ECE271: Microcomputer Architecture and Applications — University of Maine

Gitlab Setup/Usage

by Yifeng Zhu
modified by Vince Weaver
30 January 2019

Background

- We will submit our labs in ECE271 via git to the department gitlab server.

- The first part of this document describes how to set up a GUI git interface on Windows which has been used in ECE271 in the past.

- If you are on Linux or MacOSX you might prefer to use the plain git command lines tools. Directions on this are at the end of this document. It is also possible to use the command line version of git under Windows if you are so inclined.

- Git is a widely used version control system for source code maintenance. There is a free online book about Git here: `http://git.scm.com/book/en/v2`. Chapter 2 gives a rundown of basic Git commands.

Common Errors – TL;DR

- When a git clone does not work properly on Windows because it is asking for your password, you need to have a key provided by your computer. Be sure Pageant is running with your key loaded.

Workflow Overview

Basic workflow:

- **Clone** a repository
- Write your code
- **Add** the files you have changed
- **Commit** your changes and give a commit message
- **Push** all your commits to your server

Best Practices

- Commit early, commit small, commit often
  This is especially important when working on a group project as changes to the same file can lead to a dreaded “merge conflict”

- Always commit before you do a pull

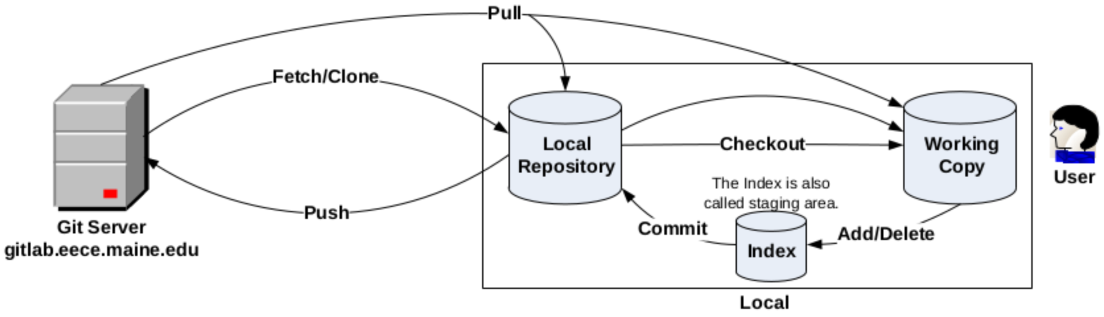


Figure 1: Diagram of what git is doing. Don't worry, for some reason most diagrams about how git work are fairly incomprehensible.

- On group projects, always pull to check for new changes before you push your changes
- Do not commit files that can be regenerated from source.
  For example, there's no need to commit .o files, executable files, or anything that is autogenerated at build time.
- Be sure to choose meaningful commit messages.

Setting up your Gitlab account

- Log into the gitlab server `https://gitlab.eece.maine.edu`. Use the LDAP login option, with your MaineStreet username and password.
- You need to log in once to get into the system so the TAs can properly add your account to the various ECE271 groups.

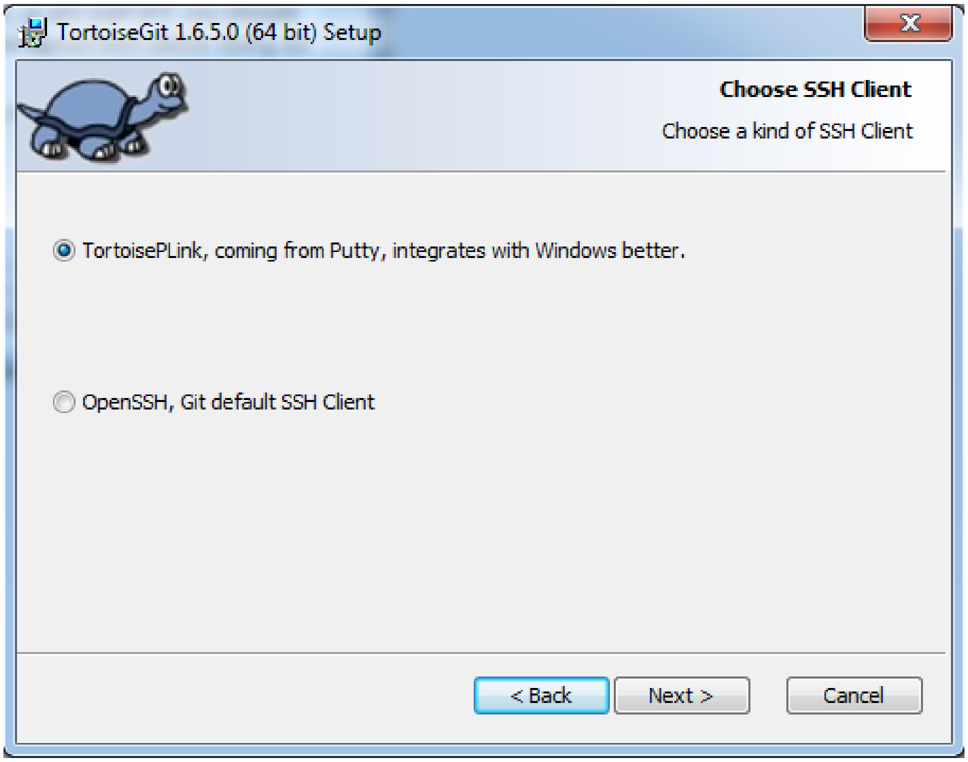
Setting up Git on Windows

1. Install git for Windows
   (a) Download the latest git for windows from `http://git-scm.com/download/win`
   (b) Just choose the default answers during the installation (one thing you might want to change is the default editor to not be vim, but that only matters if you plan to not use the GUI)

2. Install TortoiseGit

   (a) Download it from `https://code.google.com/p/tortoisegit/`. (Make sure you select the right one for your computer, which is probably the 64-bit version).

   (b) During installation use the default settings
   Especially be sure you choose the default “TortoisePLink” not OpenSSH



TortoiseGit 1.6.5.0 (64 bit) Setup
Choose SSH Client
Choose a kind of SSH Client
< Back
Next >
Cancel

● TortoisePLink, coming from Putty, integrates with Windows better.

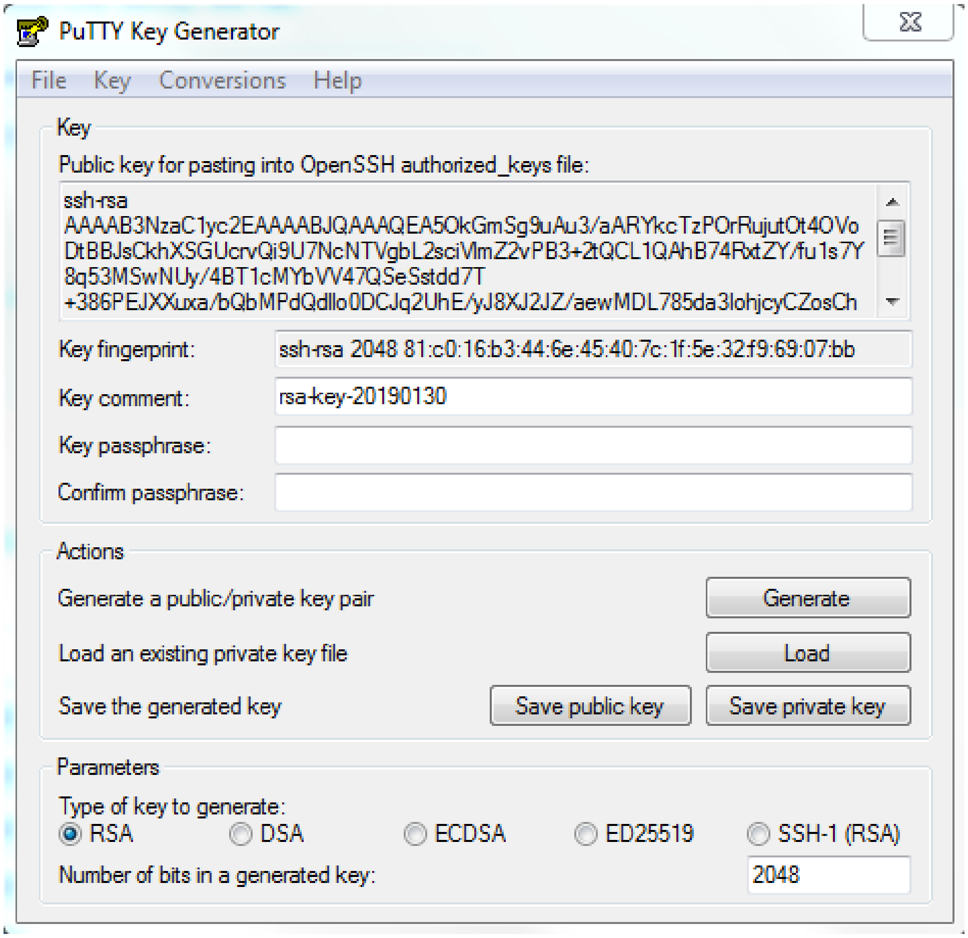
○ OpenSSH, Git default SSH Client

Figure 2: Proper ssh client choice in TortoiseGit.

3. Generate SSH Key Pair (Public and Private Keys)

   (a) Run PuTTYgen (If you use Windows 8, PuTTYgen may be blocked, click ‘more info’ then ‘Run’)

      i. Click Windows Start, All Programs, TortoiseGit, PuTTYgen
      ii. Select the type of key to generate (RSA)
      iii. Click ‘Generate’ and move your mouse randomly to generate a key.
      iv. Once you have generated a key, put your umaine e-mail address in the comment area.
      v. For extra security pick a key passphrase.
      vi. Click ‘Save private key’ and save it somewhere you won’t lose it.
      vii. Click ‘Save public key’ and save it to the same place
      NOTE: don’t close this window yet, we’ll want to cut-and paste this key at a future step.



Key

Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAQEA5OkGmSg9uAu3/aARYkcTzPOrRujutOt4OVo
DtBBJsCkhXSGUcrvQi9U7NcNTVgbL2sciVlmZ2vPB3+2tQCL1QAhB74RxtZY/fu1s7Y
8q53MSwNUy/4BT1cMYbVV47QSeSstdd7T
+386PEJXXuxa/bQbMPdQdllo0DCJq2UhE/yJ8XJ2JZ/aewMDL785da3lohjcyCZosCh
```

Key fingerprint: ssh-rsa 2048 81:c0:16:b3:44:6e:45:40:7c:1f:5e:32:f9:69:07:bb

Key comment: rsa-key-20190130

Key passphrase:

Confirm passphrase:

Actions

Generate a public/private key pair

Load an existing private key file

Save the generated key

Parameters

Type of key to generate:
● RSA ○ DSA ○ ECDSA ○ ED25519 ○ SSH-1 (RSA)

Number of bits in a generated key: 2048

Figure 3: PuTTYgen output.

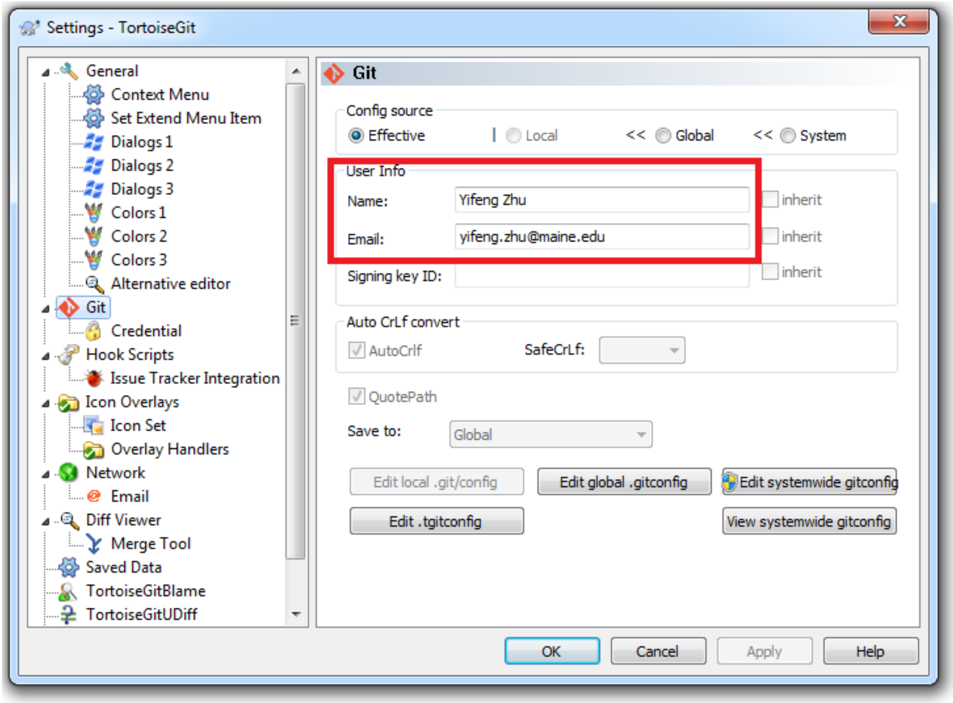
(b) Run Pageant (an SSH authentication agent)

i. Click Windows Start, All Programs, TortoiseGit, Pageant
ii. When running an icon will appear in the icon tray.
iii. Right click the icon in the tray and select “add key”
iv. Browse to the private key you selected and select it.
v. Enter your password when prompted

(c) Log in to gitlab `https://gitlab.eece.maine.edu` and add your **PUBLIC** key (not your private one).

i. Click the down arrow on the far top right to access your profile info, and click on “settings”.
ii. Click the person icon along the left side of the screen.
iii. Select the top, skinny key icon (should say “ssh keys” if you hover over it).
iv. Cut and paste the key from earlier here. (You can’t just load from the public key you saved for annoying reasons, the format is slightly different). Make sure it is your **PUBLIC** key not your private one.

4. Set up Git User and e-mail



Settings - TortoiseGit

Git

Config source

● Effective | ○ Local << ○ Global << ○ System

User Info

Name: Yifeng Zhu □ inherit

Email: yifeng.zhu@maine.edu □ inherit

Signing key ID: □ inherit

Auto CrLf convert

☑ AutoCrlf SafeCrLf:

☑ QuotePath

Save to: Global

Figure 4: Entering e-mail.

(a) You might have done this already when installing TortoiseGit, but if not you will put this info in. In git all of your commits are tagged with your name and e-mail.

(b) Right click and select "TortoiseGit" then "Settings"

(c) Enter your name and umaine e-mail address.

Checking out the Course Repository

1. Pick a location on your hard drive where you want to have the git tree located. Navigate there with Windows explorer, then right-click and choose “git clone”.

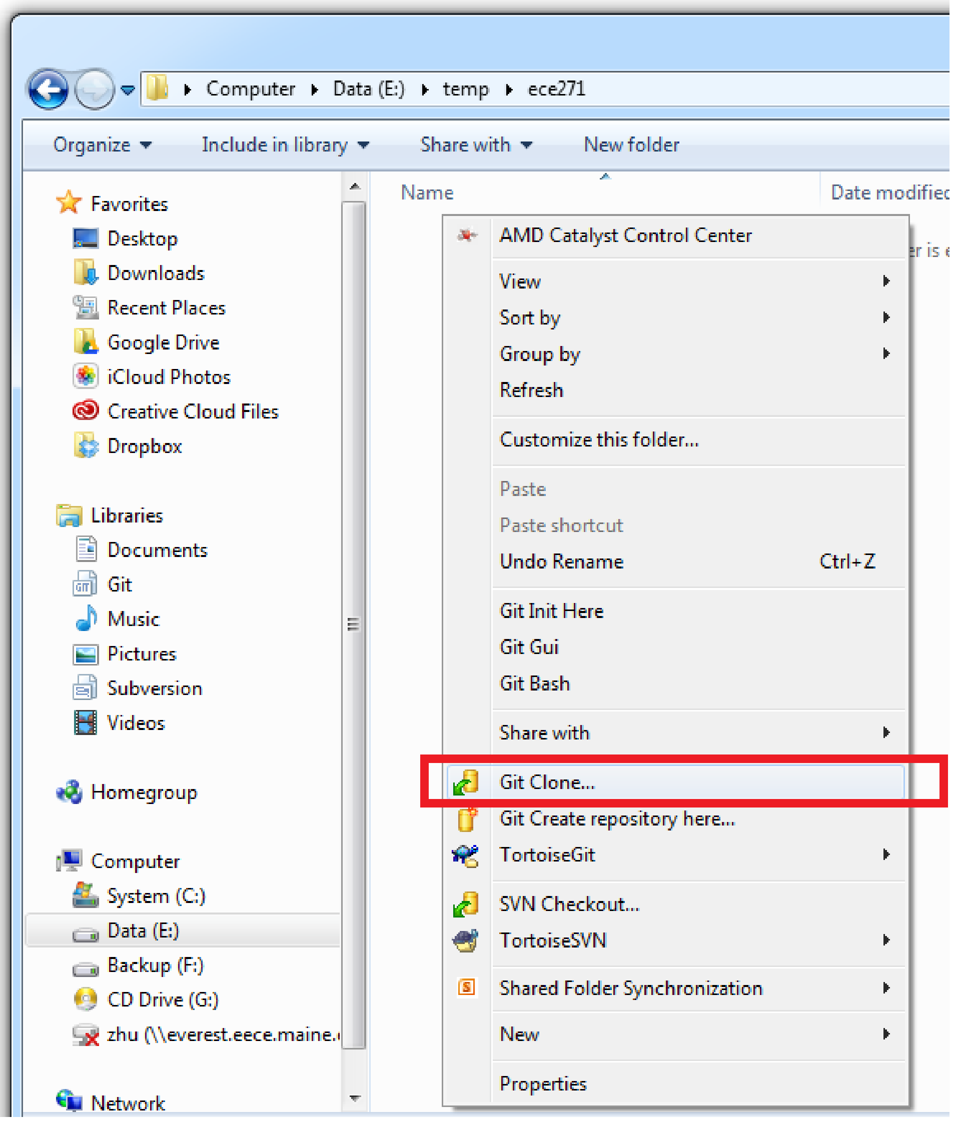


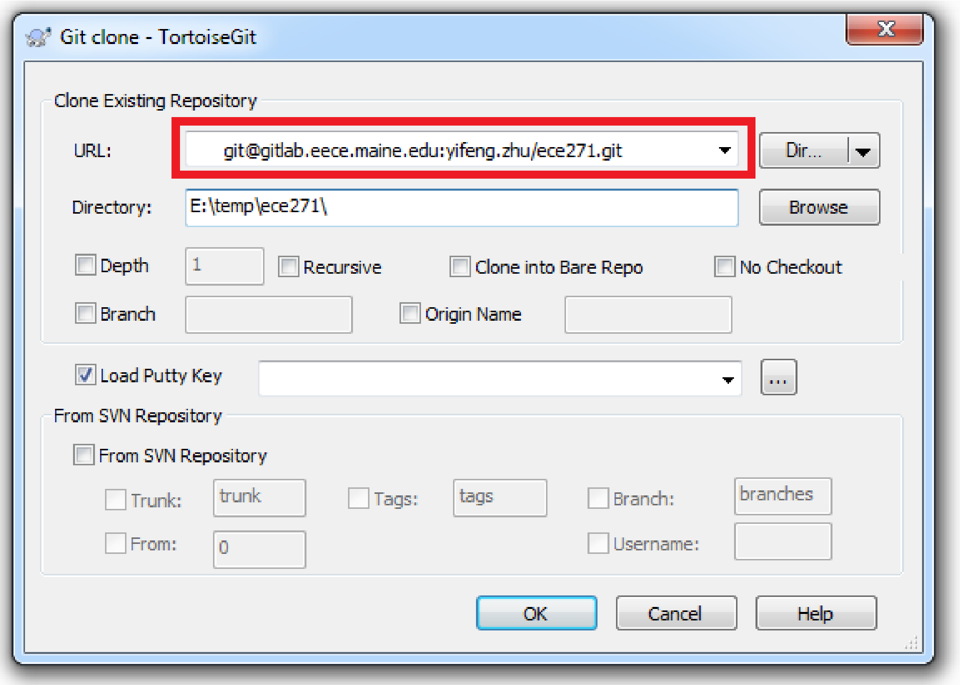
Figure 5: Cloning under windows.

2. Now cut-and paste the name of the repository for the class, it should be `git@gitlab.eece.maine.edu:vincent.weaver/ece271_2019s.git`

Submitting your Lab

1. Add a file in the cloned project directory.

2. Right click and select “git commit” then “master”



Git clone - TortoiseGit

Clone Existing Repository

URL: git@gitlab.eece.maine.edu:yifeng.zhu/ece271.git

Directory: E:\temp\ece271\

□ Depth 1 □ Recursive □ Clone into Bare Repo □ No Checkout

□ Branch □ Origin Name

☑ Load Putty Key

From SVN Repository

□ From SVN Repository

□ Trunk: trunk □ Tags: tags □ Branch: branches

□ From: 0 □ Username:

Figure 6: Entering the url.

3. Select files to be committed, then write your commit message in the box and click OK.

4. See Figure 7 for what things might look like.

5. If you have not done so already, create an “ece271” project on your personal gitlab webpage.

   (a) Log into gitlab

   (b) select “create new project”

   (c) call it “ece271”

   (d) make sure it is marked “private”

   (e) click create

6. Now push your modifications to your new gitlab project.
   First you have to change the destination to be yours.

   (a) Right click on main folder.

   (b) Select Tortoise.git

   (c) Select settings.

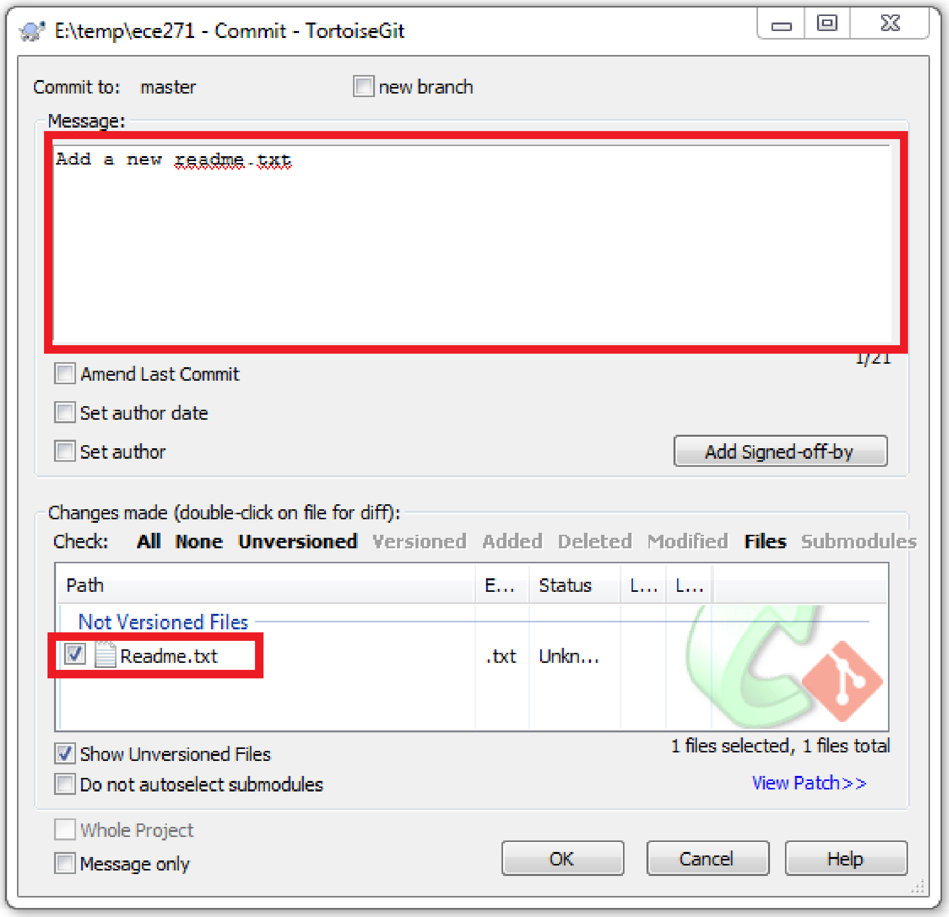
   (d) Select git.

   (e) Select remote.

   (f) Click on the word “origin” in the box under “Remote:”.

   (g) In the URL field, replace the URL with the URL of yours:
   `git@gitlab.eece.maine.edu:YOUR.NAME/ece271.git`

   (h) See Figure 8 for what things might look like.



E:\temp\ece271 - Commit - TortoiseGit

Commit to: master □ new branch

Message:

```
Add a new readme.txt
```

1/21

□ Amend Last Commit

□ Set author date

□ Set author

Changes made (double-click on file for diff):

Check: All None Unversioned Versioned Added Deleted Modified Files Submodules

Path	E...	Status	L...	L...
Not Versioned Files				
☑ Readme.txt	.txt	Unkn...		

☑ Show Unversioned Files

□ Do not autoselect submodules

1 files selected, 1 files total

□ Whole Project

□ Message only

Figure 7: Committing.

7. You will also need to add both me (vincent.weaver) and your TA to have “master” control on your repository.

   (a) Log into the gitlab webpage.

   (b) Click on your ece271 repository

   (c) On the left side, click on “settings”

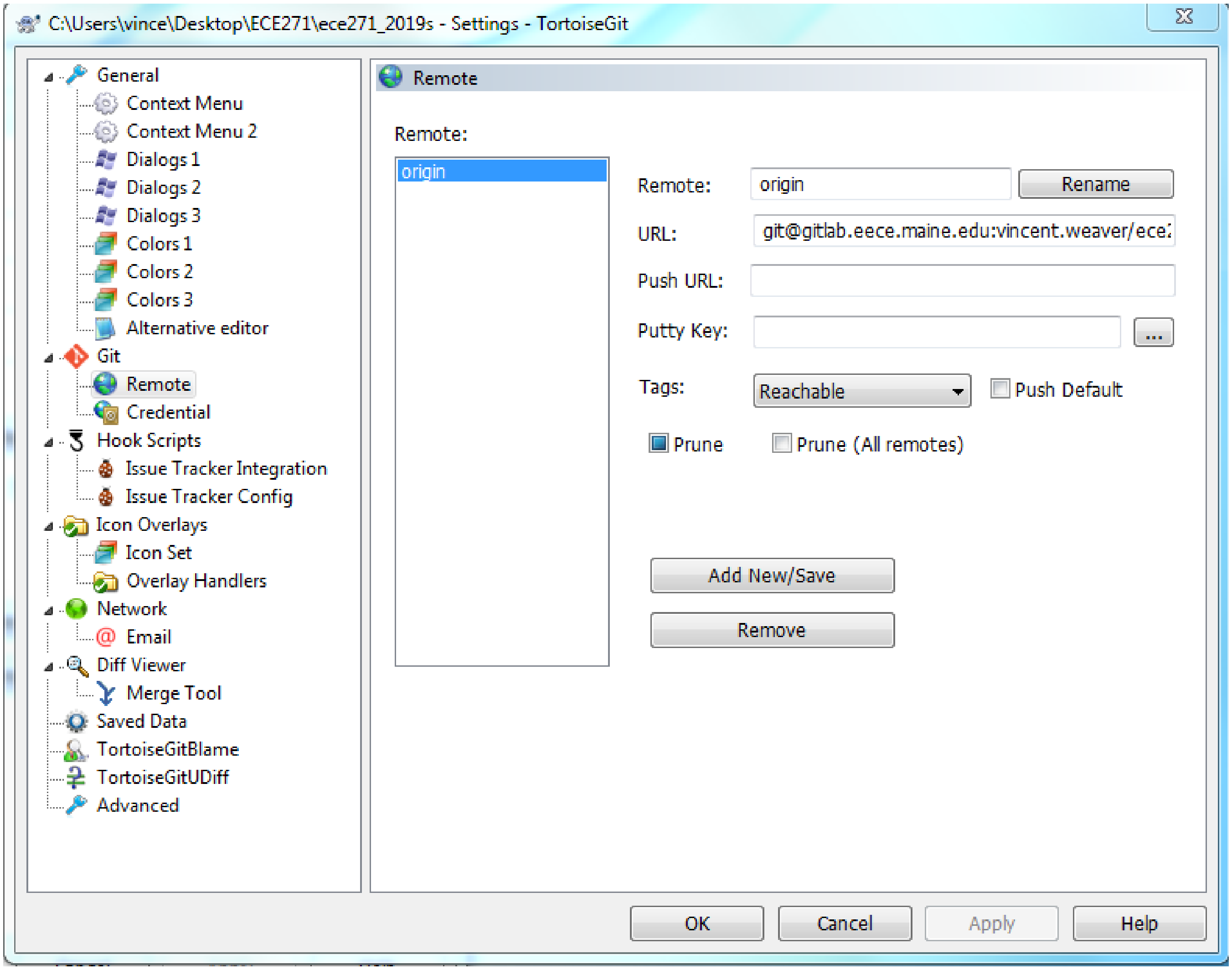
   (d) Then click on “Members”

   (e) Under “select member to invite” pick me or the TA

   (f) For role permission pick “master”

   (g) Then click “add to project”

   (h) See Figure 9 for what things might look like.



C:\Users\vince\Desktop\ECE271\ece271_2019s - Settings - TortoiseGit

Remote:

origin

Remote: origin

URL: git@gitlab.eece.maine.edu:vincent.weaver/ece2

Push URL:

Putty Key:

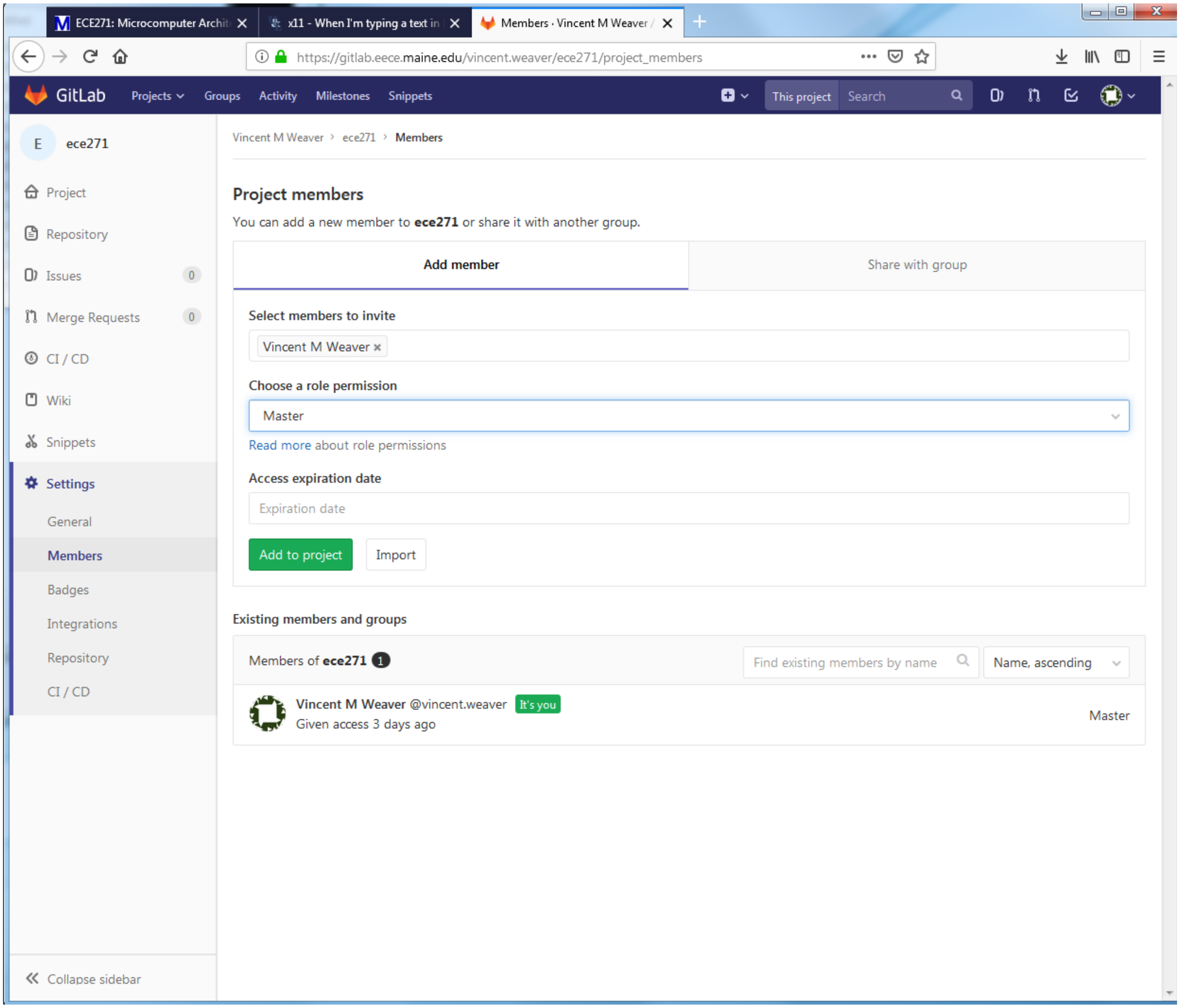
Tags: Reachable

□ Push Default

☑ Prune

□ Prune (All remotes)

Figure 8: Changing the origin.



Project members

You can add a new member to **ece271** or share it with another group.

Select members to invite: Vincent M Weaver ×

Choose a role permission: Master

Read more about role permissions

Access expiration date: Expiration date

Existing members and groups

Members of **ece271** 1

Vincent M Weaver @vincent.weaver It's you
Given access 3 days ago — Master

Figure 9: Adding TA/professor to your repository.

Linux directions

1. Make sure you log into the gitlab server.
2. To generate an RSA key, you can do this at the command prompt.

   ```
   ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
   ```

3. Give the public and private names something descriptive like ecegitlab
4. Give a passkey if you want.
5. These will be in your .ssh directory
6. To make sure these get used when you connect to gitlab, add this to a `.ssh/config` file (replace your.name with your login name)

   ```
   # gitlab.eece.maine.edu account
   Host gitlab.eece.maine.edu
   HostName gitlab.eece.maine.edu
   User your.name
   IdentityFile ~/.ssh/gitlab_rsa
   ```

7. Now log in to gitlab and copy your public key into where it asks
8. Check out the ece271 repository:

   ```
   git clone git@gitlab.eece.maine.edu:vincent.weaver/ece271_2019s.git
   ```

9. Create a New Project on your own gitlab page. For simplicity call it "ece271"
10. Change the destination of the ece271_2019s template you checked out to push to this new "ece271" project. (replace YOUR.NAME with your mainestreet/gitlab account)

    ```
    cd ece271_2019s
    git remote rename origin old-origin
    git remote add origin git@gitlab.eece.maine.edu:YOUR.NAME/ece271.git
    git push -u origin --all
    git push -u origin --tags
    ```

11. Now you can git add, git commit, and git push and it should all work
12. You will have to go to the gitlab interface and add your TA and Prof. Weaver as "master" to this account. See the windows instructions for how to do this.