Introduction to Unmanned Ground Vehicles

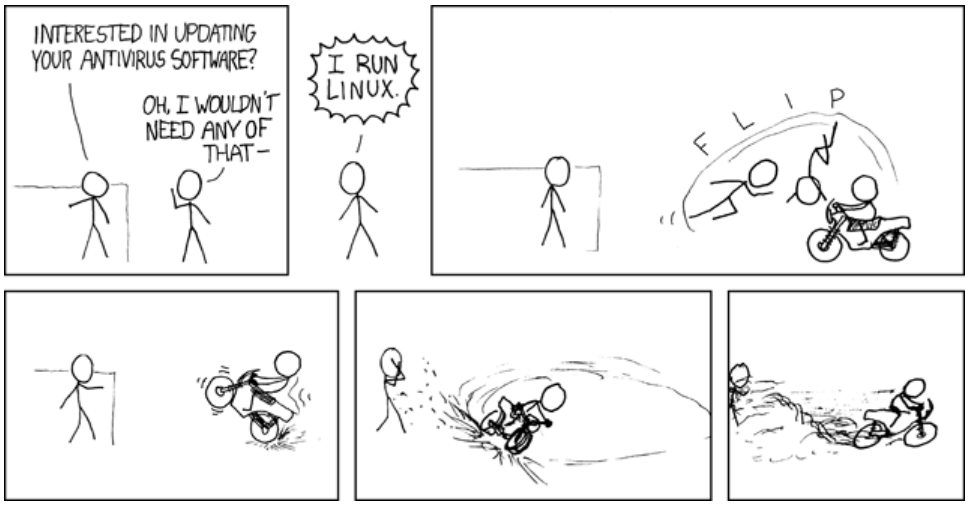
Instructor: Micho Radovnikovich
Email: mtradovn@oakland.edu
ECE 495/595 – Winter 2017

Computer Setup Tutorial

There are two ways to get up and running with Ubuntu and ROS for the course. The first is to do a fresh installation of Ubuntu, either on an empty hard drive, or on a separate partition of a hard drive. The second is to install VMWare Player and use the provided virtual machine (VM) that is already set up with ROS.

A fresh native install will run better, but the virtual machine is definitely easier to get working and space wont need to be created for a separate partition. As long as the host computer is even modestly powerful, and as long as the processor supports hardware virtualization, the VM should run fine for anything encountered in the course.

After completing Steps 1 and 2, you can find instructions for forking and cloning the homework repository in Step 3.



http://xkcd.com/272/

1. Operating System Setup

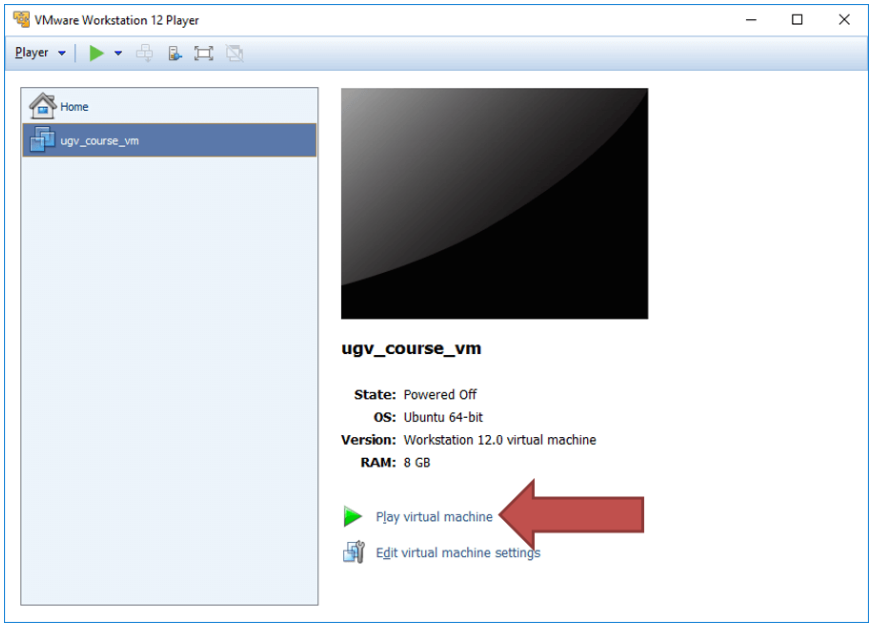
Virtual Machine

1. Download and install VMWare player for Windows (VMware-player-12.1.1-3770994.exe) from here.

2. Copy the provided virtual machine into a folder of your choosing. Typically, VMs are put in a folder called **Virtual Machines** in the **Documents** folder.

3. Open VMWare Player and open the VM from wherever it is on your drive.

4. Start up the virtual machine by clicking **Play virtual machine**.



5. Whenever Ubuntu asks for a password in the VM, the password is **robot**.

Native Install

1. Download Ubuntu version 16.04.1 LTS 64-bit from Ubuntu's website here.
2. Create a bootable USB flash drive to install Ubuntu by following the instructions found here.
3. Restart the computer and boot from the flash drive.
4. Install Ubuntu! This step depends on whether youre installing on a separate partition or on the entire hard drive. If you want to install Ubuntu next to Windows, there are tools out there that help you split your existing Windows installation into a new, second partition. However, be careful when doing this, and back up any important files just in case something happens to your original Windows installation.

2. Software

Whether you're using a native Ubuntu 16.04.1 install or the provided VM, run the following command in a terminal window to download, install and set up the software you will need for the course:

```
bash <(wget -O - http://secs.oakland.edu/~mtradovn/software_setup.bash)
```

Careful! Copying tildes doesn't work... When copying the above command, the tilde in the URL will need to be re-typed!
This will download and run a bash script that will automatically download and install all necessary software, and make the appropriate settings to make things work. Keep in mind that this will take a significant amount of time to finish, and involve downloading a couple gigabytes. Therefore, it is recommended to do this on a good internet connection.

3. Fork and Clone the Course Material Repository

The next steps outline how to fork the course material repository into your own private central repository, and clone it to your local machine. The tutorial document called **Mercurial Concepts** contains more details about the functions of TortoiseHg and Mercurial in general. It is recommended you read that, but for now, you can just follow these steps like a trained monkey

Figure 1 illustrates how the course material repository will be set up for the course. The **ugv_course_central** repository is hosted on the **oaklandrobotics** account on BitBucket.

All students will be given read-only access to this repository so they can pull changes, but not be able to push anything. Each student will then create a fork of this central repository.

The fork will only be accessible by the particular student and the instructor. The student will then clone the fork to their local machine. To receive new homework assignments, the student will pull changes from the CENTRAL repository. To submit homework, the student will push their final revision of the assignment to the fork.

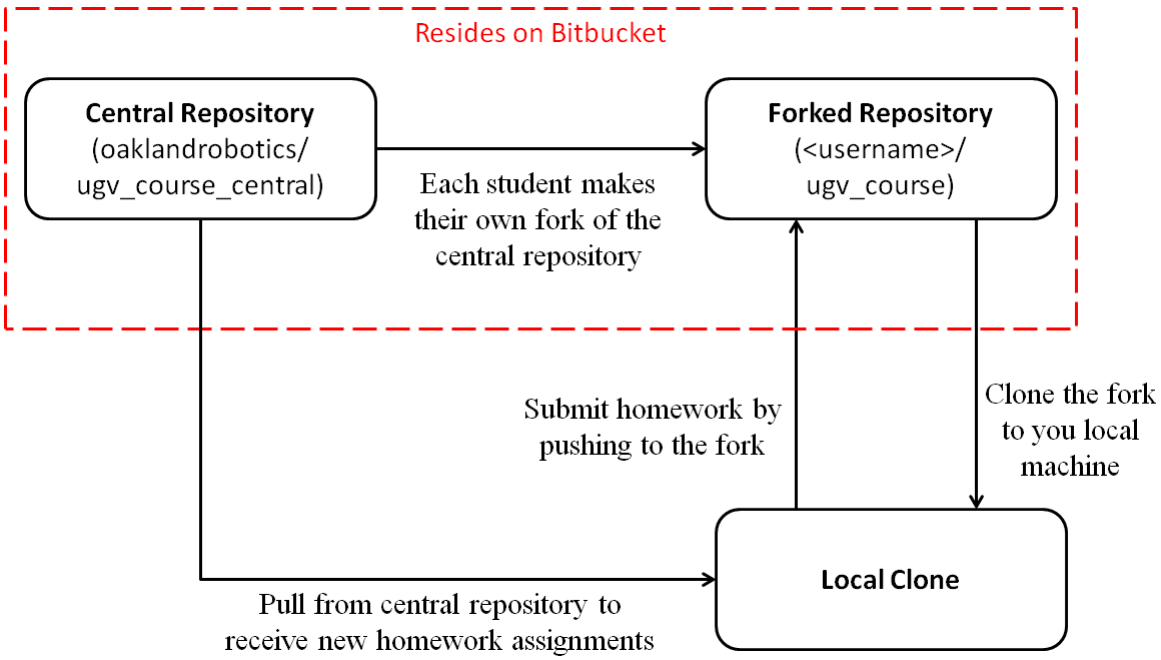
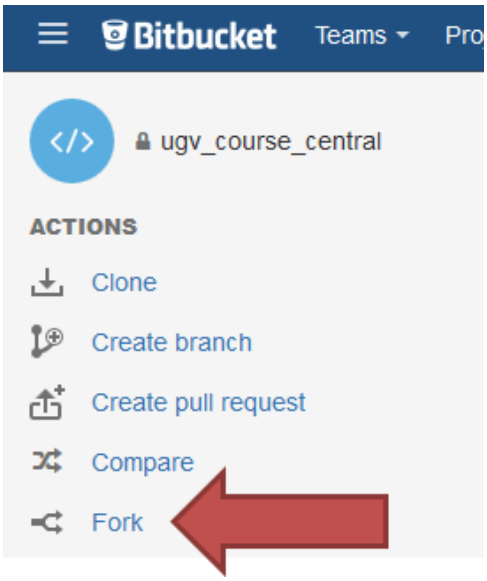
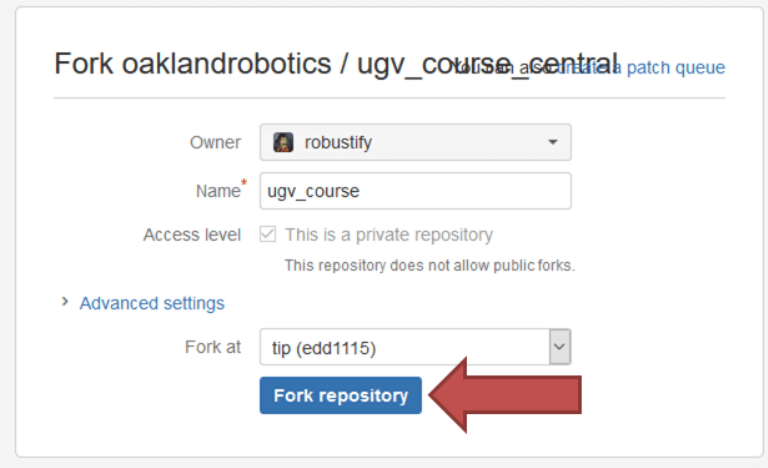


Figure 1:

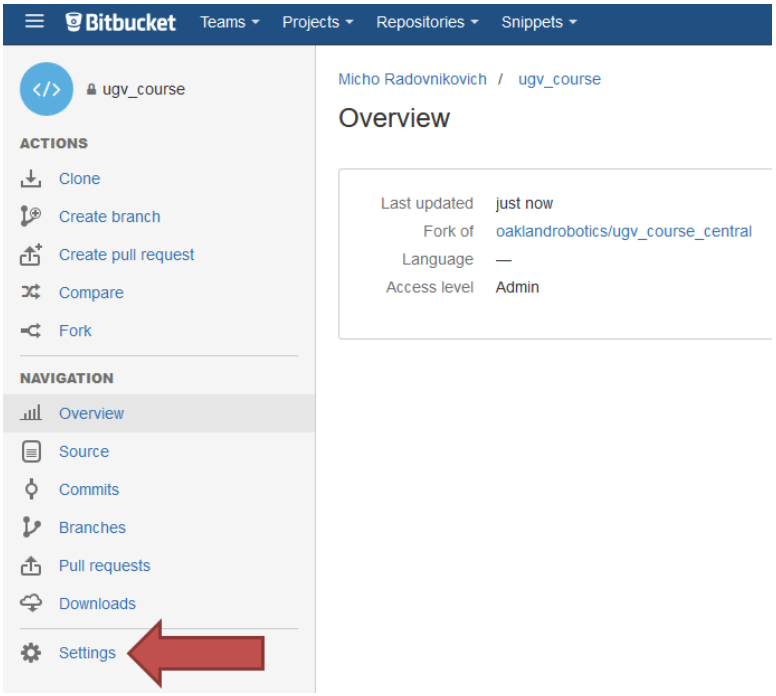
1. Login to BitBucket in a web browser.
2. From the **Dashboard**, navigate to the **oaklandrobotics/ugv_course_central** repository.
3. Click **Fork** on the left under **Actions**:



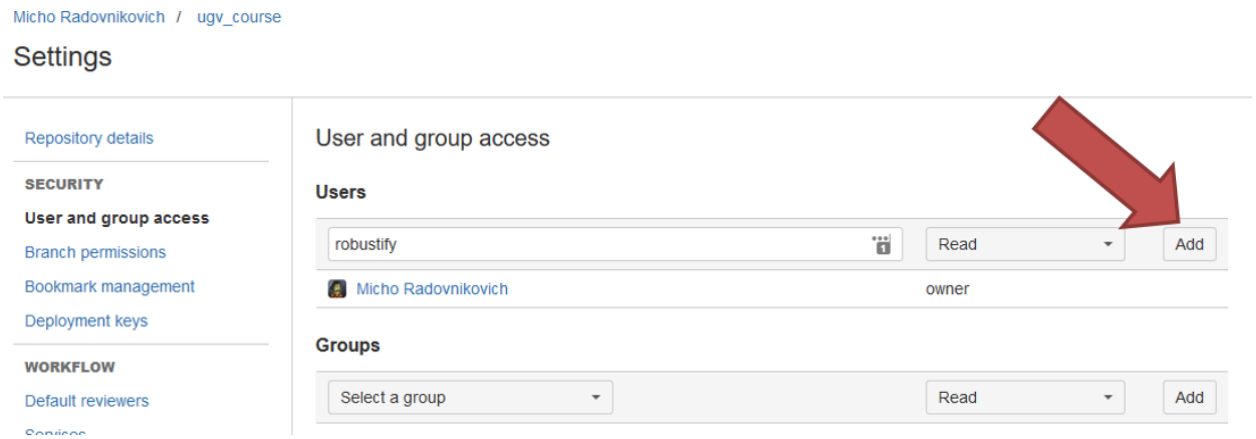
4. In the fork window, make sure the **Owner** is your Bitbucket username, change the **Name** to **ugv_course**, and click **Fork Repository**:



5. After forking the repository, click **Settings** on the bottom of the left side of the page.

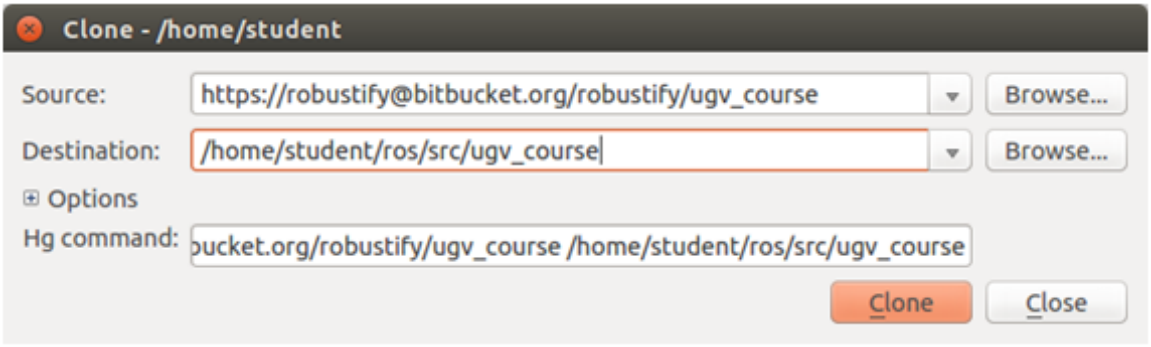


6. Under **User and group access**, put my Bitbucket username **robustify** in the text box under **Users**, set the dropdown box to **Read**, and click **Add**:



7. In Ubuntu, open TortoiseHg using the launcher icon.

8. Click **File → Clone Repository...**



9. In the **Source** box, type the URL to your forked course material repository, which is of the form

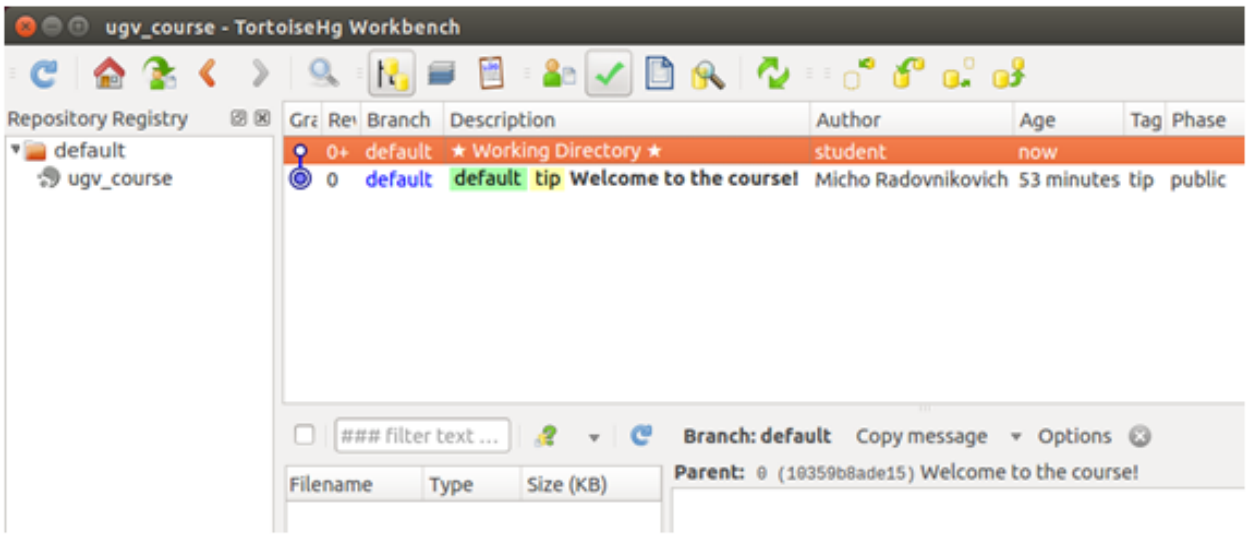
   ```
   https://<bitbucket_username>@bitbucket.org/
                        <bitbucket_username>/ugv_course
   ```

10. In the **Destination** box, type

    ```
    /home/student/ros/src/ugv_course
    ```

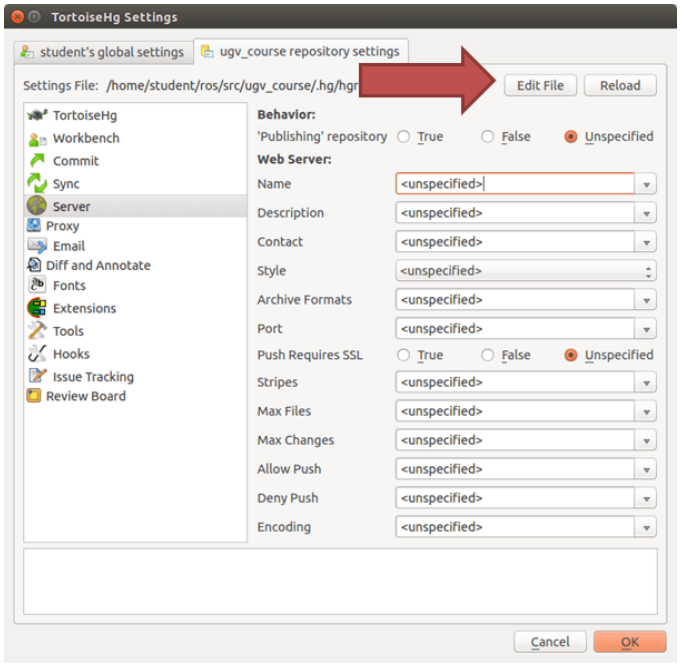
11. Click **Clone**. You will be asked for you Bitbucket account password.

12. After cloning, the main TortoiseHg screen should look like this:



13. By default, TortoiseHg will both pull from and push to the fork. You can configure it to instead pull from the central repository, but push to the fork. To do this, right-click **ugv_course** in the **Repository Registry**, and select **Settings**.

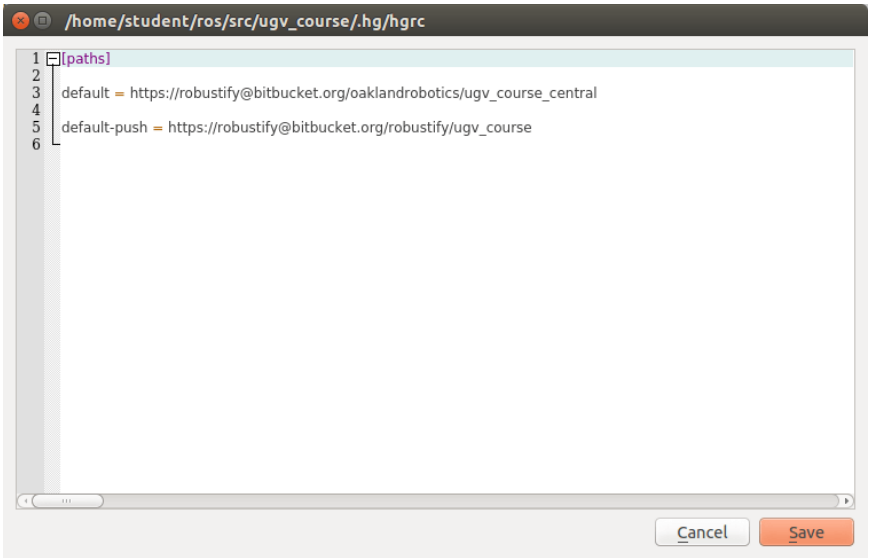
14. In the settings window, click the **Edit File** button:



15. In the settings file, replace what is there currently with the following:

```
[paths]
default = https://<username>@bitbucket.org/oaklandrobotics/ugv_course_central
default-push = https://<username>@bitbucket.org/<username>/ugv_course
```

where <**username**> should be your Bitbucket username.



16. You are now ready to do Homework #1!