

# Operating System Design & Implementation

## Lab1: Establish Lab Environment

TA:陳勇旗

### Objective:

In this lab you can learn

- Understand version control system and makefile project
- Understand use QEMU and GDB to debug Linux 0.11 kernel
- Learn how to use the diff to produce the modified kernel patch and commit to GIT system

### 1. Lab 1-1 Linux 0.11 Development Environment Setup

Open VMware and login fedora use below account.

Username: osdi

Password: osdi2014

Note: If you want to do the labs at home you can download the VM image from here

[http://grass8.cs.nctu.edu.tw/OSDI\\_VM.zip](http://grass8.cs.nctu.edu.tw/OSDI_VM.zip)

#### 1.1. Prepare Git environment

#### 1.2. Account registration

Go to: [http://grass8.cs.nctu.edu.tw:8888/users/sign\\_up](http://grass8.cs.nctu.edu.tw:8888/users/sign_up)

Use your **student id** as register username otherwise, you cannot demo in this lab.

## Sign up

9955888	
yungchi	
yungchi@cs.nctu.edu.tw	
*****	
*****	
Sign up	

Have an account? [Sign in](#)

[Forgot your password?](#)

After registration, you will receive a confirmation email in your mail box. If not, please tell TA.

### 1.3. Check out your lab files from GitLab

```
$mkdir osdi  
  
$cd osdi  
  
$git clone http://grass8.cs.nctu.edu.tw:8888/2016/osdi.git
```

Download the Linux root file system from

<http://grass8.cs.nctu.edu.tw/OSDI2015/lab1/osdi.img>

Then you will see the linux0.11 source code ***“linux-0.11”*** and Linux root file system disk ***“osdi.img”*** in your OSDI folder.



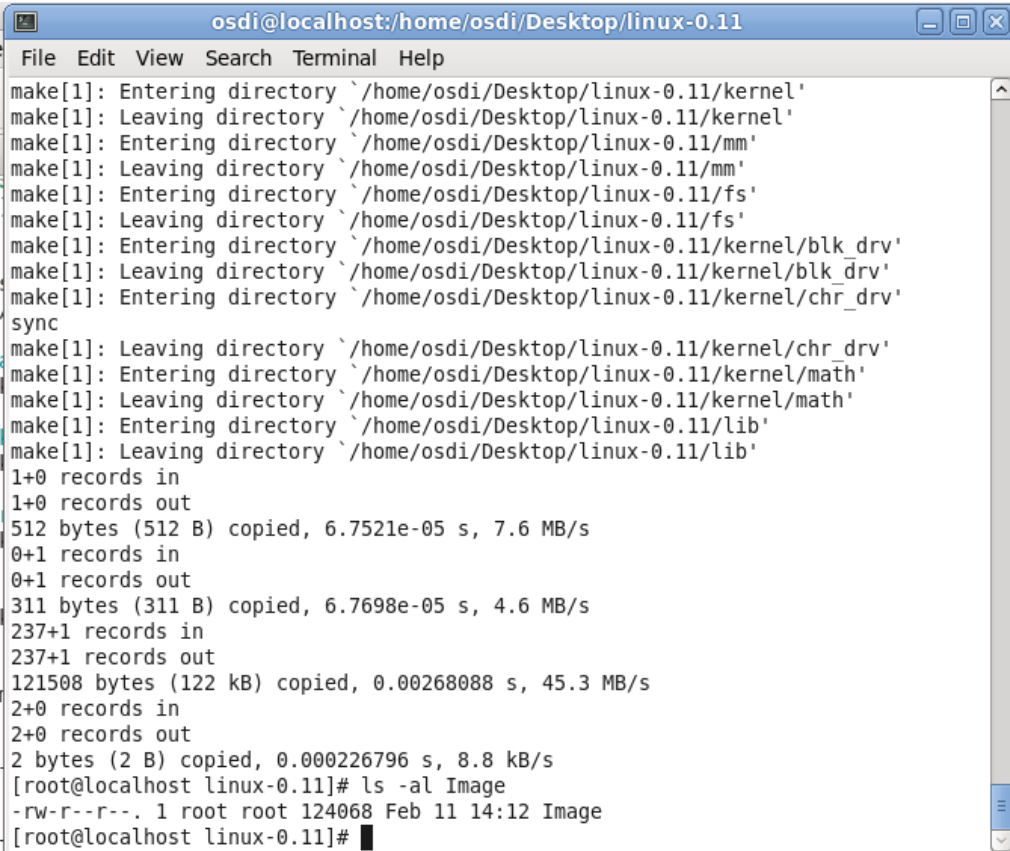
## 1.4. Find the makefile bugs

Our Linux 0.11 makefile have some syntax errors. Please find that and fix them.

## 1.5. Build the Linux 0.11

```
$cd lab1
$cd linux-0.11
$make
```

After make, you will see a bootable file ***“Image”*** in the linux-0.11 folder, it contains system bootloader and linux0.11 kernel.



```
osdi@localhost:/home/osdi/Desktop/linux-0.11
File Edit View Search Terminal Help
make[1]: Entering directory `/home/osdi/Desktop/linux-0.11/kernel'
make[1]: Leaving directory `/home/osdi/Desktop/linux-0.11/kernel'
make[1]: Entering directory `/home/osdi/Desktop/linux-0.11/mm'
make[1]: Leaving directory `/home/osdi/Desktop/linux-0.11/mm'
make[1]: Entering directory `/home/osdi/Desktop/linux-0.11/fs'
make[1]: Leaving directory `/home/osdi/Desktop/linux-0.11/fs'
make[1]: Entering directory `/home/osdi/Desktop/linux-0.11/kernel/blk_drv'
make[1]: Leaving directory `/home/osdi/Desktop/linux-0.11/kernel/blk_drv'
make[1]: Entering directory `/home/osdi/Desktop/linux-0.11/kernel/chr_drv'
sync
make[1]: Leaving directory `/home/osdi/Desktop/linux-0.11/kernel/chr_drv'
make[1]: Entering directory `/home/osdi/Desktop/linux-0.11/kernel/math'
make[1]: Leaving directory `/home/osdi/Desktop/linux-0.11/kernel/math'
make[1]: Entering directory `/home/osdi/Desktop/linux-0.11/lib'
make[1]: Leaving directory `/home/osdi/Desktop/linux-0.11/lib'
1+0 records in
1+0 records out
512 bytes (512 B) copied, 6.7521e-05 s, 7.6 MB/s
0+1 records in
0+1 records out
311 bytes (311 B) copied, 6.7698e-05 s, 4.6 MB/s
237+1 records in
237+1 records out
121508 bytes (122 kB) copied, 0.00268088 s, 45.3 MB/s
2+0 records in
2+0 records out
2 bytes (2 B) copied, 0.000226796 s, 8.8 kB/s
[root@localhost linux-0.11]# ls -al Image
-rw-r--r--. 1 root root 124068 Feb 11 14:12 Image
[root@localhost linux-0.11]#
```

Note: if you have modified any file, please ‘make clean’ before next make.

## 1.6. Compile and install newest QEMU

```
$cd /root
$git clone git://git.qemu-project.org/qemu.git
$cd qemu
$git submodule update --init dtc
$git submodule update --init pixman
$./configure
$make && make install
```

Note: Your QEMU emulator version needs larger than 2.5.x

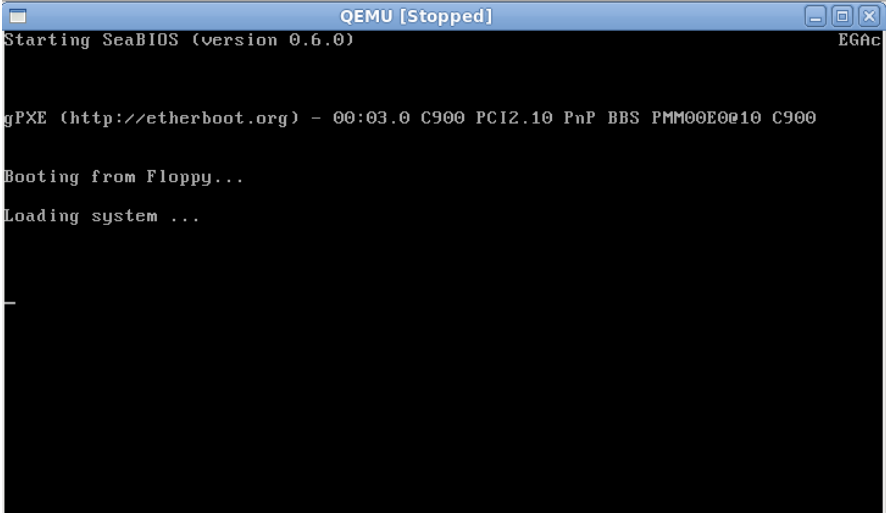
```
[osdi@localhost test]$ qemu-system-i386 -version
QEMU emulator version 2.5.50 Copyright (c) 2003-2008 Fabrice Bellard
[osdi@localhost test]$
```

Reference: <https://en.wikibooks.org/wiki/QEMU>

## 1.7. Run the Linux 0.11

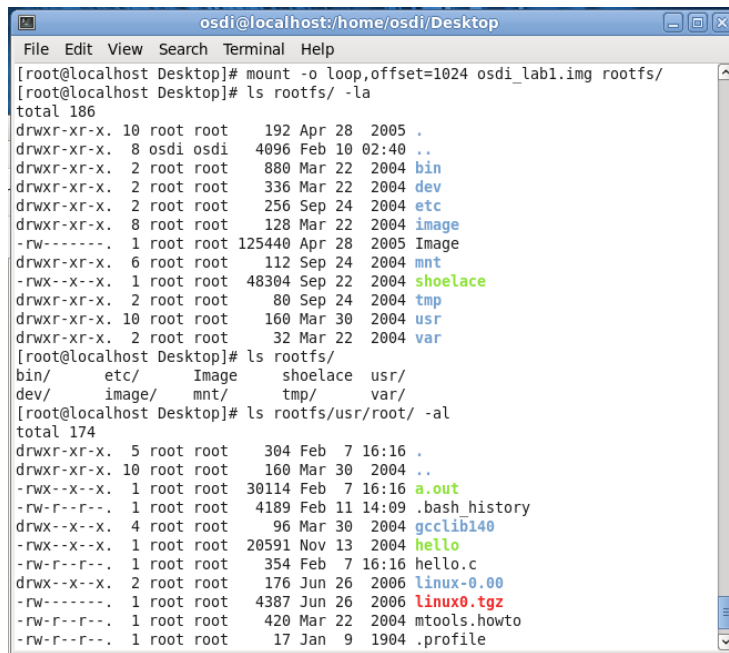
After Linux 0.11 make, the system will produce a bootable floppy disk image called ***“Image”*** in your Linux 0.11 root folder, then you can just use QEMU emulator to load this image and run Linux 0.11.

```
$qemu-system-i386 -m 16M -boot a -fda Image -hda ../osdi.img
```



Note: osdi\_lab1.img is an MINIX root file system it contains some executable files, such as shell system, gcc, etc. You can mount the root file system to modify or add the files.

```
$mkdir rootfs
$mount -o loop,offset=1024 osdi.img rootfs/
```

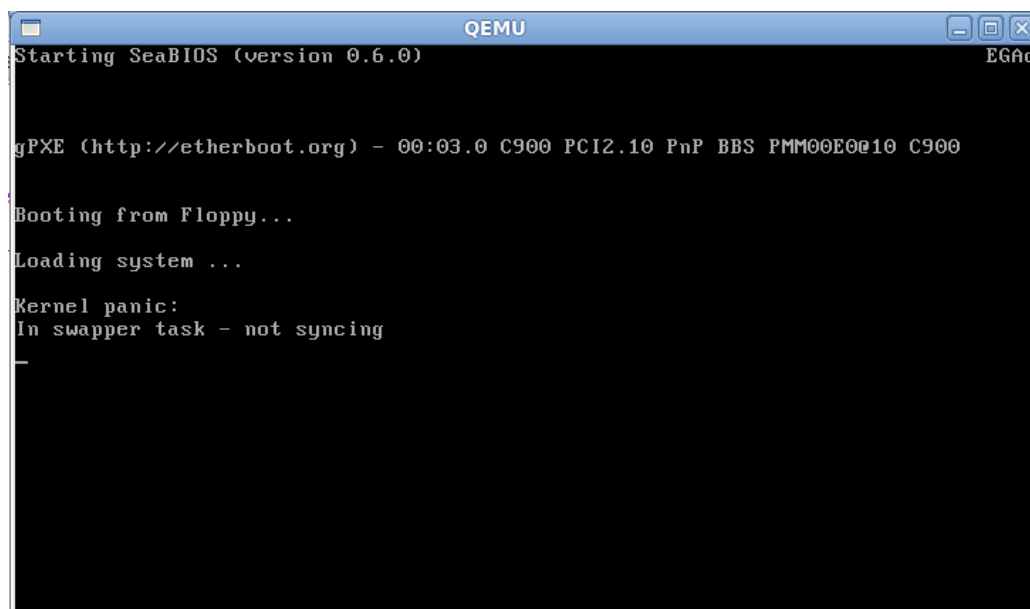


```
osdi@localhost:/home/osdi/Desktop
File Edit View Search Terminal Help
[root@localhost Desktop]# mount -o loop,offset=1024 osdi_lab1.img rootfs/
[root@localhost Desktop]# ls rootfs/ -la
total 186
drwxr-xr-x. 10 root root   192 Apr 28 2005 .
drwxr-xr-x.  8 osdi osdi 4096 Feb 10 02:40 ..
drwxr-xr-x.  2 root root   880 Mar 22 2004 bin
drwxr-xr-x.  2 root root   336 Mar 22 2004 dev
drwxr-xr-x.  2 root root   256 Sep 24 2004 etc
drwxr-xr-x.  8 root root   128 Mar 22 2004 image
-rw-r--r--.  1 root root 125440 Apr 28 2005 Image
drwxr-xr-x.  6 root root   112 Sep 24 2004 mnt
-rwx--x--x.  1 root root 48304 Sep 22 2004 shoelace
drwxr-xr-x.  2 root root    80 Sep 24 2004 tmp
drwxr-xr-x. 10 root root   160 Mar 30 2004 usr
drwxr-xr-x.  2 root root    32 Mar 22 2004 var
[root@localhost Desktop]# ls rootfs/
bin/      etc/      Image    shoelace  usr/
dev/      image/    mnt/     tmp/      var/
[root@localhost Desktop]# ls rootfs/usr/root/ -al
total 174
drwxr-xr-x.  5 root root   304 Feb  7 16:16 .
drwxr-xr-x. 10 root root   160 Mar 30 2004 ..
-rwx--x--x.  1 root root 30114 Feb  7 16:16 a.out
-rw-r--r--.  1 root root 4189 Feb 11 14:09 .bash_history
drwx--x--x.  4 root root    96 Mar 30 2004 gcclib140
-rwx--x--x.  1 root root 20591 Nov 13 2004 hello
-rw-r--r--.  1 root root   354 Feb  7 16:16 hello.c
drwx--x--x.  2 root root   176 Jun 26 2006 linux-0.00
-rw-r--r--.  1 root root 4387 Jun 26 2006 linux0.tgz
-rw-r--r--.  1 root root   420 Mar 22 2004 mtools.howto
-rw-r--r--.  1 root root    17 Jan  9 1904 .profile
```

## 2. Lab 1-2 Debug kernel

### 2.1. Find the kernel bugs

In this lab, you need to use the GDB to find out the bugs and fix them.



```
QEMU
Starting SeaBIOS (version 0.6.0)
EGAC

gPXE (http://etherboot.org) - 00:03.0 C900 PCI2.10 PnP BBS PMM00E0010 C900

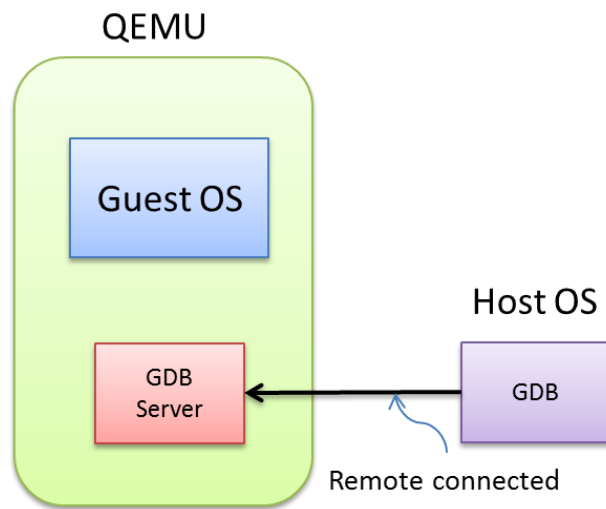
Booting from Floppy...
Loading system ...
Kernel panic:
In swapper task - not syncing
-
```

Hint: Use backtrace command to find the kernel bugs, there are 2 bugs in the lab1 Linux kernel.

### 2.2. Debug the Linux 0.11 on QEMU

In QEMU environment, you can debug the OS kernel via gdbserver and gdb. Use this

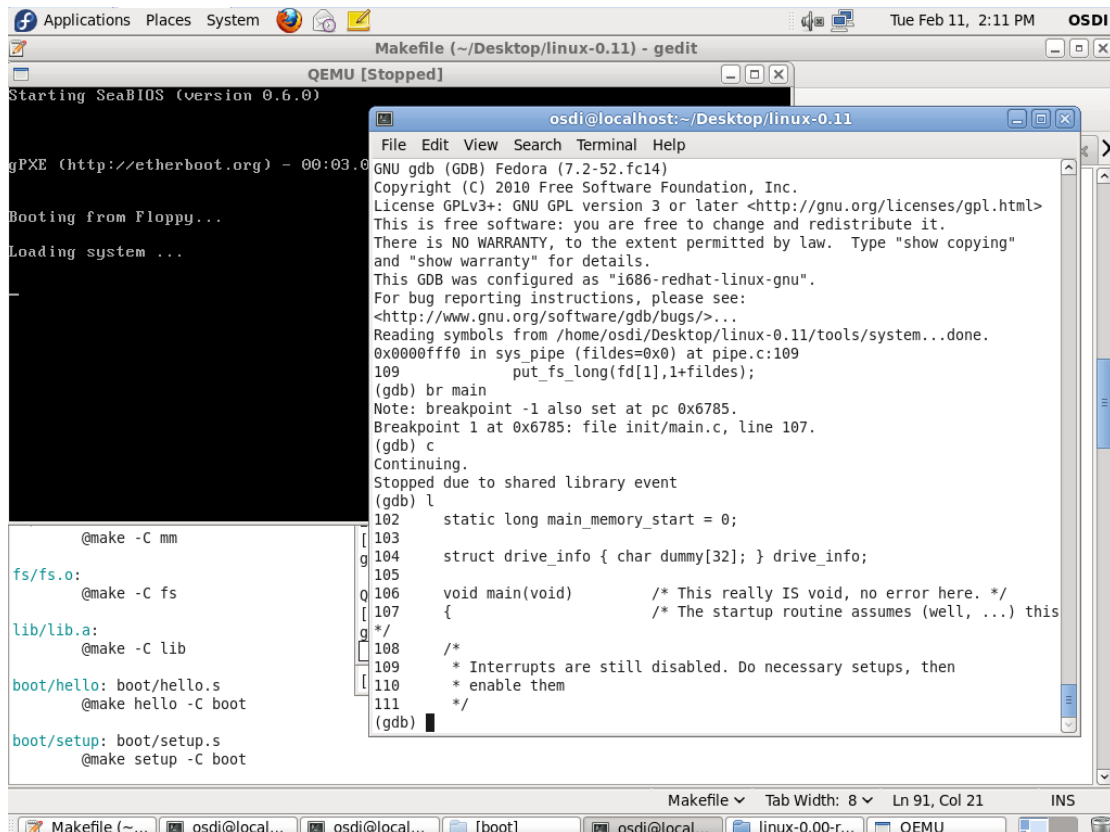
to find the kernel and open gdb with Linux 0.11 kernel symbol file.



```
$qemu-system-i386 -m 16M -boot a -fda Image -hda ../osdi.img -s -S -serial stdio  
  
Open another console  
  
$cd linux-0.11  
  
$gdb tools/system  
  
(gdb) target remote localhost:1234
```

Notes:

- You can add some breakpoint at OS main function or interrupt service routine than step by step see how the code execute.
- Sometimes (e.g. system call, context switch, ...) you may need to debug in assembly, you can see below references that could help debug easier.  
[http://www.csee.umbc.edu/~cpatel2/links/310/nasm/gdb\\_help.shtml](http://www.csee.umbc.edu/~cpatel2/links/310/nasm/gdb_help.shtml)  
<http://mohit.io/blog/assembly-and-the-art-of-debugging/>  
<https://sourceware.org/gdb/onlinedocs/gdb/Machine-Code.html>
- The kernel/panic.c is correct, you need not change it.



## ● Useful GDB commands

‘b’- set breakpoint

‘c’- continue program

‘list’- list code

‘backtrace’- show call stack

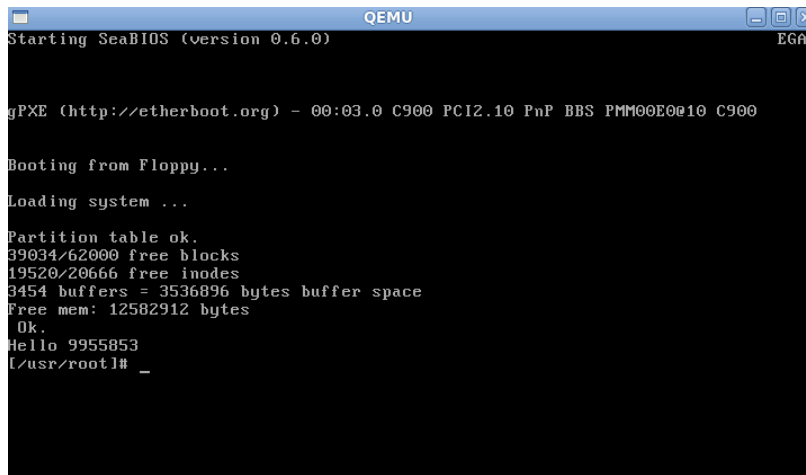
‘info r’-show current registers value

Ctrl+c – stop program

GDB command reference: <http://www.cmlab.csie.ntu.edu.tw/~daniel/linux/gdb.html>

## 2.3. Print your student id

Modify the Linux 0.11 source code and print your student id before shell startup.



```
QEMU
Starting SeaBIOS (version 0.6.0)

gPXE (http://etherboot.org) - 00:03.0 C900 PCI2.10 PnP BBS PMM00E0010 C900

Booting from Floppy...
Loading system ...
Partition table ok.
39034/62000 free blocks
19520/20666 free inodes
3454 buffers = 3536896 bytes buffer space
Free mem: 12582912 bytes
Ok.
Hello 9955853
[usr/root]# _
```

### 3. Lab 1-3 Submit your lab1 for DEMO

#### 3.1. Create the kernel path file

Create your kernel path file as name “{student id}.patch”, it needs to include your Linux 0.11 all modifications except the “osdi.img” file.

You can just use the diff command to produce the patch file.

**Note:** Before diff, please *“make clean”* your project first.

Reference: <http://blog.longwin.com.tw/2013/08/linux-diff-patch-learn-note-2013/>

<http://git-scm.com/docs/diff-generate-patch>

#### 3.2. Commit your source code and image

Create your own repository on Gitlab website.

Step1: Login into <http://grass8.cs.nctu.edu.tw:8888> and new project.




You don't have access to any projects right now.  
You can create up to **10 projects**.

**New project »**

Step2: Use your student ID as the project name and select visibility level as private.



 New Project Search

Project name

9955853

Namespace

Yung Chi Chen

[Customize repository name?](#)

[Import existing repository?](#)

Description (optional)

OSDI Lab files

Visibility Level (?)

☒ Private

Project access must be granted explicitly for each user.

☐ Internal

The project can be cloned by any logged in user.

☐ Public

The project can be cloned without any authentication.

Create project

Need

Step3: Add remote repository to your local machine.

```
$cd osdi

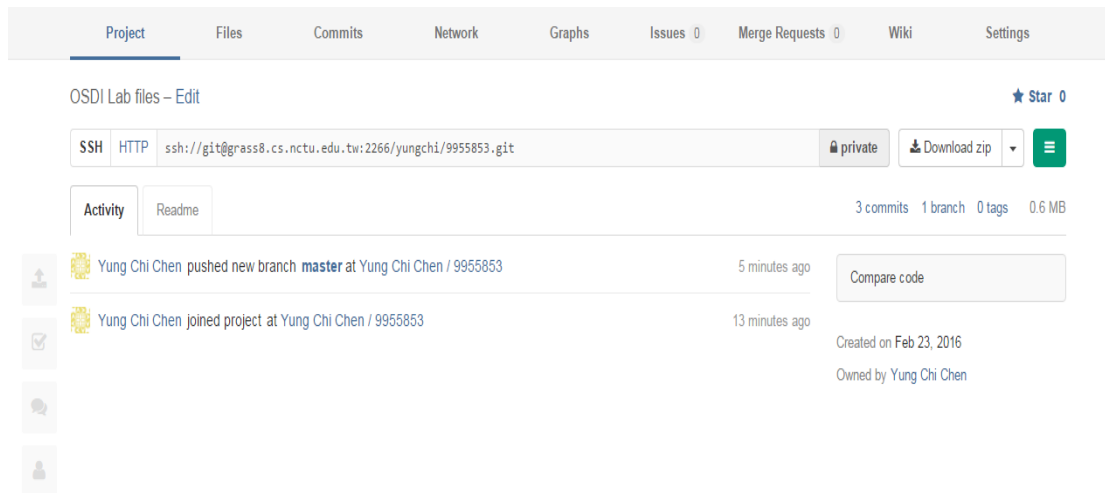
$ git config --global user.name "{Your username}"

$git config --global user.email "{Your email}"

$git remote add osdi http://grass8.cs.nctu.edu.tw/{Your username}/{Project
name}.git

$git push osdi master
```

You can see below project status on gitlab web site when complete step3.



Use “git remote -v” command will see two remote repositories setting.

```
[root@localhost osdi]# git remote -v
origin http://grass8.cs.nctu.edu.tw:8888/2016/osdi.git (fetch)
origin http://grass8.cs.nctu.edu.tw:8888/2016/osdi.git (push)
osdi http://grass8.cs.nctu.edu.tw:8888/yungchi/9955853.git (fetch)
osdi http://grass8.cs.nctu.edu.tw:8888/yungchi/9955853.git (push)
[root@localhost osdi]#
```

Step4: Commit and push the patch file to your remote repository

```
$git add {student id}.patch

$git commit -m "Lab1 demo"

$git push osdi master
```

Note: GIT documentation and command reference: <https://git-scm.com/doc>

Git-基礎-與遠端協同工作: <https://git-scm.com/book/zh-tw/v1/Git->

[https://git-scm.com/book/zh-tw/v1/Git-](https://git-scm.com/book/zh-tw/v1/Git-%E5%9F%BA%E7%A4%8E-%E8%88%87%E9%81%A0%E7%AB%AF%E5%8D%94%E5%90%8C%E5%B7%A5%E4%BD%9C)

[https://git-scm.com/book/zh-tw/v1/Git-](https://git-scm.com/book/zh-tw/v1/Git-%E5%9F%BA%E7%A4%8E-%E8%88%87%E9%81%A0%E7%AB%AF%E5%8D%94%E5%90%8C%E5%B7%A5%E4%BD%9C)

[https://git-scm.com/book/zh-tw/v1/Git-](https://git-scm.com/book/zh-tw/v1/Git-%E5%9F%BA%E7%A4%8E-%E8%88%87%E9%81%A0%E7%AB%AF%E5%8D%94%E5%90%8C%E5%B7%A5%E4%BD%9C)

#### **4. Questions**

Q1: How can we setting QEMU to emulate an Intel Core i7 CPU?

Q2: Can you roughly describe the Linux 0.11 “OS kernel” startup code flow?

Q3: What different between git pull, fetch and clone?