Lab 1. Cross Development Environment

1. Purpose

- Set up Ubuntu environment on PC and Debian on Beaglebone

- Construct a cross-development system on PC, cross complete some programs and run it.

- Construct a module program development environment on a PC, cross compile embedded module program and run it.

- Set up NFS between PC and Beaglebone

2. Experiment sequence and Experimental results

Step 1. Test operation of development PC with Ubuntu

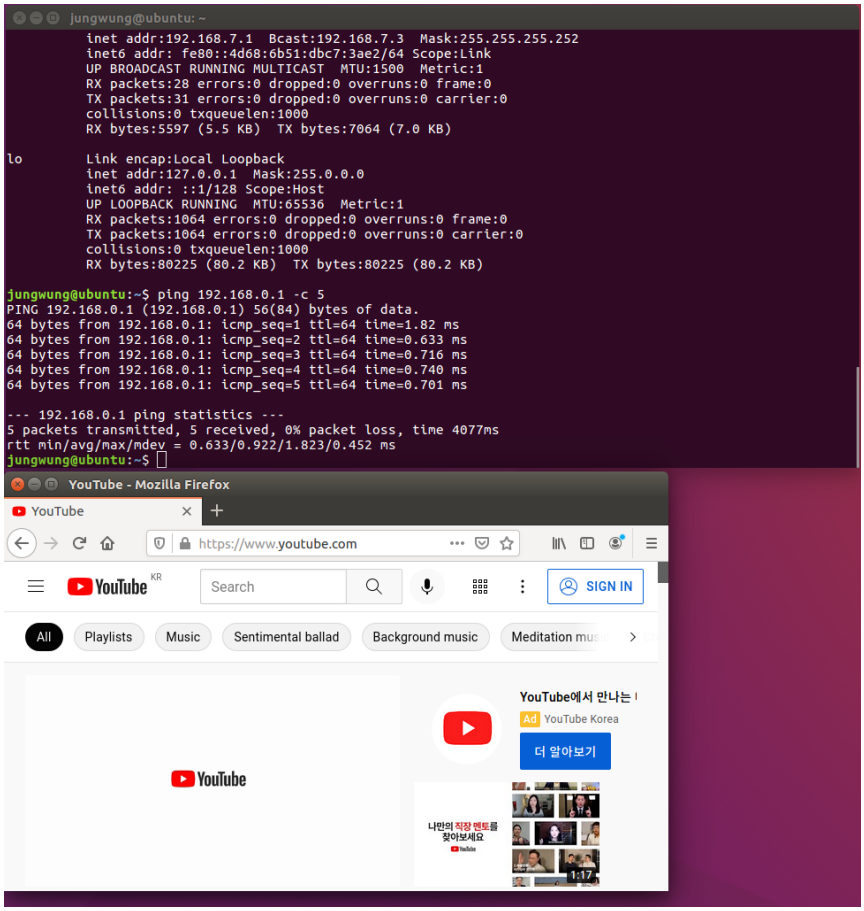
Check the kernel version of Ubuntu 16.04 and network connection.

```
jungwung@ubuntu:~$ uname -r
4.15.0-112-generic
```

```
jungwung@ubuntu:~$ ifconfig
ens33     Link encap:Ethernet  HWaddr 00:0c:29:f8:3d:a1
          inet addr:192.168.0.13  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::fed7:6d53:8861:91df/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:269 errors:0 dropped:0 overruns:0 frame:0
          TX packets:216 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:47860 (47.8 KB)  TX bytes:21534 (21.5 KB)

enxf045da813853 Link encap:Ethernet  HWaddr f0:45:da:81:38:53
          inet6 addr: fe80::4d68:6b51:dbc7:3ae2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:40 errors:0 dropped:0 overruns:0 frame:0
          TX packets:48 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7889 (7.8 KB)  TX bytes:12685 (12.6 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:320 errors:0 dropped:0 overruns:0 frame:0
          TX packets:320 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:25228 (25.2 KB)  TX bytes:25228 (25.2 KB)
```

The kernel version of Ubuntu 16.04 is 4.15.0-112-generic.

IP address is 192.168.0.13.



Ping 192.168.0.1 and firefox web browser also worked well.

Step 2. Install Beaglebone Debian on microSD

Download proven Debian OS image.

```
yeolia@ubuntu:~/Downloads$ ls -la ~/Downloads/
total 6533432
drwxr-xr-x  2 yeolia yeolia       4096 Sep 16 21:45 .
drwxr-xr-x 19 yeolia yeolia       4096 Sep 23 18:54 ..
-rw-rw-r--  1 yeolia yeolia 3565158400 Sep 15 16:46 bone-debian-7.9-lxde-4gb-armhf-2015-11-12-4gb.img
-rw-rw-r--  1 yeolia yeolia  626164520 Sep 15 16:46 bone-debian-7.9-lxde-4gb-armhf-2015-11-12-4gb.img.xz
-rw-rw-r--  1 yeolia yeolia   75903744 Sep 16 15:43 code_1.60.1-1631294805_amd64.deb
-rw-rw-r--  1 yeolia yeolia     768211 Sep 15 16:42 DesignLab_RoboCam_Lab1_note(1).pdf
-rw-rw-r--  1 yeolia yeolia     768211 Sep 14 23:42 DesignLab_RoboCam_Lab1_note.pdf
-rw-rw-r--  1 yeolia yeolia     103481 Sep 15 16:42 EE405A_DesignLab_CourseDescription.pdf
-rw-rw-r--  1 yeolia yeolia      12217 Sep 15 16:42 experimental_materials.zip
-rw-rw-r--  1 yeolia yeolia     626290 Sep 15 16:42 How to install GCC at windows.pdf
-rw-rw-r--  1 yeolia yeolia 2414960640 Sep 15 16:49 Kernel3.8.13-bone79_linux.gz
-rw-rw-r--  1 yeolia yeolia    1167725 Sep 15 16:42 Lab1 supplement1.pdf
-rw-rw-r--  1 yeolia yeolia      80135 Sep 15 16:42 Lab1 Typos.pdf
-rw-rw-r--  1 yeolia yeolia    4399540 Sep 15 16:42 lecture1.pptx
-rw-rw-r--  1 yeolia yeolia         74 Sep 16 21:45 .~lock.lecture1.pptx#
-rw-rw-r--  1 yeolia yeolia      67622 Sep 15 16:42 TA_list.pdf
yeolia@ubuntu:~/Downloads$ sha256sum bone-debian-7.9-lxde-4gb-armhf-2015-11-12-4gb.img.xz
f6e67ba01ff69d20f2c655f5e429c3e6c2398123bcd3d8d548460c597275d277  bone-debian-7.9-lxde-4gb-armhf-2015-11-12-4gb.img.xz
yeolia@ubuntu:~/Downloads$
```

Unpack the file and Check size using the commands.

$ cd ~/Downloads

$ unxz -k bone-debian-7.9-lxde-4gb-armhf-2015-11-12-4gb.img.xz

$ ls -la

size of files, sha256 match with the provided Lab1 supplement pdf data.

Write Debian image to uSD.

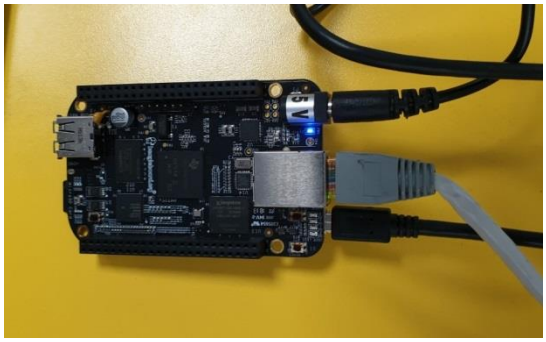
```
yeolia@ubuntu:~$ df
Filesystem     1K-blocks     Used Available Use% Mounted on
udev             1977436        0   1977436   0% /dev
tmpfs             401592    11380    390212   3% /run
/dev/sda1       19525500 11430136   7080480  62% /
tmpfs            2007952    57400   1950552   3% /dev/shm
tmpfs               5120        4      5116   1% /run/lock
tmpfs            2007952        0   2007952   0% /sys/fs/cgroup
tmpfs             401592      128    401464   1% /run/user/1000
/dev/sdb2        3263536  1930228   1147812  63% /media/yeolia/rootfs
/dev/sdb1          98094    30360     67734  31% /media/yeolia/BEAGLEBONE
```

The device name of microSD is /dev/sdb, which contains two partitions.

```
yeolia@ubuntu:~$ umount /dev/sdb1
yeolia@ubuntu:~$ umount /dev/sdb2
yeolia@ubuntu:~$ sudo fdisk /dev/sdb

Welcome to fdisk (util-linux 2.27.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): d
Partition number (1,2, default 2): 1

Partition 1 has been deleted.

Command (m for help): d
Selected partition 2
Partition 2 has been deleted.

Command (m for help): d
No partition is defined yet!
Could not delete partition 1

Command (m for help): w

The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

yeolia@ubuntu:~$ sudo mkfs.ext3 /dev/sdb
mke2fs 1.42.13 (17-May-2015)
Found a dos partition table in /dev/sdb
Proceed anyway? (y,n) y
Creating filesystem with 957440 4k blocks and 239520 inodes
Filesystem UUID: cf99c69d-fb1c-4a48-81e2-490277b4eea1
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

yeolia@ubuntu:~$ sudo dd if=./Downloads/bone-debian-7.9-lxde-4gb-armhf-2015-11-12-4gb.img of=/dev/sdb status=progress
3564585472 bytes (3.6 GB, 3.3 GiB) copied, 3550 s, 1.0 MB/s
6963200+0 records in
6963200+0 records out
3565158400 bytes (3.6 GB, 3.3 GiB) copied, 3555.47 s, 1.0 MB/s
```

cleaned original usb data and made a new partition using above commands.

Connect Beaglebone on PC.



Beaglebone is connected with 5V adapter, USB connection between computer, and LAN connection with local router.

Prepare PC for Bone console.

```
jungwung@ubuntu:~$ sudo apt-get install minicom
[sudo] password for jungwung:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  lrzsz
The following NEW packages will be installed:
  lrzsz minicom
0 upgraded, 2 newly installed, 0 to remove and 186 not upgraded.
Need to get 302 kB of archives.
After this operation, 1,428 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 lrzsz amd64 0.12.21-10~build0.16.04.1 [70.5 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 minicom amd64 2.7-1+deb8u1build0.16.04.1 [232 kB]
Fetched 302 kB in 2s (150 kB/s)
Selecting previously unselected package lrzsz.
(Reading database ... 177262 files and directories currently installed.)
Preparing to unpack .../lrzsz_0.12.21-10~build0.16.04.1_amd64.deb ...
Unpacking lrzsz (0.12.21-10~build0.16.04.1) ...
Selecting previously unselected package minicom.
Preparing to unpack .../minicom_2.7-1+deb8u1build0.16.04.1_amd64.deb ...
Unpacking minicom (2.7-1+deb8u1build0.16.04.1) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up lrzsz (0.12.21-10~build0.16.04.1) ...
Setting up minicom (2.7-1+deb8u1build0.16.04.1) ...
```

Type the command "$ sudo minicom -s –w".

Select "Serial port setup" and change Serial Device to "/dev/ttyACM0".

Select "Save setup as dfl" and "Configuration saved".

Start Beaglebone and login as user "debian" with password "temppwd".

```
beaglebone login: debian
Password:
Last login: Thu Nov 12 19:10:10 UTC 2015 on ttyGS0
Linux beaglebone 3.8.13-bone79 #1 SMP Tue Oct 13 20:44:55 UTC 2015 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.
```

Check Beaglebone Debian kernel version.

```
debian@beaglebone:~$ uname -r
3.8.13-bone79
```

Add superuser, become superuser, and see the directory "/root" using the commands.

```
debian@beaglebone:~$ sudo passwd root
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
debian@beaglebone:~$ su
Password:
root@beaglebone:/home/debian# ls -la /root
total 32
drwx------  5 root root 4096 Nov 12  2015 .
drwxr-xr-x 22 root root 4096 Nov 12  2015 ..
-rw-------  1 root root  252 Nov 12  2015 .bash_history
-rw-r--r--  1 root root  570 Jan 31  2010 .bashrc
drwxr-xr-x  8 root root 4096 Nov 12  2015 .c9
drwxr-xr-x  3 root root 4096 Nov 12  2015 .cache
drwxr-xr-x  3 root root 4096 Nov 12  2015 .node-gyp
-rw-r--r--  1 root root  140 Nov 19  2007 .profile
```

Add user, check "/home" directory, and exit.

```
root@beaglebone:/home/debian# adduser jungwungpark
Adding user `jungwungpark' ...
Adding new group `jungwungpark' (1002) ...
Adding new user `jungwungpark' (1002) with group `jungwungpark' ...
Creating home directory `/home/jungwungpark' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for jungwungpark
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:

Is the information correct? [Y/n] Adding new user `jungwungpark' to extra groups ...
Adding user `jungwungpark' to group `dialout' ...
Adding user `jungwungpark' to group `i2c' ...
Adding user `jungwungpark' to group `spi' ...
Adding user `jungwungpark' to group `cdrom' ...
Adding user `jungwungpark' to group `floppy' ...
Adding user `jungwungpark' to group `audio' ...
Adding user `jungwungpark' to group `video' ...
Adding user `jungwungpark' to group `plugdev' ...
Adding user `jungwungpark' to group `users' ...
root@beaglebone:/home/debian# ls /home
debian  jungwung  jungwungpark
root@beaglebone:/home/debian# su
root@beaglebone:/home/debian# adduser jungwungpark sudo
Adding user `jungwungpark' to group `sudo' ...
Adding user jungwungpark to group sudo
Done.
```

We can exit using the command "# exit". With command adduser jungwungpark sudo, we became superuser.

Login as user you just made.

```
Debian GNU/Linux 7 beaglebone ttyGS0

BeagleBoard.org Debian Image 2015-11-12

Support/FAQ: http://elinux.org/Beagleboard:BeagleBoneBlack_Debian

default username:password is [debian:temppwd]

The IP Address for usb0 is: 192.168.7.2
beaglebone login: jungwungpark
Password:
Linux beaglebone 3.8.13-bone79 #1 SMP Tue Oct 13 20:44:55 UTC 2015 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.
```

Check network.

```
jungwungpark@beaglebone:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr f0:45:da:81:38:51
          inet addr:192.168.0.12  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::f245:daff:fe81:3851/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:55 errors:0 dropped:0 overruns:0 frame:0
          TX packets:99 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:9137 (8.9 KiB)  TX bytes:15917 (15.5 KiB)
          Interrupt:40

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

usb0      Link encap:Ethernet  HWaddr f0:45:da:81:38:50
          inet addr:192.168.7.2  Bcast:192.168.7.3  Mask:255.255.255.252
          inet6 addr: fe80::f245:daff:fe81:3850/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:32 errors:0 dropped:0 overruns:0 frame:0
          TX packets:40 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6022 (5.8 KiB)  TX bytes:10209 (9.9 KiB)
```

The IP address is 192.168.0.12.

Use remote login.

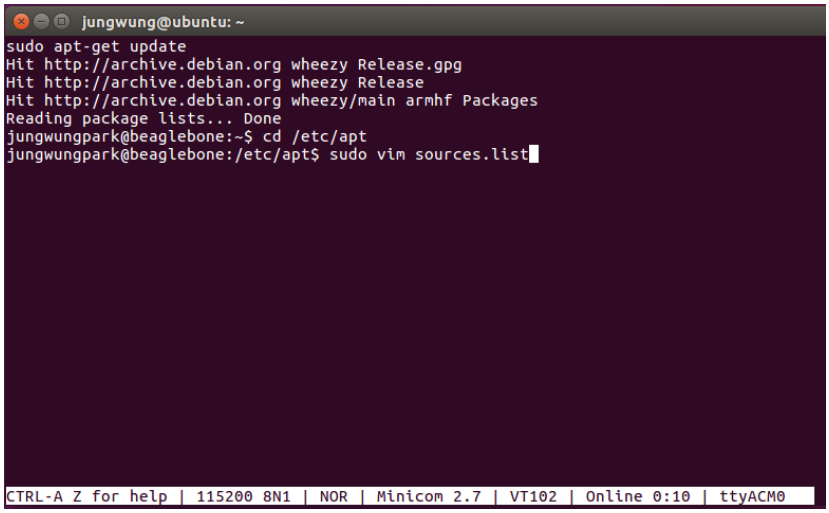
```
jungwung@ubuntu:~$ ssh jungwungpark@192.168.0.12
Debian GNU/Linux 7

BeagleBoard.org Debian Image 2015-11-12

Support/FAQ: http://elinux.org/Beagleboard:BeagleBoneBlack_Debian

default username:password is [debian:temppwd]

jungwungpark@192.168.0.12's password:
Last login: Thu Sep 16 14:55:13 2021 from ubuntu.local
```

Update package.

```
jungwungpark@beaglebone:~$ sudo apt-get update
[sudo] password for jungwungpark:
Get:1 http://repos.rcn-ee.com wheezy Release.gpg [833 B]
Ign http://ftp.us.debian.org wheezy Release.gpg
Ign http://security.debian.org wheezy/updates Release.gpg
Get:2 http://repos.rcn-ee.com wheezy Release [2,169 B]
Ign http://ftp.us.debian.org wheezy-updates Release.gpg
Ign http://security.debian.org wheezy/updates Release
Ign http://ftp.us.debian.org wheezy Release
Get:3 http://repos.rcn-ee.com wheezy/main armhf Packages [124 kB]
Ign http://ftp.us.debian.org wheezy-updates Release
Err http://security.debian.org wheezy/updates/main armhf Packages
  404  Not Found [IP: 151.101.130.132 80]
Err http://security.debian.org wheezy/updates/contrib armhf Packages
  404  Not Found [IP: 151.101.130.132 80]
Err http://security.debian.org wheezy/updates/non-free armhf Packages
  404  Not Found [IP: 151.101.130.132 80]
Err http://ftp.us.debian.org wheezy/main armhf Packages
  404  Not Found [IP: 208.80.154.15 80]
Err http://ftp.us.debian.org wheezy/contrib armhf Packages
  404  Not Found [IP: 208.80.154.15 80]
Err http://ftp.us.debian.org wheezy/non-free armhf Packages
  404  Not Found [IP: 208.80.154.15 80]
Err http://ftp.us.debian.org wheezy-updates/main armhf Packages
  404  Not Found [IP: 208.80.154.15 80]
Err http://ftp.us.debian.org wheezy-updates/contrib armhf Packages
  404  Not Found [IP: 208.80.154.15 80]
Err http://ftp.us.debian.org wheezy-updates/non-free armhf Packages
  404  Not Found [IP: 208.80.154.15 80]
Fetched 127 kB in 8s (14.7 kB/s)
W: Failed to fetch http://security.debian.org/dists/wheezy/updates/main/binary-armhf/Packages  404  N
t Found [IP: 151.101.130.132 80]

W: Failed to fetch http://security.debian.org/dists/wheezy/updates/contrib/binary-armhf/Packages  404
 Not Found [IP: 151.101.130.132 80]

W: Failed to fetch http://security.debian.org/dists/wheezy/updates/non-free/binary-armhf/Packages  40
  Not Found [IP: 151.101.130.132 80]

W: Failed to fetch http://ftp.us.debian.org/debian/dists/wheezy/main/binary-armhf/Packages  404  Not
ound [IP: 208.80.154.15 80]
```

```
jungwungpark@beaglebone:~$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages have been kept back:
  c9-core-installer
The following packages will be upgraded:
  libdrm-radeon1 libdrm2 nodejs nodejs-dev nodejs-legacy pastebinit
  rcn-ee-archive-keyring ti-pru-cgt-installer tiomapconf xinput-calibrator
10 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
Need to get 2,534 kB of archives.
After this operation, 1,143 kB of additional disk space will be used.
Do you want to continue [Y/n]? y
Get:1 http://repos.rcn-ee.com/debian/ wheezy/main libdrm2 armhf 2.4.66-2rcnee2~bpo70+20160202+1 [33.9
kB]
Get:2 http://repos.rcn-ee.com/debian/ wheezy/main libdrm-radeon1 armhf 2.4.66-2rcnee2~bpo70+20160202+1
 [30.0 kB]
Get:3 http://repos.rcn-ee.com/debian/ wheezy/main ti-pru-cgt-installer all 2.1.4-0rcnee0~bpo70+2017021
6+1 [13.3 kB]
Get:4 http://repos.rcn-ee.com/debian/ wheezy/main nodejs-dev armhf 0.10.42~dfsg-0rcnee2~bpo70+20160217
+1 [305 kB]
Get:5 http://repos.rcn-ee.com/debian/ wheezy/main nodejs armhf 0.10.42~dfsg-0rcnee2~bpo70+20160217+1 [
1,267 kB]
Get:6 http://repos.rcn-ee.com/debian/ wheezy/main pastebinit all 1.5-1~bpo70+20160303+1 [53.9 kB]
Get:7 http://repos.rcn-ee.com/debian/ wheezy/main xinput-calibrator armhf 0.7.5+git20140213-1rcnee1~bp
o70+20160129+1 [49.7 kB]
Get:8 http://repos.rcn-ee.com/debian/ wheezy/main nodejs-legacy all 0.10.42~dfsg-0rcnee2~bpo70+2016021
7+1 [86.4 kB]
Get:9 http://repos.rcn-ee.com/debian/ wheezy/main rcn-ee-archive-keyring all 2016.04.24~bpo70+20160424
+1 [5,858 B]
Get:10 http://repos.rcn-ee.com/debian/ wheezy/main tiomapconf armhf 1.72.0-git20160414-1~bpo70+2016041
4+1 [690 kB]
Fetched 2,534 kB in 5s (457 kB/s)
(Reading database ... 60401 files and directories currently installed.)
Preparing to replace libdrm2:armhf 2.4.65-3rcnee5~bpo70+20151109+1 (using .../libdrm2_2.4.66-2rcnee2~b
po70+20160202+1_armhf.deb) ...
Unpacking replacement libdrm2:armhf ...
Preparing to replace libdrm-radeon1:armhf 2.4.65-3rcnee5~bpo70+20151109+1 (using .../libdrm-radeon1_2.
4.66-2rcnee2~bpo70+20160202+1_armhf.deb) ...
Unpacking replacement libdrm-radeon1:armhf ...
Preparing to replace ti-pru-cgt-installer 2.1.1-1~bpo70+20150413+1 (using .../ti-pru-cgt-installer_2.1
.4-0rcnee0~bpo70+20170216+1_all.deb) ...
Unpacking replacement ti-pru-cgt-installer ...
Preparing to replace nodejs-dev 0.10.38~dfsg-1~bpo70+20150526+1 (using .../nodejs-dev_0.10.42~dfsg-0rc
```

With Debian 7.9, there is some error. As support for Debian 7.9 has ended, we need to change source.list file to use apt-get command.



We made every line as a comment in original source.list file and added one command at last :"deb http://archive.debian.org/devian wheezy main".

With this change, we could successfully update packages.

```
jungwung@ubuntu: ~
Reading package lists... Done
jungwungpark@beaglebone:/etc/apt$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be upgraded:
  base-files bind9-host cpio dpkg dpkg-dev gir1.2-gdkpixbuf-2.0 git git-core
  git-man isc-dhcp-client isc-dhcp-common libavcodec-dev libavcodec53
  libavformat-dev libavformat53 libavutil-dev libavutil51 libbind9-80 libc-bin
  libc-dev-bin libc6 libc6-dev libdns88 libdpkg-perl libfreetype6
  libfreetype6-dev libgcrypt11 libgd2-xpm libgdk-pixbuf2.0-0
  libgdk-pixbuf2.0-common libgdk-pixbuf2.0-dev libgif4 libgnutls26
  libgssapi-krb5-2 libgtk-3-0 libgtk-3-bin libgtk-3-common libimlib2 libisc84
  libisccc80 libisccfg82 libjasper-dev libjasper1 libk5crypto3 libkrb5-3
  libkrb5support0 libldap-2.4-2 liblwres80 libnspr4 libpixman-1-0
  libpixman-1-dev libpng12-0 libpng12-dev libpostproc52 librsvg2-2
  librsvg2-common libsmbclient libssh2-1 libssl-dev libssl-doc libssl1.0.0
  libswscale-dev libswscale2 libwbclient0 libxapian22 libxml2 libxml2-dev
  libxml2-utils linux-libc-dev locales multiarch-support openssh-client
  openssh-server openssl perl perl-base perl-modules sudo tzdata xscreensaver
  xscreensaver-data
81 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 66.6 MB of archives.
After this operation, 5,292 kB disk space will be freed.
Do you want to continue [Y/n]? y
Get:1 http://archive.debian.org/debian/ wheezy/main base-files armhf 7.1wheezy11 [67.1
```

Shut down using the command.

```
jungwungpark@beaglebone:~$ sudo shutdown -h now
[sudo] password for jungwungpark:
Sorry, try again.
[sudo] password for jungwungpark:

Broadcast message from root@beaglebone (ttyGS0) (Thu Sep 16 15:49:43 2021):
The system is going down for system halt NOW!
```

Step 3. Test Cross-compile on PC Ubuntu

Install cross-compiler for ARM in the PC.

```
jungwung@ubuntu:~$ sudo apt-get install gcc-arm-linux-gnueabihf
[sudo] password for jungwung:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils-arm-linux-gnueabihf cpp-5-arm-linux-gnueabihf
  cpp-arm-linux-gnueabihf gcc-5-arm-linux-gnueabihf
  gcc-5-arm-linux-gnueabihf-base gcc-5-cross-base libasan2-armhf-cross
  libatomic1-armhf-cross libc6-armhf-cross libc6-dev-armhf-cross
  libgcc-5-dev-armhf-cross libgcc1-armhf-cross libgomp1-armhf-cross
  libstdc++6-armhf-cross libubsan0-armhf-cross linux-libc-dev-armhf-cross
Suggested packages:
  binutils-doc gcc-5-locales cpp-doc gcc-5-multilib-arm-linux-gnueabihf
  gcc-5-doc libgcc1-dbg-armhf-cross libgomp1-dbg-armhf-cross
  libitm1-dbg-armhf-cross libatomic1-dbg-armhf-cross libasan2-dbg-armhf-cross
  liblsan0-dbg-armhf-cross libtsan0-dbg-armhf-cross libubsan0-dbg-armhf-cross
  libcilkrts5-dbg-armhf-cross libmpx0-dbg-armhf-cross
  libquadmath0-dbg-armhf-cross autoconf automake libtool flex bison
  gdb-arm-linux-gnueabihf gcc-doc
The following NEW packages will be installed:
  binutils-arm-linux-gnueabihf cpp-5-arm-linux-gnueabihf
  cpp-arm-linux-gnueabihf gcc-5-arm-linux-gnueabihf
```

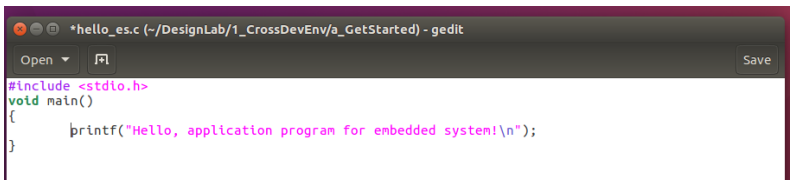
```
jungwung@ubuntu:~$ arm-linux-gnueabihf-gcc --version
arm-linux-gnueabihf-gcc (Ubuntu/Linaro 5.4.0-6ubuntu1~16.04.9) 5.4.0 20160609
Copyright (C) 2015 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

Cross-compiler is installed.

Make a working directory.

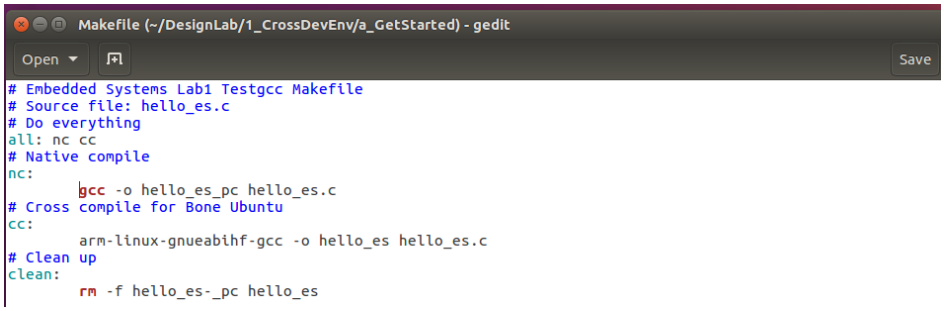
```
jungwung@ubuntu:~$ mkdir -p ~/DesignLab/1_CrossDevEnv/a_GetStarted
jungwung@ubuntu:~$ cd ~/D
DesignLab/ Desktop/   Documents/ Downloads/
jungwung@ubuntu:~$ cd ~/DesignLab/1_CrossDevEnv/a_GetStarted/
```

Edit the example program and cross-compile the program.




```
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/a_GetStarted$ gedit hello_es.c
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/a_GetStarted$ ls
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/a_GetStarted$ arm-linux-gnueabihf-gcc -o hello_es hello_es.c
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/a_GetStarted$ ls -la
total 24
drwxrwxr-x 2 jungwung jungwung 4096 Sep 17 02:15 .
drwxrwxr-x 3 jungwung jungwung 4096 Sep 17 02:12 ..
-rwxrwxr-x 1 jungwung jungwung 8248 Sep 17 02:15 hello_es
-rw-rw-r-- 1 jungwung jungwung   97 Sep 17 02:14 hello_es.c
```

Use makefile.




```
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/a_GetStarted$ gedit Makefile
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/a_GetStarted$ make clean
rm -f hello_es-_pc hello_es
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/a_GetStarted$ ls
hello_es.c  Makefile
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/a_GetStarted$ make all
gcc -o hello_es_pc hello_es.c
arm-linux-gnueabihf-gcc -o hello_es hello_es.c
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/a_GetStarted$ ls
hello_es  hello_es.c  hello_es_pc  Makefile
```

"make clean" can remove all the compiled files.

"make all" can compile all the files to run on the PC and Bone each.

Try to run on PC.

```
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/a_GetStarted$ ./hello_es_pc
Hello, application program for embedded system!
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/a_GetStarted$ ./hello_es
bash: ./hello_es: cannot execute binary file: Exec format error
```

On PC, we can only run "hello_es_pc". "hello_es" is not executable as it is binary for ARM CPU Debian in Beaglebone, not Ubuntu AMD64.

Download.

On Bone console, make a working directory.

```
jungwungpark@beaglebone:~$ mkdir -p ~/test_scp
jungwungpark@beaglebone:~$ cd ~/test_scp
```

On PC terminal, type the commands.

```
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/a_GetStarted$ scp hello_es jungwungpar
k@192.168.0.12:/home/jungwungpark/test_scp
Debian GNU/Linux 7

BeagleBoard.org Debian Image 2015-11-12

Support/FAQ: http://elinux.org/Beagleboard:BeagleBoneBlack_Debian

default username:password is [debian:temppwd]

jungwungpark@192.168.0.12's password:
hello_es                                      100% 8248     8.1KB/s   00:00
```

```
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/a_GetStarted$ scp hello_es_pc jungwung
park@192.168.0.12:/home/jungwungpark/test_scp
Debian GNU/Linux 7

BeagleBoard.org Debian Image 2015-11-12

Support/FAQ: http://elinux.org/Beagleboard:BeagleBoneBlack_Debian

default username:password is [debian:temppwd]

jungwungpark@192.168.0.12's password:
hello_es_pc                                   100% 8608     8.4KB/s   00:00
```

On Bone console, check the files. Two files were transmitted correctly.

```
jungwungpark@beaglebone:~/test_scp$ ls
hello_es  hello_es_pc
```

Run on Beaglebone.

```
jungwungpark@beaglebone:~/test_scp$ ./hello_es
Hello, application program for embedded system!
jungwungpark@beaglebone:~/test_scp$ ./hello_es_pc
-bash: ./hello_es_pc: cannot execute binary file
```

On Beaglebone, we can only run "hello_es". "hello_es_pc" is not executable in Beaglebone. "hello_es_pc" is executable only in amd64 architecture and "hello_es" is executable in ARM architecture. Cross-compile enable us to compile arm-executable files on the amd64-based Ubuntu.

Step 4. Setup NFS

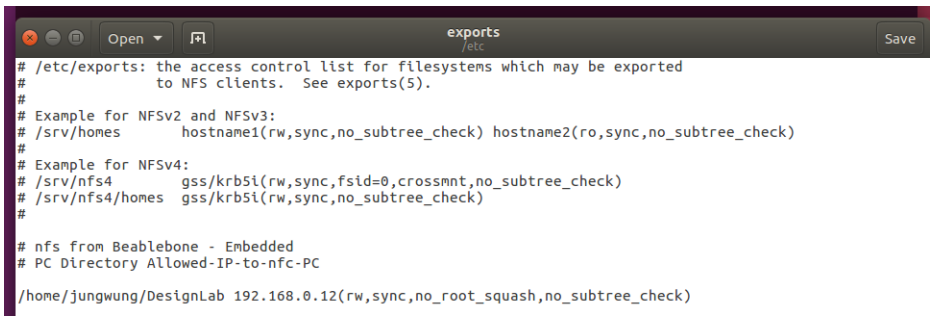
Install NFS server on PC using the command.

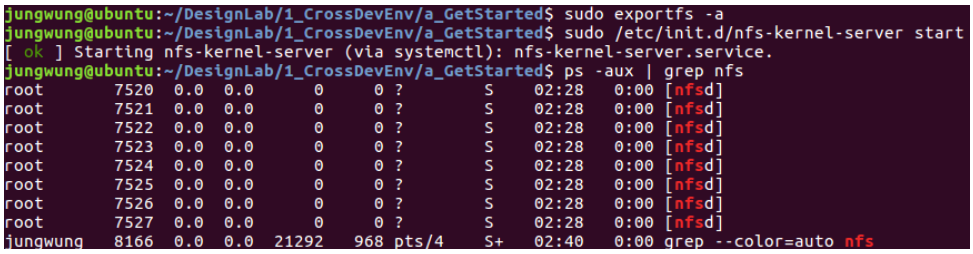
$ sudo apt-get install nfs-kernel-server

Configure NFS and start PC NFS server.

Write the command and edit the file.

$ sudo gedit /etc/exports



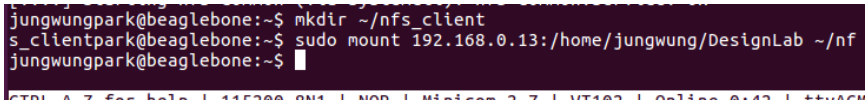


We checked that NFS started.

Install NFS client in Bone using the command.

sudo apt-get install nfs-common

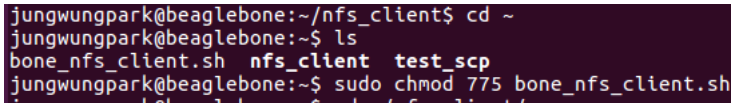
Make the mount point and start Beaglebone NFS client.



Due to the bug when shortening the command window, the last texts were overlapped to the front.

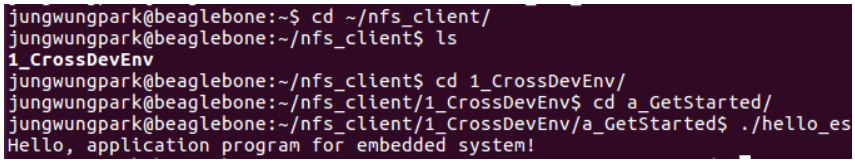
For later use, edit ~/bone_nfs_client.sh using vi.

sudo vi ~/bone_nfs_client.sh



Now, we can use "# ~/bone_nfs_client.sh" to start Beaglebone NFS client.

Run hello_es using NFS.



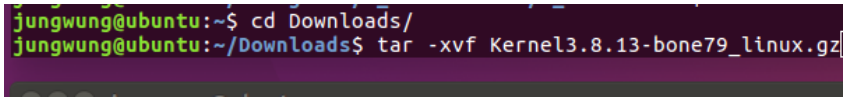
It works well.

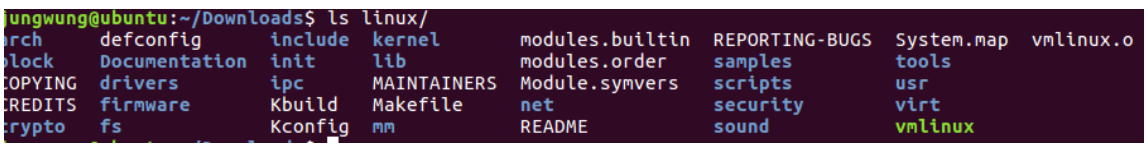
Step 5. Prepare Module Build

Get Kernel source from Web for RoboCam.

We got "Kernel3.8.13-bone79_linux.tgz".

Install Kernel source.





Configure cross compile environment.

*cde_bd_k3813_bone79 (~/Downloads) - gedit

Open Save

```
#!/bin/bash
### Shell script cde_bd_k3813-bone79
### Setup cross-compile environment for Bone-Debian with Kernel 3.8.13-bone79 ### Usage: source sh/
cde_bd_k3813-bone79
## Line starting with '#' means comment line.
## Set MACHINE
MACHINE=beaglebone
## Set SYSROOTSDIR & STAGEDIR
SYSROOTSDIR=/usr
STAGEDIR=${SYSROOTSDIR}

## Set CROSSBINDIR (where cross compiler exists)
CROSSBINDIR=/usr/bin
## Set KERNELDIR (where the Linux kernel source is located)
## NOTE: This path to KernelDir should be exact.
export KERNELDIR=/home/jungwung/Downloads/linux
## Set PATH
PATH=${CROSSBINDIR}:${PATH}
unset CFLAGS CPPFLAGS CXXFLAGS LDFLAGS MACHINE
export ARCH="arm"
export CROSS_COMPILE="arm-linux-gnueabihf-"
export CC="arm-linux-gnueabihf-gcc"
export LD="arm-linux-gnueabihf-ld"
export STRIP="arm-linux-gnueabihf-strip"
echo "Set cross-development environment for Beaglebone Debian (3.8.13-bone79  kernel)."
```

```
jungwung@ubuntu:~/Downloads$ gedit cde_bd_k3813_bone79
jungwung@ubuntu:~/Downloads$ source cde_bd_k3813_bone79
Set cross-development environment for Beaglebone Debian (3.8.13-bone79  kernel).
jungwung@ubuntu:~/Downloads$
```

set KERNELDIR as /home/jungwung/Downloads/linux

Step 6. MODULE COMPILE

Make another working directory and edit Makefile.

```
jungwung@ubuntu:~$ mkdir -p ~/DesignLab/1_CrossDevEnv/b_ModuleBuild
jungwung@ubuntu:~$ cd ~/DesignLab/1_CrossDevEnv/b_ModuleBuild/
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/b_ModuleBuild$ gedit Makefile
```

*Makefile (~/DesignLab/1_CrossDevEnv/b_ModuleBuild) - gedit

Open Save

```
# Bone-Debian cross-compile module makefile
ifneq ($(KERNELRELEASE),)
        obj-m := hellomod.o
        else
        SUBDIRS := $(shell pwd)

default:
ifeq ($(strip $(KERNELDIR)),)
        $(error "KERNELDIR is undefined!")
else
        $(MAKE) -C $(KERNELDIR) M=$(SUBDIRS) modules
endif

clean:
        rm -rf *~ *.ko *.o *.mod.c modules.order Module.symvers .pwm* .tmp_versions
endif
```

Edit hellomod.c.

```
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/b_ModuleBuild$ gedit hellomod.c
```

*hellomod.c (~/DesignLab/1_CrossDevEnv/b_ModuleBuild) - gedit

Open

```
#include <linux/module.h> /* Needed by all modules */
static int __init hellomod_init(void) // init_module()
{
        printk("Hello, hellomod!\n");
        return 0;
}
static void __exit hellomod_exit(void) // cleanup_module()
{
        printk("Goodbye, hellomod!\n");
}
module_init(hellomod_init);
module_exit(hellomod_exit);
MODULE_AUTHOR("Christoper Hallinan");
MODULE_DESCRIPTION("Hello Module Example");
MODULE_LICENSE("GPL");
```

Use make clean and make.

```
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/b_ModuleBuild$ make clean
rm -rf *~ *.ko *.o *.mod.c modules.order Module.symvers .pwm* .tmp_versions
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/b_ModuleBuild$ make
make -C /home/jungwung/Downloads/linux M=/home/jungwung/DesignLab/1_CrossDevEnv/b_ModuleBuild modules
make[1]: Entering directory '/home/jungwung/Downloads/linux'
  CC [M]  /home/jungwung/DesignLab/1_CrossDevEnv/b_ModuleBuild/hellomod.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/jungwung/DesignLab/1_CrossDevEnv/b_ModuleBuild/hellomod.mod.o
  LD [M]  /home/jungwung/DesignLab/1_CrossDevEnv/b_ModuleBuild/hellomod.ko
make[1]: Leaving directory '/home/jungwung/Downloads/linux'
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/b_ModuleBuild$ ls
hellomod.c   hellomod.mod.c  hellomod.o  modules.order
hellomod.ko  hellomod.mod.o  Makefile    Module.symvers
```

Insert module on Beaglebone.

Go to the working directory.

cd ~/nfs_client/1_CrossDevEnv/b_ModuleBuild

Become superuser.

su

We can insert module and check the module.

```
d hellomod.kone:/home/jungwungpark/nfs_client/1_CrossDevEnv/b_ModuleBuild# insmd
Module                  Size  Used bys_client/1_CrossDevEnv/b_ModuleBuild# lsmod
hellomod                 775  0
nfsd                  187513  2
g_multi                50407  2
libcomposite           15028  1 g_multi
omap_rng                4062  0
mt7601Usta            458758  0
```

We can check output using "# dmesg" command.

```
[ 2438.290489] net eth0: phy found : id is : 0x7c0f1
[ 2438.290527] libphy: PHY 4a101000.mdio:01 not found
[ 2438.295602] net eth0: phy 4a101000.mdio:01 not found on slave 1
[ 2438.311421] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
[ 2440.298433] libphy: 4a101000.mdio:00 - Link is Up - 100/Full
[ 2440.298555] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
[ 3153.554158] Installing knfsd (copyright (C) 1996 okir@monad.swb.de).
[ 5134.020000] Hello, hellomod!
```

We can check output using "# modinfo hellomod.ko" command.

```
filename:       /home/jungwungpark/nfs_client/1_CrossDevEnv/b_ModuleBuild/hellomod.ko
license:        GPL
description:    Hello Module Example
author:         Christoper Hallinan
srcversion:     8AE57EE0AB2FDCB308B7263
depends:
vermagic:       3.8.13 SMP mod_unload modversions ARMv7 thumb2 p2v8
```

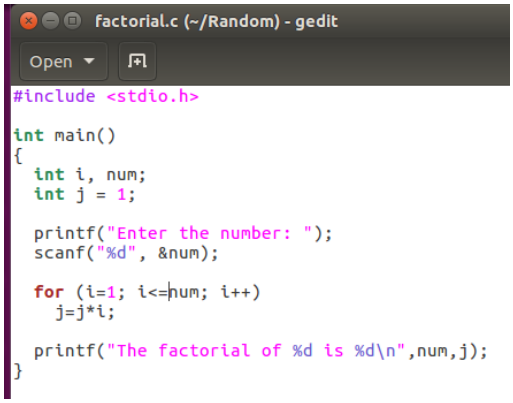
Remove the module.

We can remove module using "# rmmod hellomod" command and check output using "# dmesg" command.

```
[ 2440.298433] libphy: 4a101000.mdio:00 - Link is Up - 100/Full
[ 2440.298555] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
[ 3153.554158] Installing knfsd (copyright (C) 1996 okir@monad.swb.de).
[ 5134.020000] Hello, hellomod!
[ 5239.302012] Goodbye, hellomod!
```

Step 7. Exercise Gnu Debugger gdb

Make factorial.c




```
#include <stdio.h>

int main()
{
  int i, num;
  int j = 1;

  printf("Enter the number: ");
  scanf("%d", &num);

  for (i=1; i<=num; i++)
    j=j*i;

  printf("The factorial of %d is %d\n",num,j);
}
```

Compile factorial.c and start gdb

```
heejin@heejin-HP-Compaq-6531s:~/Random$ gcc -o factorial factorial.c -ggdb
heejin@heejin-HP-Compaq-6531s:~/Random$ gdb factorial
GNU gdb (Ubuntu 7.11.1-0ubuntu1~16.5) 7.11.1
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "i686-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from factorial...done.
(gdb)
```

Test gdb commands.

```
(gdb) l 10
5         int i, num;
6         int j = 1;
7
8         printf("Enter the number: ");
9         scanf("%d", &num);
10
11        for (i=1; i<=num; i++)
12          j=j*i;
13
14        printf("The factorial of %d is %d\n",num,j);
(gdb) b 12
Breakpoint 1 at 0x804850b: file factorial.c, line 12.
```

"list" command shows the code.

"break number" command sets a break point in line "number".

```
(gdb) r
Starting program: /home/heejin/Random/factorial
Enter the number: 4

Breakpoint 1, main () at factorial.c:12
12          j=j*i;
(gdb) p j
$1 = 1
(gdb) c
Continuing.

Breakpoint 1, main () at factorial.c:12
12          j=j*i;
```

"run" command runs the program.

"print variable" command shows the value of variable in that time.

```
(gdb) c
Continuing.

Breakpoint 1, main () at factorial.c:12
12          j=j*i;
(gdb) p j
$3 = 6
(gdb) n
11        for (i=1; i<=num; i++)
(gdb) p j
$4 = 24
(gdb) s
14        printf("The factorial of %d is %d\n",num,j);
```

We can go to next break point using "continue" command.

We can go next instruction using "next" and "step" commands.

However, "step" can go into function and "next" cannot.

```
(gdb) c
Continuing.
The factorial of 4 is 24
[Inferior 1 (process 3263) exited normally]
(gdb) quit
heejin@heejin-HP-Compaq-6531s:~/Random$
```

We can quit gdb using "quit" command.

Step 8. Test Reaction Timer Components

Test test_rand_pc on PC

```
jungwung@ubuntu: ~/DesignLab/1_CrossDevEnv/c_ReactionTimer/test_rand
hellomod.ko  hellomod.mod.o  Makefile    Module.symvers
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/b_ModuleBuild$ cd ..
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv$ cd c_ReactionTimer/
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/c_ReactionTimer$ cd test_rand/
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/c_ReactionTimer/test_rand$ make
gcc -o test_rand_pc test_rand.c
arm-linux-gnueabihf-gcc -o test_rand test_rand.c
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/c_ReactionTimer/test_rand$ ./test_rand_pc
Rand(1 to 10) Avg= 5.92142, Min= 1.14671, Max= 9.99032
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/c_ReactionTimer/test_rand$ ./test_rand_pc
Rand(1 to 10) Avg= 5.92142, Min= 1.14671, Max= 9.99032
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/c_ReactionTimer/test_rand$ ./test_rand_pc
Rand(1 to 10) Avg= 5.92142, Min= 1.14671, Max= 9.99032
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/c_ReactionTimer/test_rand$ ./test_rand_pc
Rand(1 to 10) Avg= 5.92142, Min= 1.14671, Max= 9.99032
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/c_ReactionTimer/test_rand$ ./test_rand_pc
Rand(1 to 10) Avg= 5.92142, Min= 1.14671, Max= 9.99032
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/c_ReactionTimer/test_rand$ ./test_rand_pc
Rand(1 to 10) Avg= 5.92142, Min= 1.14671, Max= 9.99032
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/c_ReactionTimer/test_rand$ ./test_rand_pc
Rand(1 to 10) Avg= 5.92142, Min= 1.14671, Max= 9.99032
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/c_ReactionTimer/test_rand$ ./test_rand_pc
Rand(1 to 10) Avg= 5.92142, Min= 1.14671, Max= 9.99032
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/c_ReactionTimer/test_rand$
```

"./test_rand_pc" program produces same results, not random, because seed for rand() function is fixed. Let's change the seed using srand() function.

We modified test_rand.c as below.

test_rand.c (~/DesignLab/1_CrossDevEnv/c_ReactionTimer/test_rand) - gedit

Open

```c
/*
        Program test_rand.c
        Test function rand()
        Programmed by Byung Kook Kim, Feb. 1, 2017
*/

// Declare include hader files
#include <stdio.h>
#include <stdlib.h>     // rand(), srand()
#include <time.h>       // time()

int main(void)
{
        // Declare variables to be used
        int k;  // Loop index
        double Lower, Upper, Range;     // Random number range
        double rn[100]; // Random number array
        // 1. Init variables
        Lower = 1.;
        Upper = 10.;
        Range = Upper - Lower;

        srand((unsigned int)time(NULL));

        // 2. Loop 100 times
        for (k=0; k<100; ++k) {
                // Generate a random number [Lower, Upper] to rn[k]
                rn[k] = Lower + Range*rand()/RAND_MAX;
        }
        // 3. Compute avg, min, and max of rm[k]
        double Min = 1e9;
        double Max = -1e9;
        double sum = 0.;
        for (k=0; k<100; ++k) {
                sum += rn[k];
                if (rn[k] < Min) Min = rn[k];
                if (rn[k] > Max) Max = rn[k];
        }
        double Avg = sum/100.;

        // 4. Print result
        printf("Rand(1 to 10) Avg= %g, Min= %g, Max= %g\n", Avg, Min, Max);
        return 0;
}
```

```
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/c_ReactionTimer/test_rand$ make clean
rm -f test_rand test_rand_pc
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/c_ReactionTimer/test_rand$ make
gcc -o test_rand_pc test_rand.c
arm-linux-gnueabihf-gcc -o test_rand test_rand.c
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/c_ReactionTimer/test_rand$ ./test_rand_pc
Rand(1 to 10) Avg= 5.51875, Min= 1.06037, Max= 9.72918
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/c_ReactionTimer/test_rand$ ./test_rand_pc
Rand(1 to 10) Avg= 6.02724, Min= 1.18065, Max= 9.92953
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/c_ReactionTimer/test_rand$ ./test_rand_pc
Rand(1 to 10) Avg= 5.89755, Min= 1.20446, Max= 9.87434
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/c_ReactionTimer/test_rand$ ./test_rand_pc
Rand(1 to 10) Avg= 5.61072, Min= 1.014, Max= 9.96267
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/c_ReactionTimer/test_rand$ ./test_rand_pc
Rand(1 to 10) Avg= 5.26204, Min= 1.0144, Max= 9.91669
```

The results came out randomly. We used time() function as a input parameter of srand() fuction. As the minimum unit in time() function is second, we need to execute “test_rand_pc” with the time term larger than 1 second.

Test test_gettimeofday_usleep_pc on PC

```
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/c_ReactionTimer/test_gettimeofday$ make clean
rm -f test_gettimeofday_usleep test_gettimeofday_usleep_pc
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/c_ReactionTimer/test_gettimeofday$ make
gcc -o test_gettimeofday_usleep_pc test_gettimeofday_usleep.c
arm-linux-gnueabihf-gcc -o test_gettimeofday_usleep test_gettimeofday_usleep.c
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/c_ReactionTimer/test_gettimeofday$ ls
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/c_ReactionTimer/test_gettimeofday$ ./test_gettimeofday_usleep_pc
Elapsed_time= 2000.73 ms.
Elapsed_time= 2000.52 ms.
Elapsed_time= 2000.55 ms.
```

The results came out in close to 2 seconds.

Test test_getch_pc on PC

--Number & lower alphabet & upper alphabet

```
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/c_ReactionTimer/test_getch$ make clean
rm -f test_getch test_getch_pc
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/c_ReactionTimer/test_getch$ make
gcc -g -o test_getch_pc getche.c test_getche.c
arm-linux-gnueabihf-gcc -o test_getch getche.c test_getche.c
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/c_ReactionTimer/test_getch$ ./test_getch_pc
a, 97, 61h
b, 98, 62h
c, 99, 63h
d, 100, 64h
e, 101, 65h
f, 102, 66h
g, 103, 67h
h, 104, 68h
i, 105, 69h
j, 106, 6ah
1, 49, 31h
2, 50, 32h
3, 51, 33h
4, 52, 34h
5, 53, 35h
6, 54, 36h
```

The lower case letters increase by one from 97 in alphabetical order.

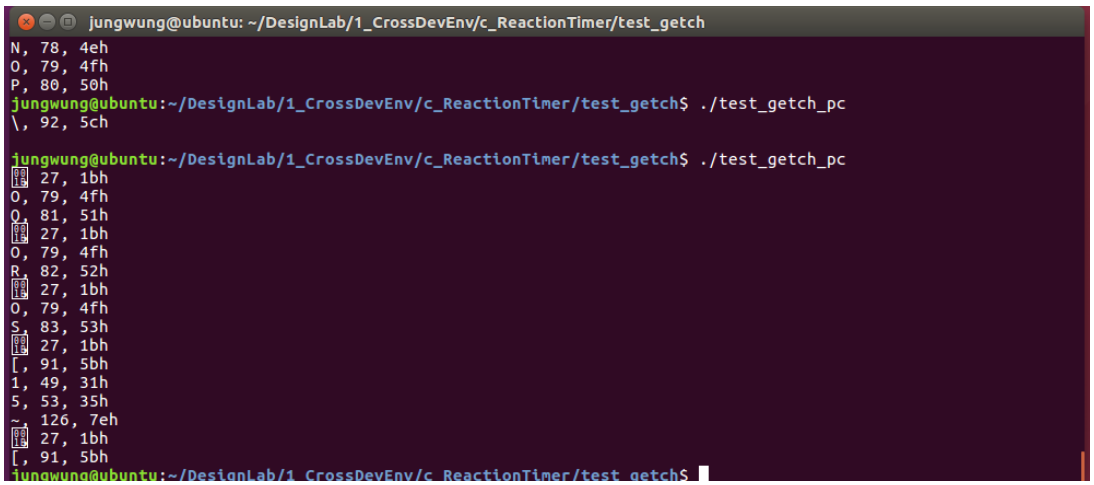
The numbers increase by one from 49.

```
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/c_ReactionTimer/test_getch$ ./test_getch_pc
A, 65, 41h
B, 66, 42h
C, 67, 43h
D, 68, 44h
E, 69, 45h
F, 70, 46h
G, 71, 47h
H, 72, 48h
I, 73, 49h
J, 74, 4ah
K, 75, 4bh
L, 76, 4ch
M, 77, 4dh
N, 78, 4eh
O, 79, 4fh
P, 80, 50h
```

The upper case letters increase by one from 65 in alphabetical order.

This result comes from the ASCII code. For example, ASCII value for a is 97 and its hex form is 0x61. For the number & lower alphabet & upper alphabet, “test_getch_pc” produced a good result.

-- Function keys

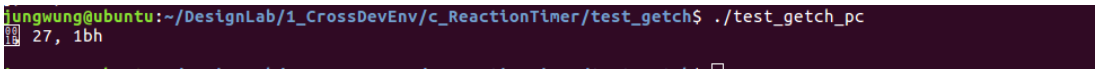


Above image shows the result when we pushed F1 to F6.

F1 did not show any results, instead it showed terminal help page. It is because default keyboard shortcut for terminal help page is F1 key.

For F2 to F6, each key showed 3 results at once. For example, F2 showed 27, 79, 81 and leftmost value is strange. It is because ASCII code does not contain any information about function keys. Every i/o system is abstracted as stream of characters. Thus, keyboard is also abstracted as a stream of characters. If we press function keys, keyboard will send some stream of characters to Ubuntu, but the data is not meaningful in ASCII code encoding, which results in showing a strange output.

- ESC & Ctrl + C



ASCII code has a information about ESC key. ESC key produced some strange alphabet, 27, and 1bh.

However, when we pressed Ctrl+C key, program stopped. It's because Ctrl+C ket sends a signal to process and the default action is to terminate process.

Test test_rand on Bone

```
_rand# ./test_randome/jungwungpark/nfs_client/1_CrossDevEnv/c_ReactionTimer/test_
Rand(1 to 10) Avg= 5.40768, Min= 1.05992, Max= 9.99733
_rand# aglebone:/home/jungwungpark/nfs_client/1_CrossDevEnv/c_ReactionTimer/test_
_rand# ./test_randome/jungwungpark/nfs_client/1_CrossDevEnv/c_ReactionTimer/test_
Rand(1 to 10) Avg= 6.05708, Min= 1.32201, Max= 9.98721
_rand# ./test_randome/jungwungpark/nfs_client/1_CrossDevEnv/c_ReactionTimer/test_
Rand(1 to 10) Avg= 5.28894, Min= 1.14769, Max= 9.92859
_rand# ./test_randome/jungwungpark/nfs_client/1_CrossDevEnv/c_ReactionTimer/test_
Rand(1 to 10) Avg= 5.76566, Min= 1.03479, Max= 9.94936
_rand# ./test_randome/jungwungpark/nfs_client/1_CrossDevEnv/c_ReactionTimer/test_
Rand(1 to 10) Avg= 5.16356, Min= 1.08344, Max= 9.98669
_rand# ./test_randome/jungwungpark/nfs_client/1_CrossDevEnv/c_ReactionTimer/test_
Rand(1 to 10) Avg= 5.58231, Min= 1.04518, Max= 9.93013
```

Due to the bug when shortening the command window, the last texts were overlapped to the front.

The results came out randomly.

Test test_gettimeofday_usleep on Bone

```
_gettimeofday# ./test_gettimeofday_usleepient/1_CrossDevEnv/c_ReactionTimer/test_
Elapsed_time= 2000.27 ms.
Elapsed_time= 2000.19 ms.
Elapsed_time= 2000.19 ms.
 gettimeofday# :/home/jungwungpark/nfs_client/1_CrossDevEnv/c_ReactionTimer/test
```

The results came out in close to 2 seconds.

Test test_getch on Bone

--Number & lower alphabet & upper alphabet

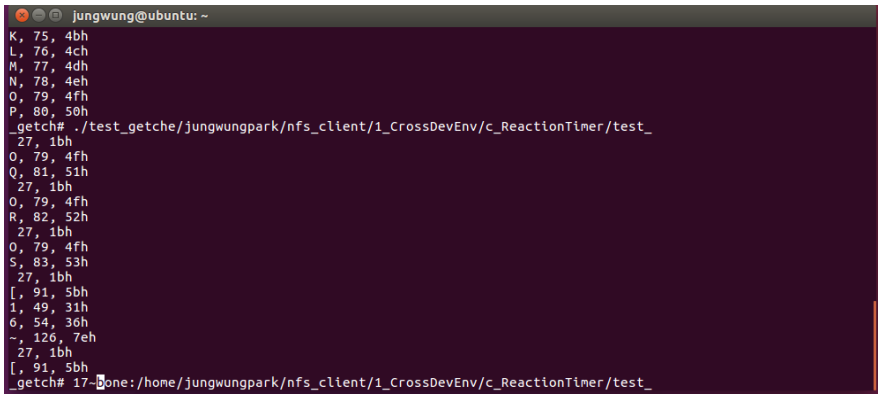
```
_getch# ./test_getche/jungwungpark/nfs_client/1_CrossDevEnv/c_ReactionTimer/test_
a, 97, 61h
b, 98, 62h
c, 99, 63h
d, 100, 64h
e, 101, 65h
f, 102, 66h
g, 103, 67h
h, 104, 68h
i, 105, 69h
j, 106, 6ah
1, 49, 31h
2, 50, 32h
3, 51, 33h
4, 52, 34h
5, 53, 35h
6, 54, 36h
```

The lower case letters increase by one from 97 in alphabetical order.

The numbers increase by one from 49.

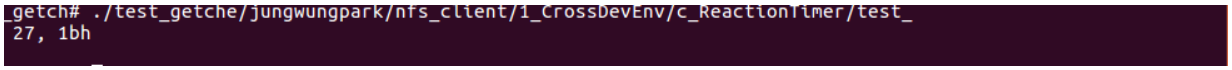
The upper case letters increase by one from 65 in alphabetical order.

-- Function keys



Pressed F1 to F6. Same result with PC.

- ESC & Ctrl + C



Same result with PC.

Step 9. Test reaction_timer

Test reaction_timer on PC

```
Type the 'f' character: f
Correct response f in 481.445ms.
Type the 'f' character: f
Correct response f in 441.604ms.
Type the 'j' character: f
InCorrect response f in 302.184ms.
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/c_ReactionTimer/test_reaction$ ./reaction_pc
Type the 'j' character: f
InCorrect response f in 341.037ms.
j
Type the 'j' character: j
Correct response j in 0.004ms.
Type the 'f' character:

InCorrect response
 in 0.005ms.
jungwung@ubuntu:~/DesignLab/1_CrossDevEnv/c_ReactionTimer/test_reaction$ ./reaction_pc
Type the 'f' character: f
Correct response f in 660.577ms.
Type the 'j' character: j
Correct response j in 564.093ms.
Type the 'f' character: f
Correct response f in 571.708ms.
```

If the input is incorrect, an incorrect message is displayed and reaction rate is measured.

If the input is correct, a correct message is displayed and reaction rate is measured.

Test reaction_timer on Bone

```
_reaction# ./reaction/jungwungpark/nfs_client/1_CrossDevEnv/c_ReactionTimer/test_
Type the 'f' character: f
Correct response f in 671.646ms.
Type the 'j' character: j
Correct response j in 378.624ms.
Type the 'j' character: j
Correct response j in 646.529ms.
_reaction# ./reaction/jungwungpark/nfs_client/1_CrossDevEnv/c_ReactionTimer/test_
Type the 'f' character: f
Correct response f in 495.266ms.
Type the 'j' character: j
Correct response j in 473.238ms.
Type the 'j' character: j
Correct response j in 453.833ms.
```

```
jungwungpark@beaglebone:~/nfs_client/1_CrossDevEnv/c_ReactionTimer/test_reaction
$ ./reaction
Type the 'f' character: f
Correct response f in 1683.960ms.
Type the 'f' character: f
Correct response f in 300.365ms.
Type the 'j' character: f
InCorrect response f in 402.727ms.
jungwungpark@beaglebone:~/nfs_client/1_CrossDevEnv/c_ReactionTimer/test_reaction
$
```

If the input is incorrect, an incorrect message is displayed and reaction rate is measured.

If the input is correct, a correct message is displayed and reaction rate is measured.

Works well!