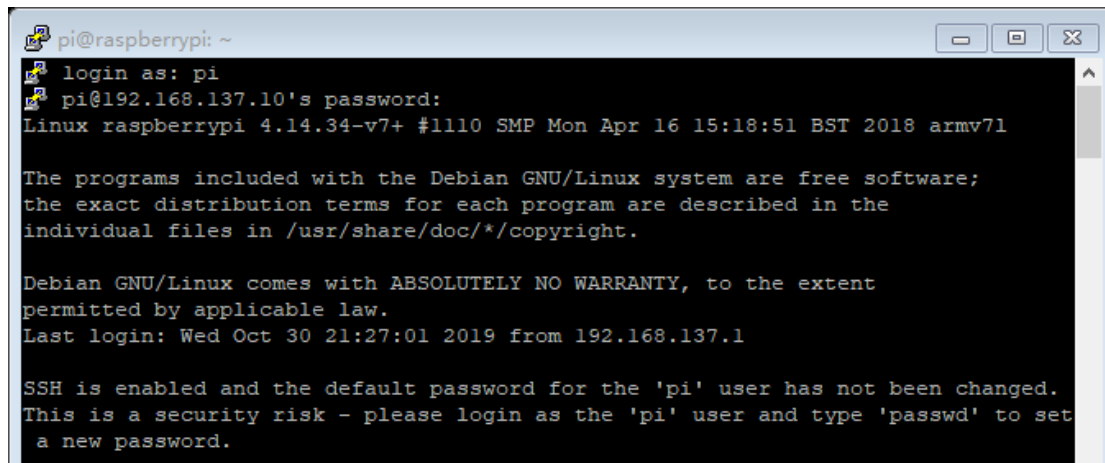


Connecting to GitHub with SSH

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



```
pi@raspberrypi: ~
login as: pi
pi@192.168.137.10's password:
Linux raspberrypi 4.14.34-v7+ #1110 SMP Mon Apr 16 15:18:51 BST 2018 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Oct 30 21:27:01 2019 from 192.168.137.1

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.
```

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 rsa -C "d[REDACTED]l@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:+AmAfdg29aIDzJ2lW2YCNUihLixWmvtjQAYZZxjAz8 [REDACTED]gmail.com

```

```

root@raspberrypi:~# cd /root
root@raspberrypi:~# cd /root/.ssh
root@raspberrypi:~/.ssh# more id_rsa.pub
ssh-rsa AAAAB3NzaClyc2EAAAADAQABAAQCS5Bw17UNxBfIa3BTBStorxVYdMWO9baK196SUKgW
AhQ0/Tw1MSNtTavhGJ6amchnj23zHzcf+FOUm+UYACV5TezqBx2ungkhVmUvojdLDERiHr7ptZHv8HD
eNzYJTqh6xdY054VAAG8Q5hIs1aP/mIlt7Rh1p1CQtGNuHryIhfiJUXlyunXm2RyzY391mJ/Tq916Uy
sjpGa2V3WPP/Ops4XqT0YM5idaR4JYq+lw5fgoEVS2GM2JxpAdq9gQ0HgXWDnXfpjVGqDycgkuQA8V4
jXpeVApfPtpg5hCN/mwnEwSmzUttTStHNrlscgV7XUvNQnktMiREMtNPz1Ll 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:

Need help? Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH Problems](#)

SSH Keys Add SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

- [REDACTED] 2f:ca:71:59:b5
 Added on Jun 19, 2013 — No recent activity Delete
- [REDACTED] :3c:51:ea:94:de:40:34:ea:04:0c
 Added 22 minutes ago — Last used on May 28, 2014 Delete

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:**

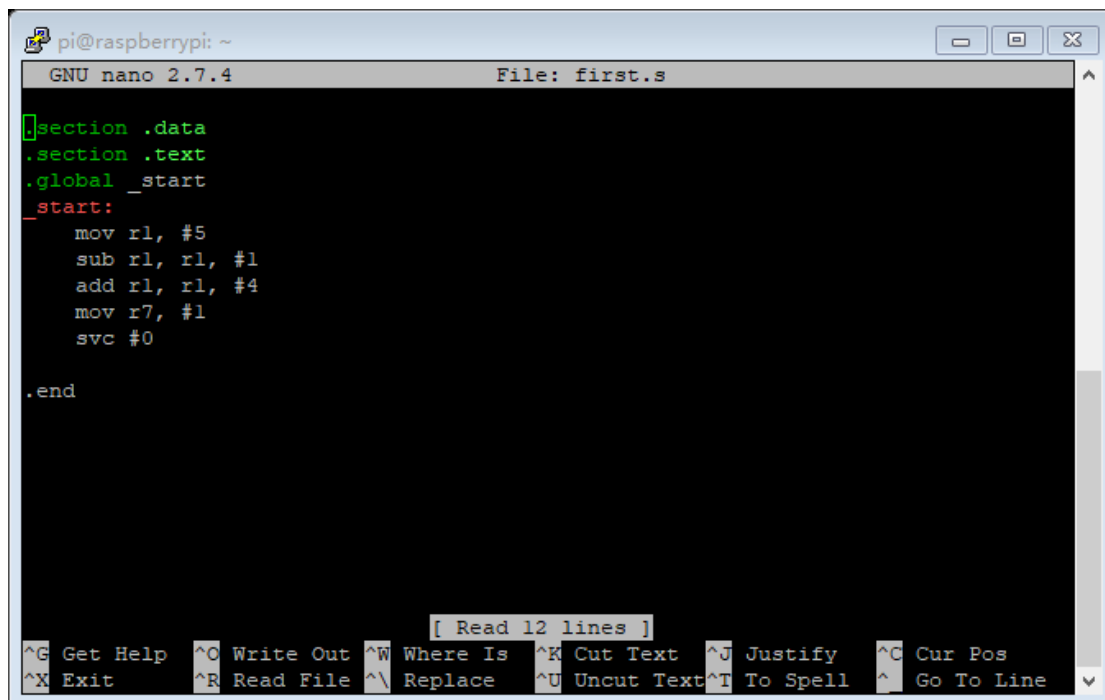
```
mkdir test.git
cd test.git
git --bare init
```

```
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:



```
pi@raspberrypi: ~
GNU nano 2.7.4 File: first.s

[section .data
.section .text
.global _start
_start:
    mov r1, #5
    sub r1, r1, #1
    add r1, r1, #4
    mov r7, #1
    svc #0

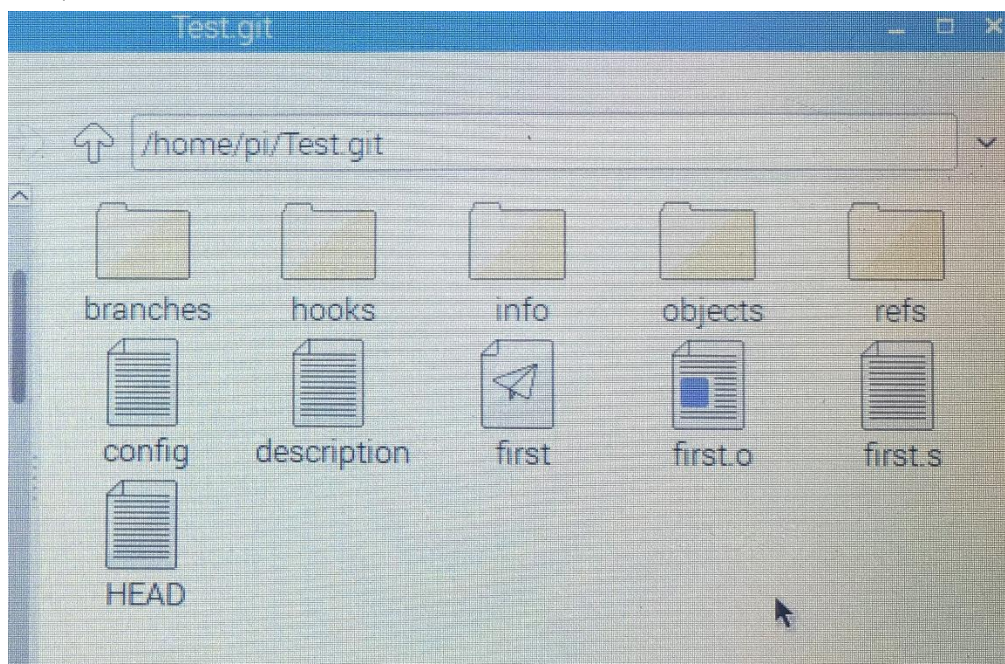
.end

[ Read 12 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line
```

```
pi@raspberrypi: ~
(gdb) run
Starting program: /home/pi/Test.git/first

Breakpoint 1, _start () at first.s:10
10      mov r7, #1
(gdb) info registers
r0          0x0      0
r1          0x8      8
r2          0x0      0
r3          0x0      0
r4          0x0      0
r5          0x0      0
r6          0x0      0
r7          0x0      0
r8          0x0      0
r9          0x0      0
r10         0x0      0
r11         0x0      0
r12         0x0      0
sp          0x7efff3c0 0x7efff3c0
lr          0x0      0
pc          0x10060 0x10060 <_start+12>
cpsr       0x10     16
(gdb) █
```

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>