git and .gitignore

Hui Chen

Graduate Center
The City University of New York

February 14, 2019

Using git

A simple workflow:

1. Clone a remote repository
   `git clone FOO_REPO_URL`
2. Edit files
3. Stage the files edited
   `git add foo`
4. Commit the changes
   `git commit -m "edited foo"`
5. Repeat steps 2 – 4
6. Push the changes to the remote repository
   `git push`
7. Repeat steps 2 – 6

Discussion Question

What files should be in the repository? What should not be?

1. foo.c
2. Foo.java
3. foo.o
4. Foo.class

Discussion Question

What files should be in the repository? What should not be?

1. paper.tex
2. paper.bib
3. acmart.cls
4. paper.aux
5. paper.log
6. ACM-Reference-Format.bst

How about “git add .“

What if we wish to track files in a directory conveniently?

gitignore and .gitignore

- A gitignore file specifies intentionally untracked files that Git should ignore. Files already tracked by Git are not affected
- .gitignore: per-directory settings
- See https://git-scm.com/docs/gitignore

gitignore Templates

- Github maintains a list of common gitignore templates at https://github.com/github/gitignore
- Example: https://github.com/github/gitignore/blob/master/TeX.gitignore

Clean repository based on .gitignore

We can clean up a repository based on .gitignore, i.e., untrack unnecessary files that was put in the repository

1. Create and edit .gitignore to current directory
2. untrack in the directory files that should be ignored according to .gitignore
   ```
   git rm -r --cached .
   ```
3. add everything to stage the change
   ```
   git add .
   ```
4. commit the changes
   ```
   git commit -m "untracked unnecessary files"
   ```
5. push the changes
   ```
   git push
   ```