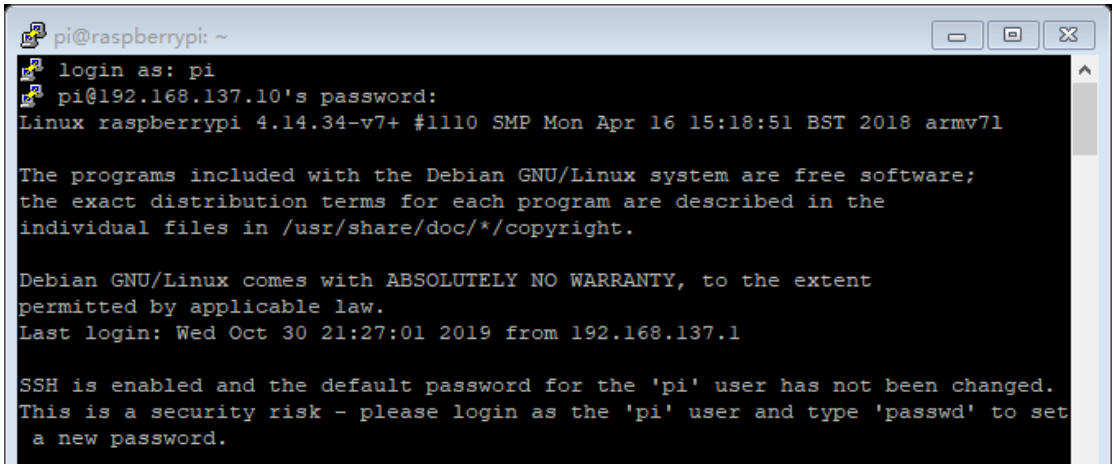
Connecting to GitHub with SSH

Use any tool you like to connect PI with your PC, I use PuTTY as an example:



Login as root:

```
sudo -i
```

First you need to install Git on the raspberry PI

```
sudo apt-get install git
```

Then configure it with:

```
git config --global user.name "Your Name"
git config --global user.email email@example.com
```

If you don't have SSH installed, install it with the following command:

```
sudo apt-get install ssh
```

Start it with the following command:

```
sudo /etc/init.d/ssh start
```

SSH is now starting, but once you restart, you'll need to execute the above command again. You can solve this problem by executing the following command once:

```
sudo update-rc.d ssh defaults
```

I assume that you've already signed up for GitHub. First create the SSH key locally:

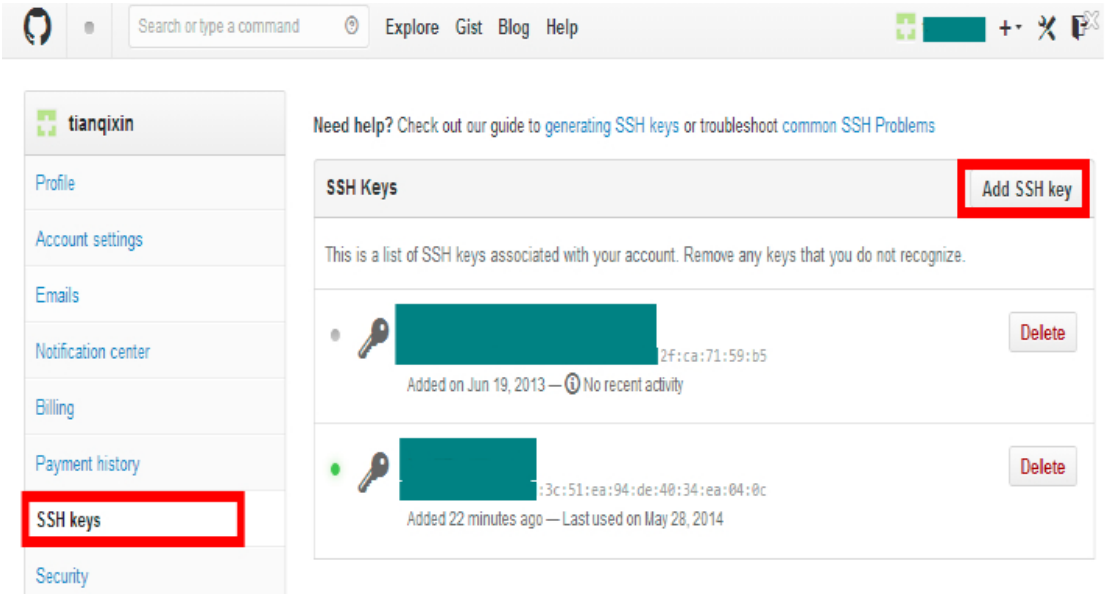
```
ssh-keygen -t rsa -C "your_email@youremail.com"
```

Change your_email@youremail.com to the one you registered on github. And then you're asked to confirm the path and enter your password, and we're going to use the default, just enter all the way. If successful, it will generate a folder **.ssh** under ~/, go to this folder, open **id_rsa.pub**, copy the key inside.

```
root@raspberrypi:~# sudo apt-get install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version (1:2.11.0-3+deb9u2).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@raspberrypi:~# ssh-keygen -t rua -C "d[redacted]1@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:+AmAfdg29aIDzJ2lW2YCNUihLIxWmvtjQAyYZZxjAz8 [redacted]gmail.com
```

```
root@raspberrypi:~# cd /root
root@raspberrypi:~# cd /root/.ssh
root@raspberrypi:~/.ssh# more id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC+S5Bwl7UNxBfIa3BTBStorxVYdMWO9baK196SUKgW
AhQO/TwlMSNtTavhGJ6amnchnjZ3zHzcf+FOUm+UYACV5TezqBx2ungkhVmUvojdLDERiHr7ptZHv8HD
eNzYJTqh6xdY054VAAG8Q5hIslaP/mIlt7RhipplCQtGNuHRyIhfiJJXlyunXm2RyzY39lmJ/Tq916Uy
sjpGaAV3WPP/Ops4XqTOYM5idaR4JYq+lwm5fqoEVS2GM2JxpAdq9gQOHqXWDnXfpjVGqDycgkuQA8V4
jXpeVApfPtpg5hCN/mwnEwSmzUttTStHNrlscgV7XUvNQnktMiREMtNPzlLl dkx2201@gmail.com
root@raspberrypi:~/.ssh# ^C
```

Go back to github, go to Account Settings, select SSH Keys, Add SSH Key and title on the left, and paste the key just generated:



To verify success, type under git bash:

```
ssh -T git@github.com
```

You will see:

Hi username! You've successfully authenticated, but GitHub does not # provide shell access.

Let's now create an empty Git repository. First change the directory to the path where you stored git, for example:

```
cd /home/pi
```

Create a folder for your repository, and initialize and empty the repository. **Note that I am using a "git" user. This user has control over the /home/git directory**:

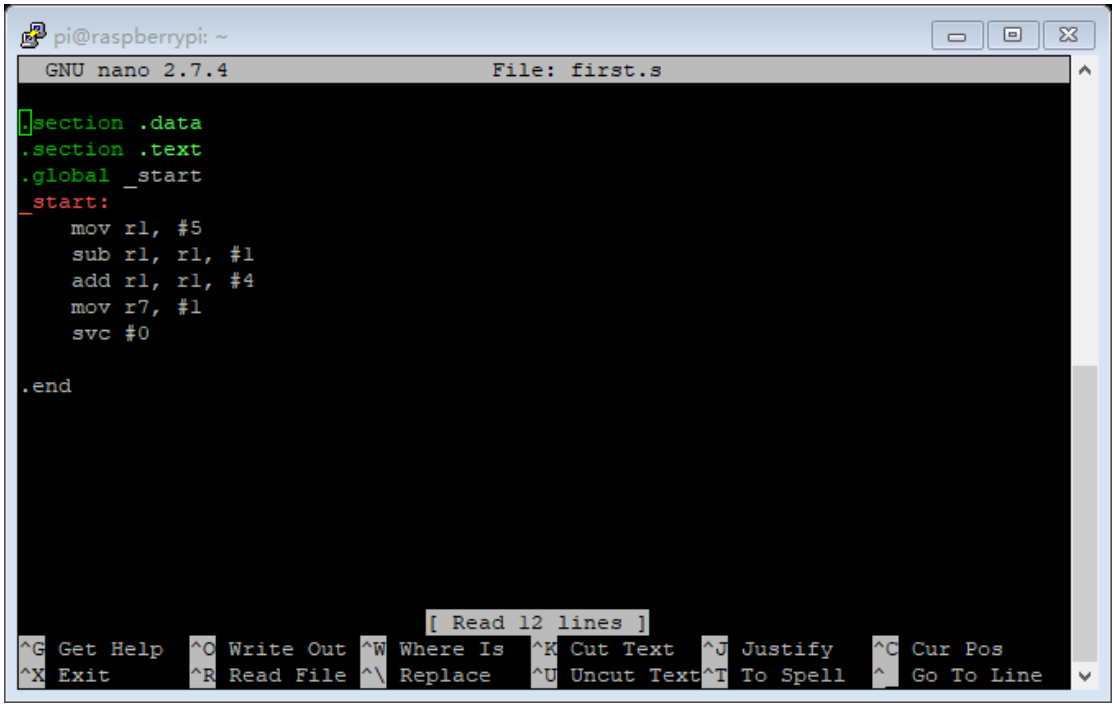
```
mkdir test.git
cd test.git
git --bare init
```

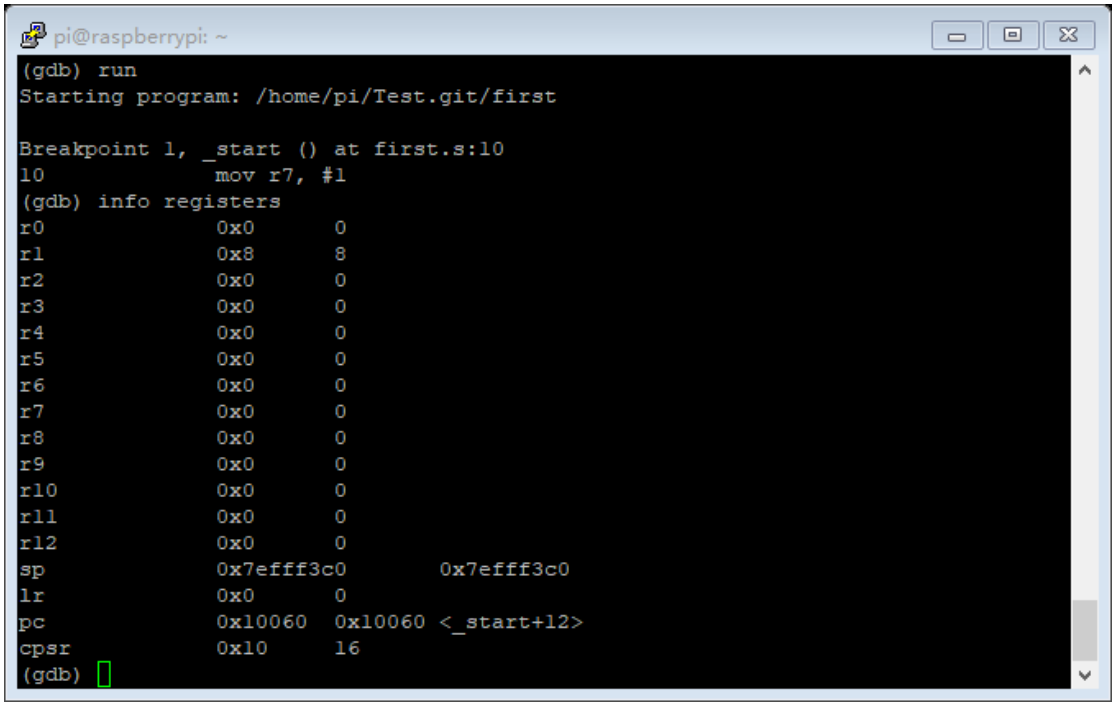
```
root@raspberrypi:~# cd /home/pi
root@raspberrypi:/home/pi# mkdir test.git
root@raspberrypi:/home/pi# cd test.git
root@raspberrypi:/home/pi/test.git# git --bare init
Initialized empty Git repository in /home/pi/test.git/
```

Now you can edit your code. Note that you have to create your code under the repository directory, or git can't manage it. I use your first program in assignment 1 for an example:

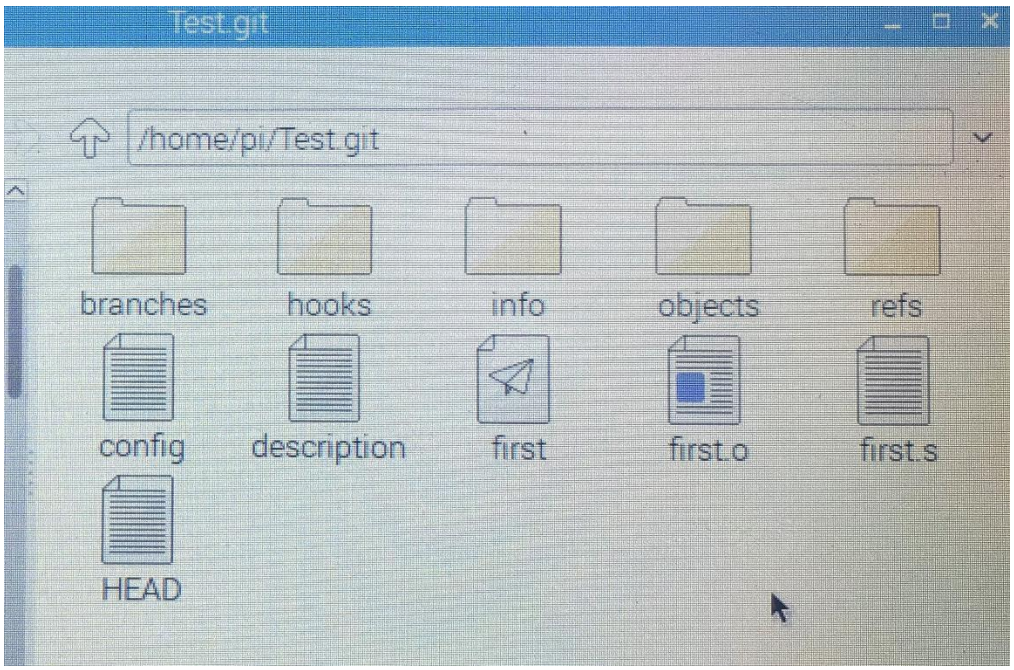
```
root@raspberrypi:/home/pi# nano first.s
root@raspberrypi:/home/pi# cd Test.git
root@raspberrypi:/home/pi/Test.git# nano first.s
root@raspberrypi:/home/pi/Test.git# as -g -o first.o first.s
root@raspberrypi:/home/pi/Test.git# ld -o first first.o
root@raspberrypi:/home/pi/Test.git# gdb first
```

Here are the nano and results:





After above steps, if you check the repository, you'll find three new files were created. In my case, it looks like:



Eventually, we want to push the files to remote repository.

```
$ git remote remove origin
$ git remote add origin git@github.com: <Username>/<Project>.git

//some operations like add and commit//

$ git push -u origin master
```

Or you can clone remote repository:

```
$ git clone git@github.com: <Username>/<Project>.git
$ git add *
$ git status
$ git commit -m "some info"
$ git push -u origin master
```

References:

https://help.github.com/en/github/authenticating-to-github/connecting-to-github-with-ssh
https://help.github.com/en/github/using-git/which-remote-url-should-i-use#cloning-with-ssh