- If you cannot find your NetID, let me know!
- Now create your labs repository by clicking the link below
  - <https://classroom.github.com/a/EMTSXqqS>

# Basic virtual machine setup

- Follow <https://nyu-cso.github.io/labs/> instructions to
  - Download VirtualBox 6.0.10
  - Download our VirtualBox image
  - Launch VirtualBox and import the image
  - Launch Lubuntu Linux
    - ▶ Username “lab”, password “lab12345”

# Advanced VM setup

- After finishing the basic setup, you are good to go
- But if you want to
  - Resize the VM window to full screen nicely
  - Copy and paste between the VM and your laptop
  - Move files between my laptop and the VM
- Then check out the Lab instruction page!
  - <https://nyu-cso.github.io/labs/>
  - I recommend it!

# Attention: You **MUST** test your code in your class virtual machine

- Throughout the semester, all your labs and recitations will be graded inside the class virtual machine we provide to you
- You may do your assignments in your native OS, **but you must test them inside the virtual machine**
- If you fail to do so, you may be penalized if the code doesn't work correctly within the VM!

# Open up a terminal

- Click start icon (the bottom left icon)
  - Click “System Tools” and then “LXterminal”
- OR use the keyboard shortcut
  - Ctrl + Alt + T
- OR use the shortcut on the desktop
- To copy paste in a terminal, you need to use
  - Ctrl + Shift + C to copy
  - Ctrl + Shift + V to paste
  - Or just right click

# Basic Commands

- Some useful commands to know:
  - man
  - ls, cd, pwd, mkdir
  - cp, mv, rm
  - echo, cat
  - wc
  - grep
  - ctrl-c, ctrl-d, ctrl-z, fg, bg
  - |, >, <, >>
  - apt install/search
  - history, ctrl-r

# Basic Commands

- Whenever you want to find out how to do something using command line, ask google first
- Here is a link contains useful command, for both beginners and experienced users:
  - <https://github.com/jlevy/the-art-of-command-line>

# Editor

- You need a good editor to code with for productivity
- Popular editors used by programmers:
  - vim
  - emacs
  - **sublime**
- We recommend you use Sublime Text
  - Which should be already installed on the VM image

# Version Control

- What is version control?
  - Manages changes to documents, source files and other collections of information
- Why is version control indispensable?
  - Track code changes
  - Roll back to older version
  - Collaborate with others
- We are going to use the popular “Git” as our version control system

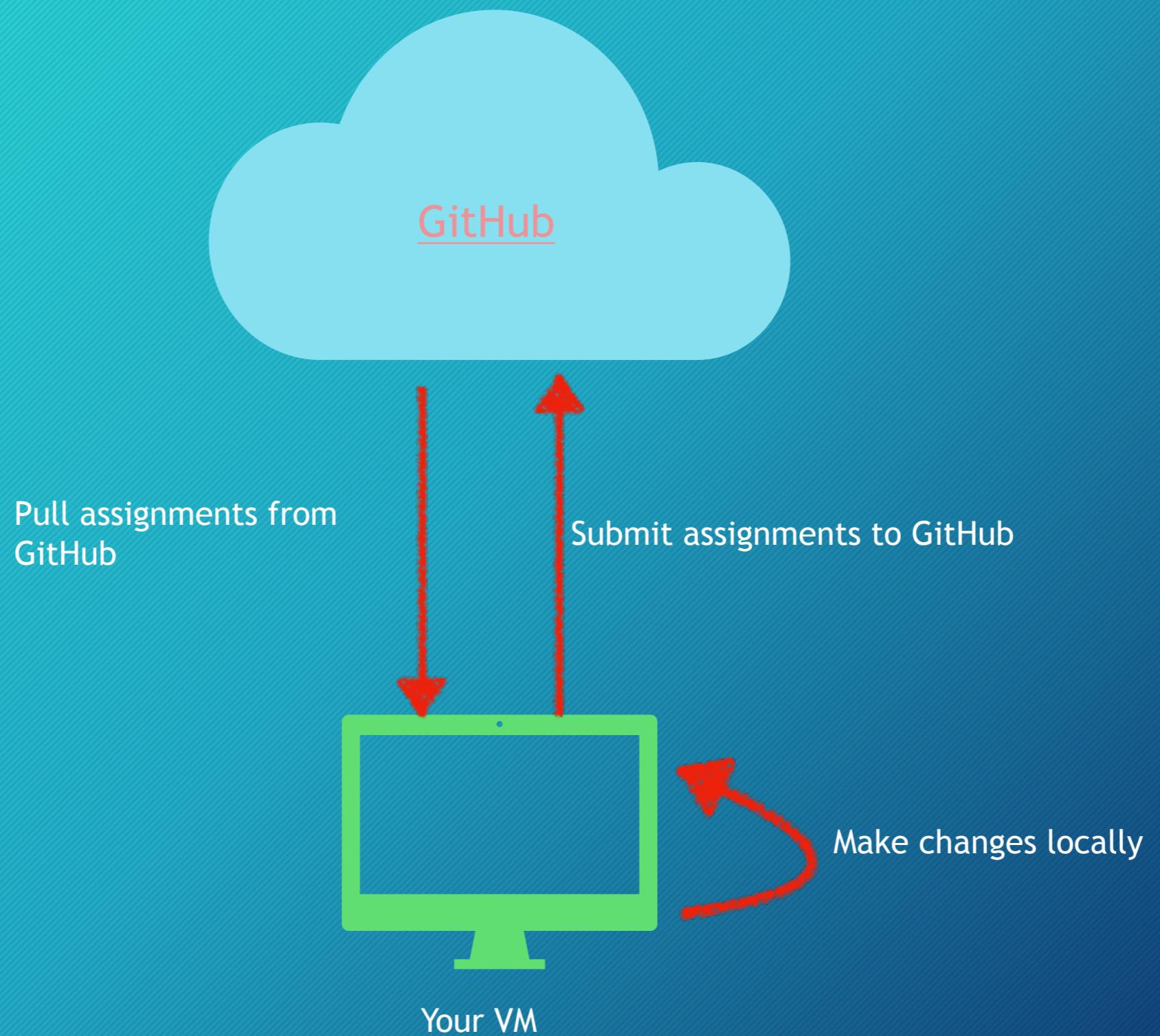
# A list of git commands you need

- `git clone`
- `git status`
- `git remote`
- `git add <file name>`
- `git commit -m <commit messages>`
- `git push origin master`
- `git pull upstream master`

# You need to config git first!

- **git config --global user.email “<Your Email>”**
- **git config --global user.name "<Your Name>"**
- **You can issue “git config --list” to check your configuration**

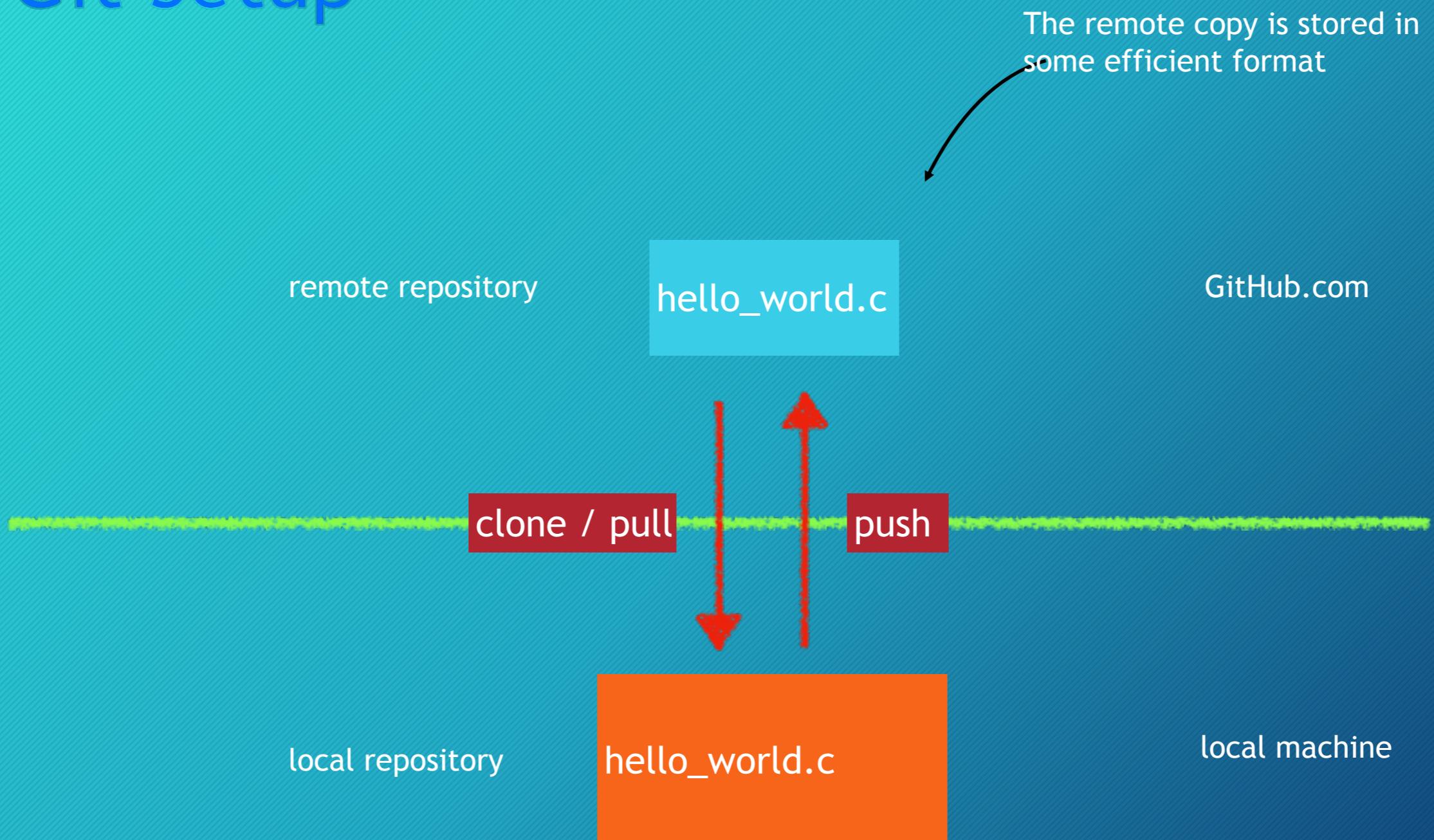
# Git Overview



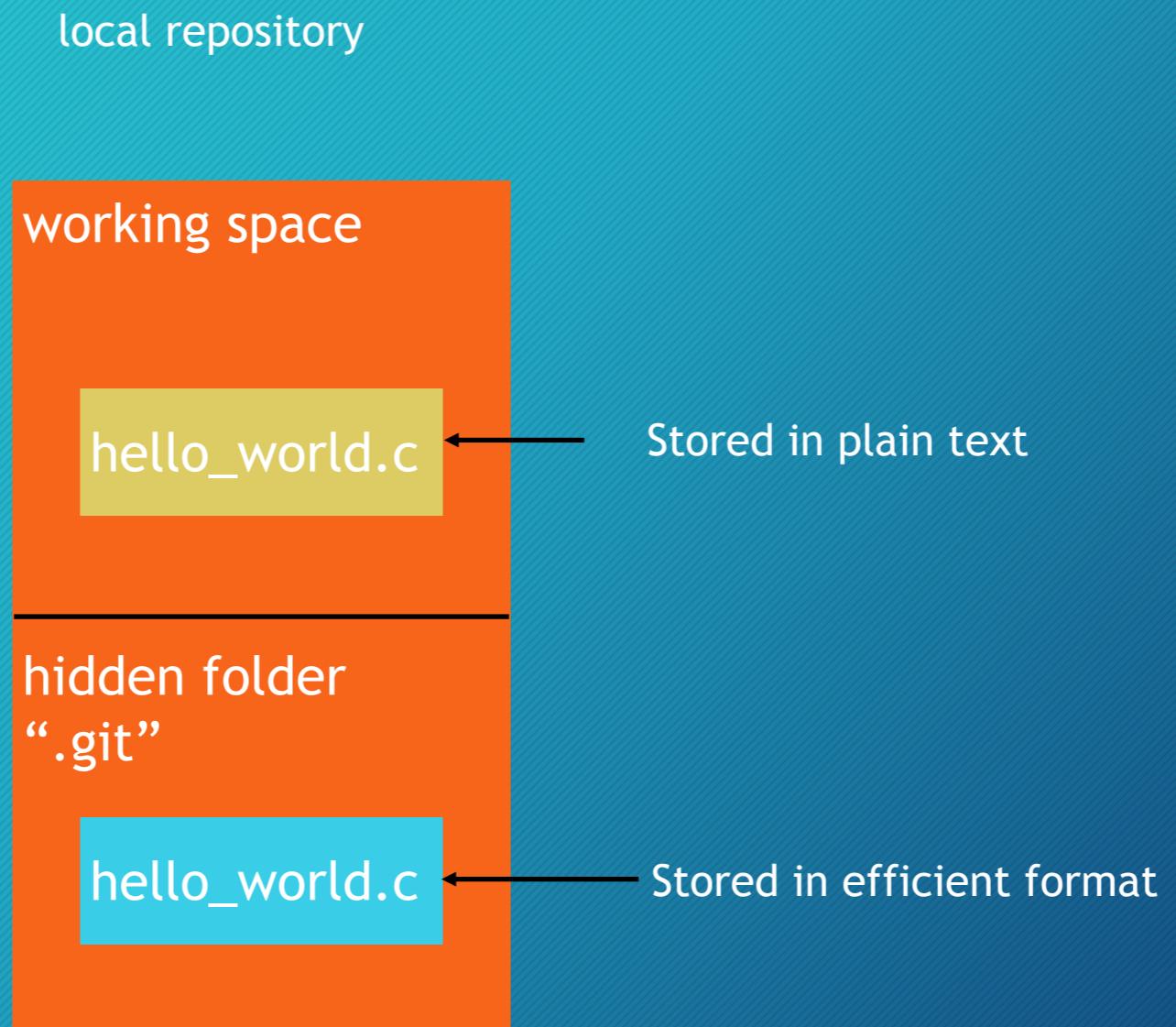
# Setting up your recitation repo

- In command line, type:
  - `git clone https://github.com/nyu-cso-sp19/recitations-“Your GitHub Username”`
    - ▶ If you copy the above command to command line, don’t let the line break
    - ▶ Replace “Your GitHub Username” (including the quote marks) with your GitHub username.
  - `cd recitations-“Your GitHub Username”`
- You only need to “git clone” once

# Git Setup

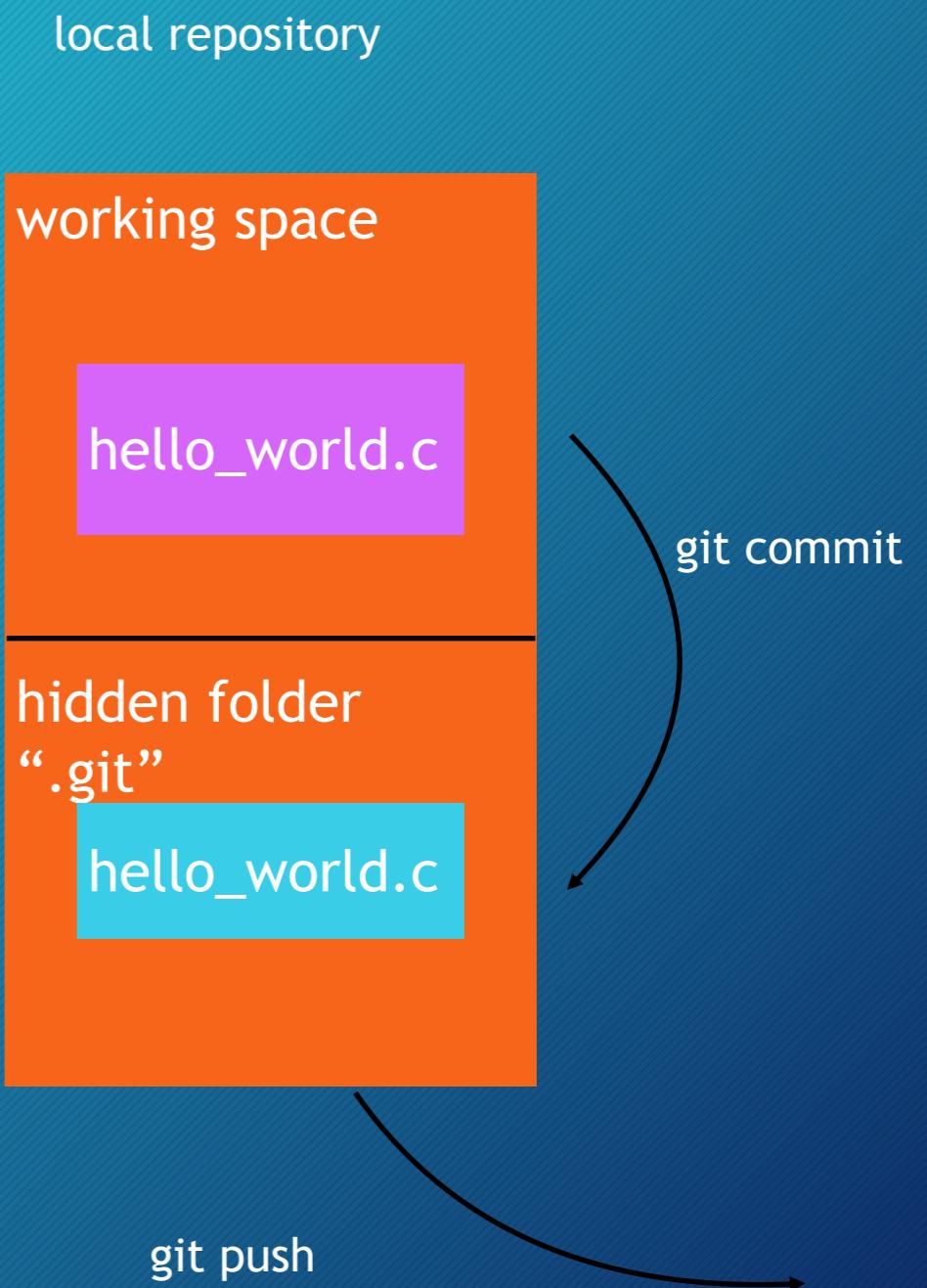


# A closer look at your local repository



# How to interact with Git

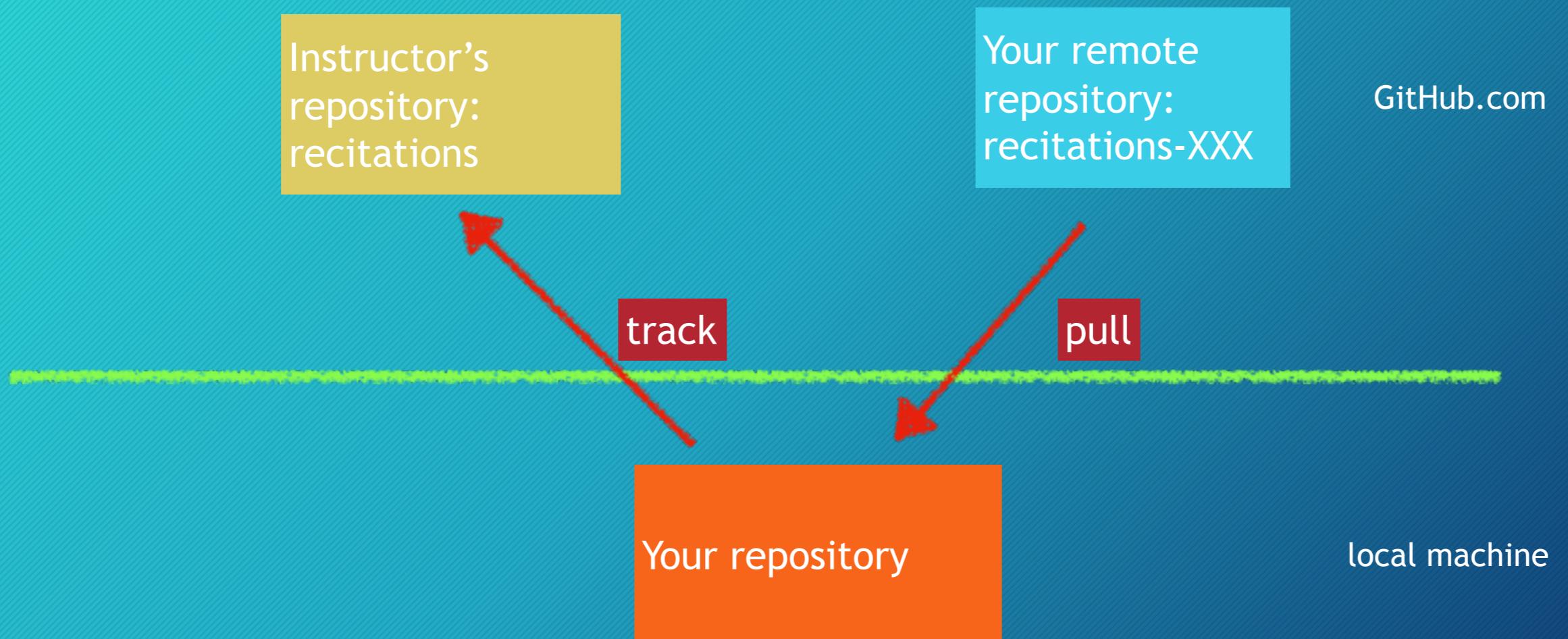
- `git add hello_world.c`
  - Tell git to track changes to `hello_world.c`
- `git commit`
  - Store tracked file to `.git`
- `git push`
  - Submit commits to your remote repository



## It's not the end of story

- What if an instructor posts a new lab/recitation
- Where can I find it?
- How can I download it into my repo?

# Lab/Recitation Setup



# Track your instructor's repository

- In command line, type:
  - `git remote add upstream https://github.com/nyu-cso-fa19/recitations`
    - ▶ Don't let the line break for above command when copy paste to terminal
- You only need to do this once

# For each new recitation assignment

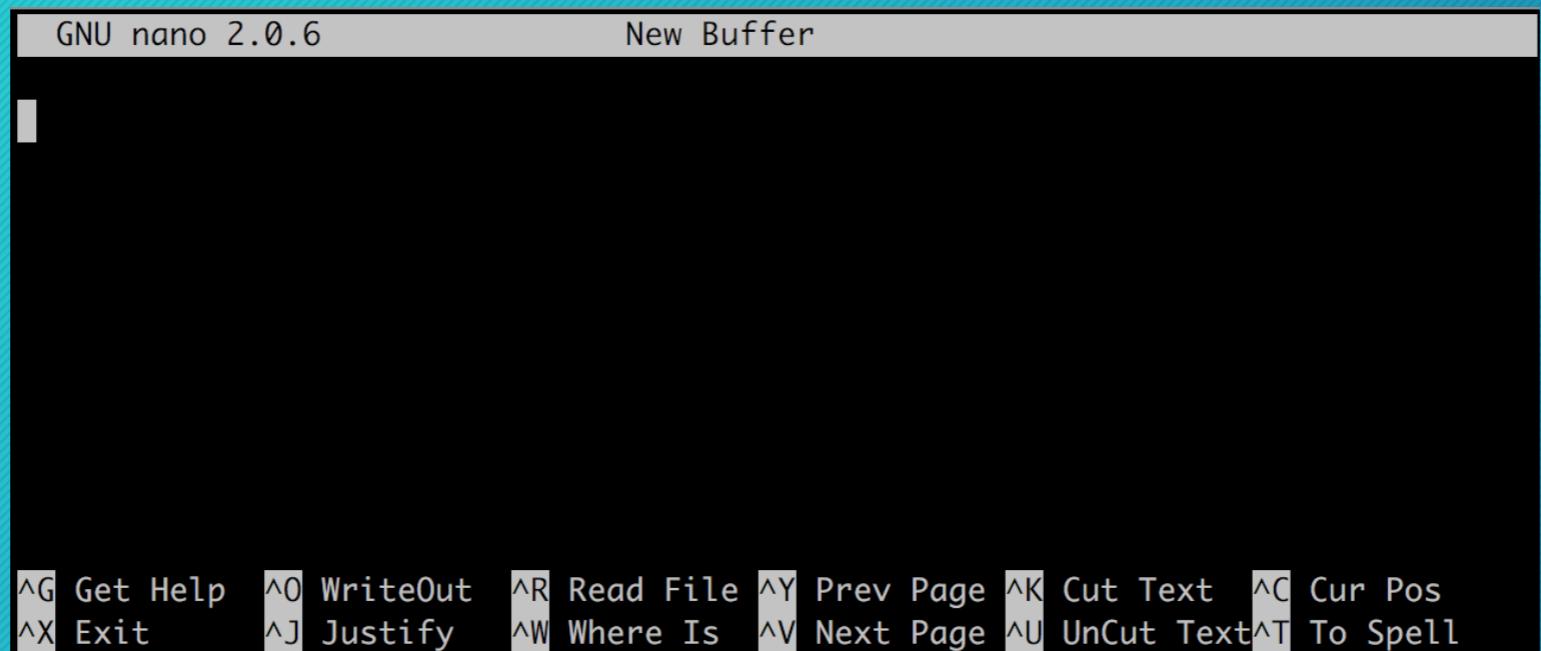
- Pull latest assignment
  - `git pull upstream master`
- Then make changes locally in the VM
- Tell git to track changes
  - `git add “file name”`
- Commit your changes
  - `git commit -m “commit messages”`
- Submit to your remote repository (on GitHub)
  - `git push origin master`

# Git commit

- When you issue “git commit”, you need to provide a message which is a short description of the changes you made
- You can use “-m” option to provide the commit message
  - `git commit -m "my first commit"`
- If you don’t use “-m” option, a command line editor will pop up for you to edit the commit message
  - By default, nano

# How to get out of Nano Editor

- The default editor is called Nano.



- To add a commit message from nano
  - First type in some commit message
  - Hit Ctrl + O to save your commit message (^ means Ctrl)
  - Hit Ctrl + X to exit

## Double check with “git status”

- Sometimes, you might forget to do some (or all) of
  - git add, git commit, git push
- It's always good to check the status of your repository
- git status tells you
  - What files are going to commit
  - What files are not tracked
  - Whether you forgot to push commits to remote

# Triple check with GitHub

- Still not sure/confident about whether assignment was submitted properly?
- Go to [github.com](https://github.com), then go to your repo
- Manually check if every file contains the up-to-date information

# Git is much more powerful than that

- Our git introduction only covers a small part of Git
- Git tutorial:
  - <https://www.atlassian.com/git/tutorials/what-is-version-control>
  - <https://try.github.io/levels/1/challenges/1>

# All the git commands you need for CSO

- For beginners, it's super easy to mess up Git
- After setting recitation and lab repository, you ONLY need to use the following git commands:
  - `git add filenames`
  - `git commit -m "commit message"`
  - `git push origin master`
  - `git pull upstream master`
  - `git status`

Warning: unless you know what you are doing, do not use any other git commands or git command flags

# Ask the staff for help

- If you really cannot fix conflicts or other git problems, you should ask course staff for help
  - You need to email the staff to make an appointment
    - ▶ Or go to office hours
  - You should start your lab earlier
- Don't randomly issue commands to further mess things up

# Things you should Never do

- Don't use `git add *`, `git add .`
  - instead, you should always specify the file names you want to commit
  - Please don't add compiled programs to git
- Don't modify any file using GitHub website
  - Instead, you should always make changes locally on our laptop and then push commits to GitHub
  - Otherwise, there will be conflicts, which will lead to sadness

# Exercise to submit for today

- Read and sign the README file in r01
- Submit your cheat sheet to GitHub by
  - `git add README.md`
  - `git commit -m "I did my first assignment"`
  - `git push origin master`
- Due tonight