Embedded System Design
Practice 2

TaeWook Kim & SeokHyun Hong
Hanyang University

Necessary file download

- **download “embedded.zip” from LMS**

- **List of files**
  - PL2303_Prolific_DriverInstaller (Windows)
  - toolchain-s5pc1xx.tar.gz
  - u-boot-1.3.4.tar.gz
  - vpos.bin
  - vpos.zip

Contents

1. U-Boot booting
2. Install TFTP, ARM Cross compiler
3. U-Boot porting
4. VPOS booting

U-BOOT

Bootloader

- **What is bootloader?**
  - Initialize system hardware
  - Load OS image into memory and branch to OS startup routine
  - The first program that runs when the system is powered on

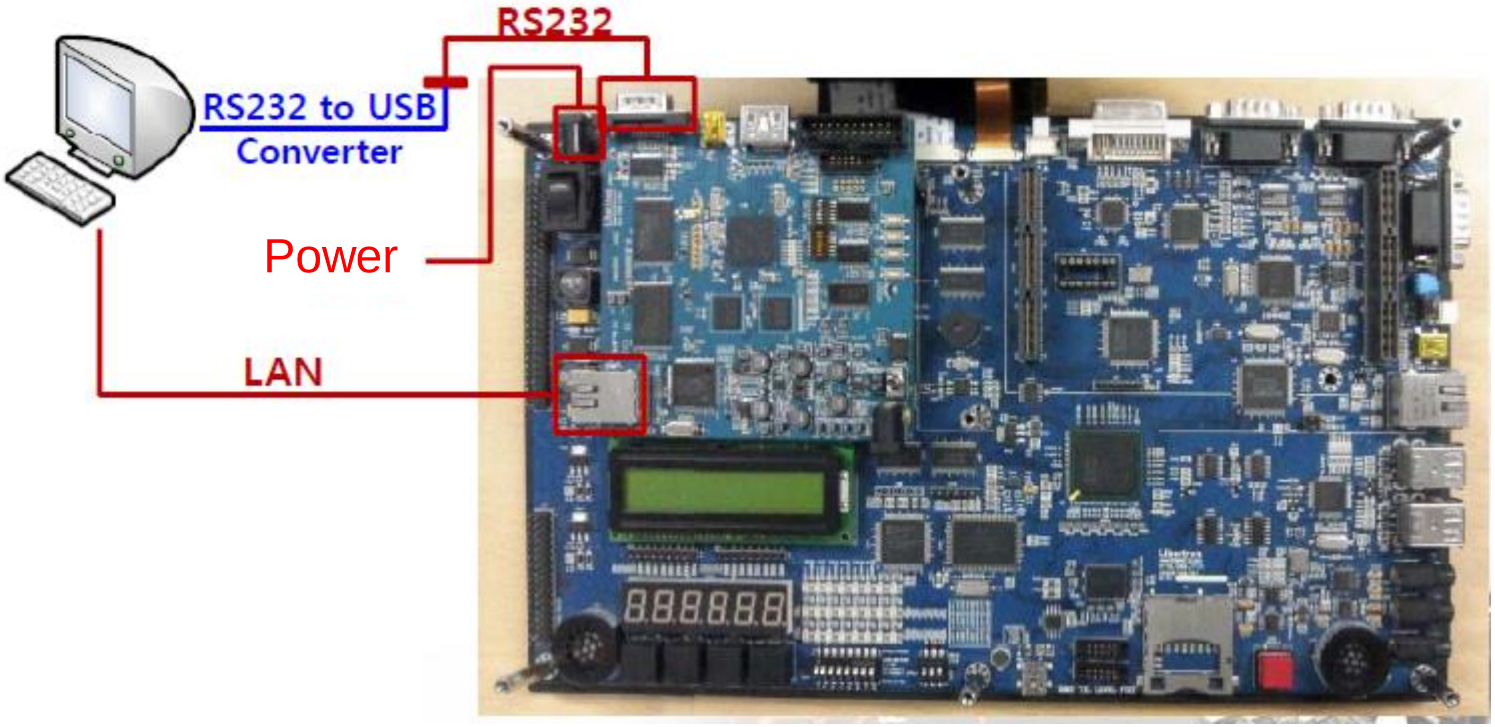
- **Location of the bootloader**
  - Generally, the physical address 0
  - Located in static memory such as ROM, Flash ROM, SRAM
- **Features of the bootloader**
  - Memory initialization
  - Hardware initialization

U-Boot

- **What is U-Boot?**
  - Universal Bootloader
  - Bootloader for embedded boards based on PowerPC and ARM
  - Widely used for bootloader of Linux on embedded board
- **Advantages**
  - Ported to various platforms (ARM, MIPS, x86, etc.)
  - Clean code, good structure
  - Easy configuration

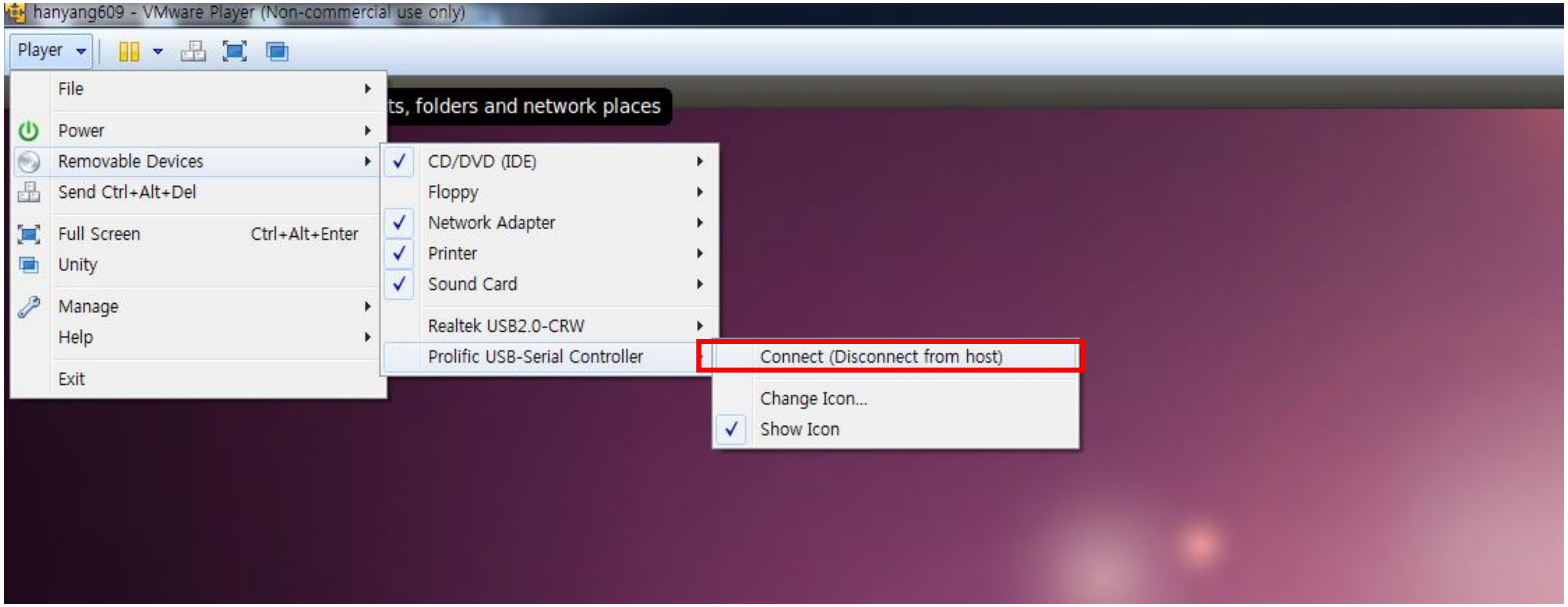
Serial Cable Connection

- **Serial connection**
  - Connect power cable to board
  - Connect serial cable to board and pc using converter



Serial Cable Connection

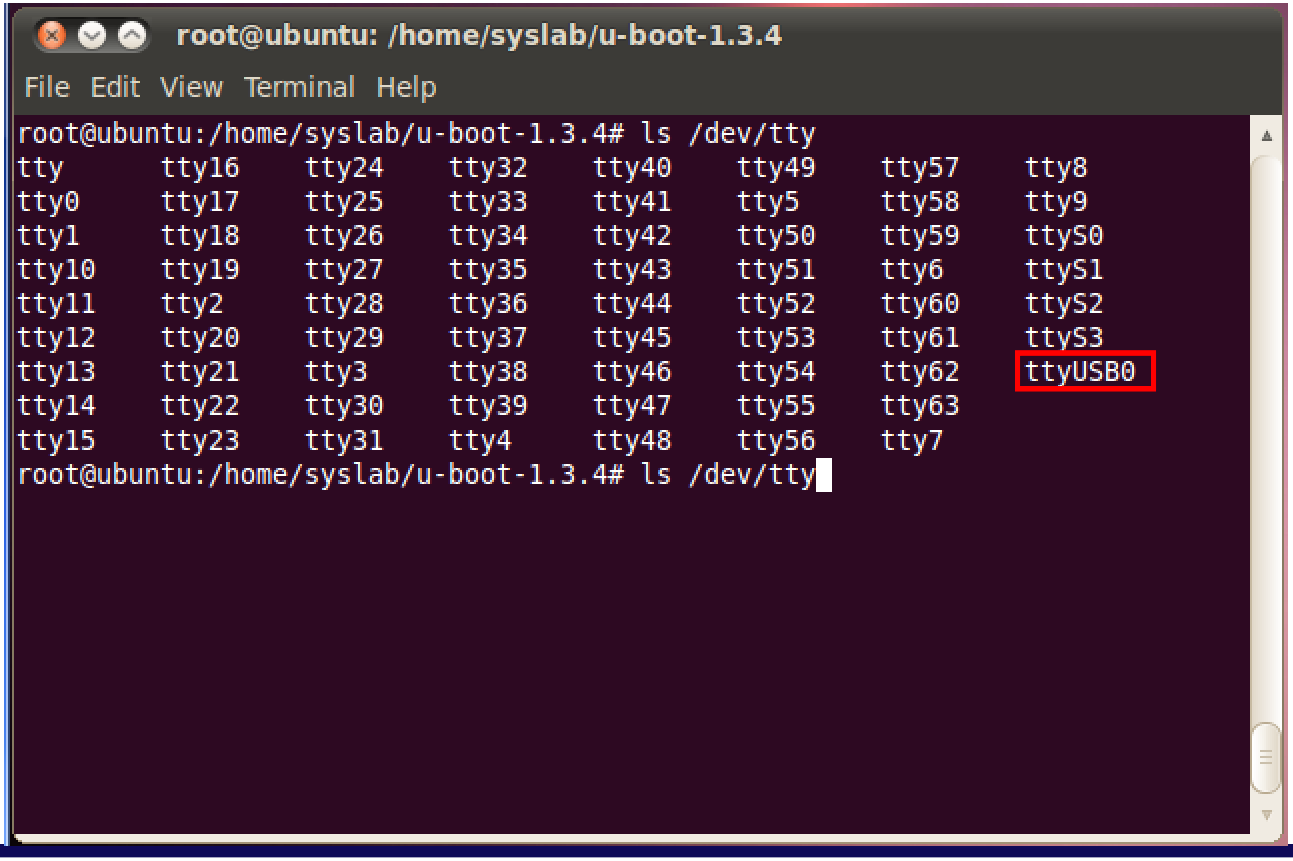
- **Player → Removable Devices**
  - Check ‘Prolific USB-Serial Controller → Connect



Serial Cable Connection

- **Verify that the board is connected**

  >> ls /dev/tty + Tap twice



root@ubuntu: /home/syslab/u-boot-1.3.4
File Edit View Terminal Help

```
root@ubuntu:/home/syslab/u-boot-1.3.4# ls /dev/tty
tty      tty16    tty24    tty32    tty40    tty49    tty57    tty8
tty0     tty17    tty25    tty33    tty41    tty5     tty58    tty9
tty1     tty18    tty26    tty34    tty42    tty50    tty59    ttyS0
tty10    tty19    tty27    tty35    tty43    tty51    tty6     ttyS1
tty11    tty2     tty28    tty36    tty44    tty52    tty60    ttyS2
tty12    tty20    tty29    tty37    tty45    tty53    tty61    ttyS3
tty13    tty21    tty3     tty38    tty46    tty54    tty62    ttyUSB0
tty14    tty22    tty30    tty39    tty47    tty55    tty63
tty15    tty23    tty31    tty4     tty48    tty56    tty7
root@ubuntu:/home/syslab/u-boot-1.3.4# ls /dev/tty
```

Install USB to Serial Driver

- **Install USB to Serial Driver (except linux)**

- **Windows**
  - Install using PL2303_Prolific_DriverInstaller_v1.10.0
- **MAC OS**
  - Search PL-2303 Mac driver → Download and install

- **Reboot after installation**

Minicom

- **Install minicom**

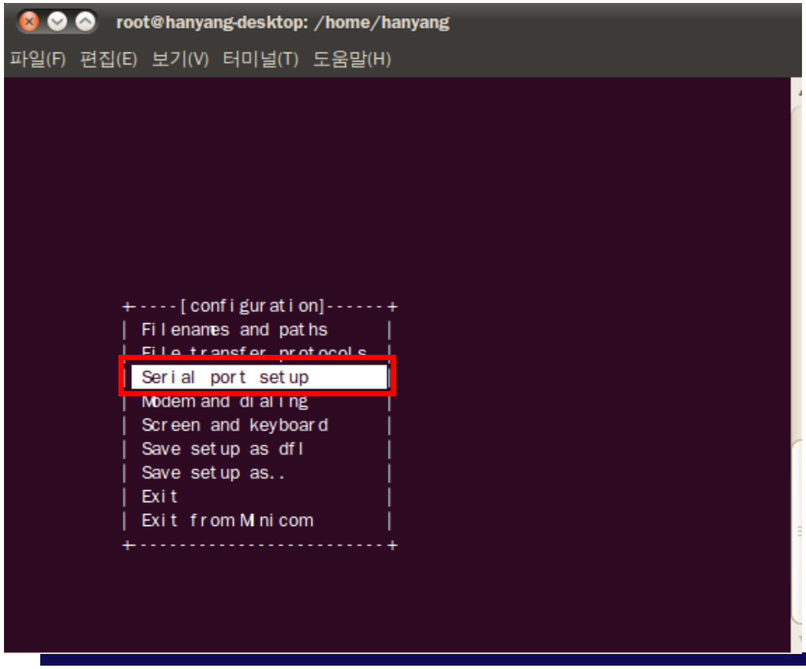
```
>> apt-get install minicom
```

Minicom

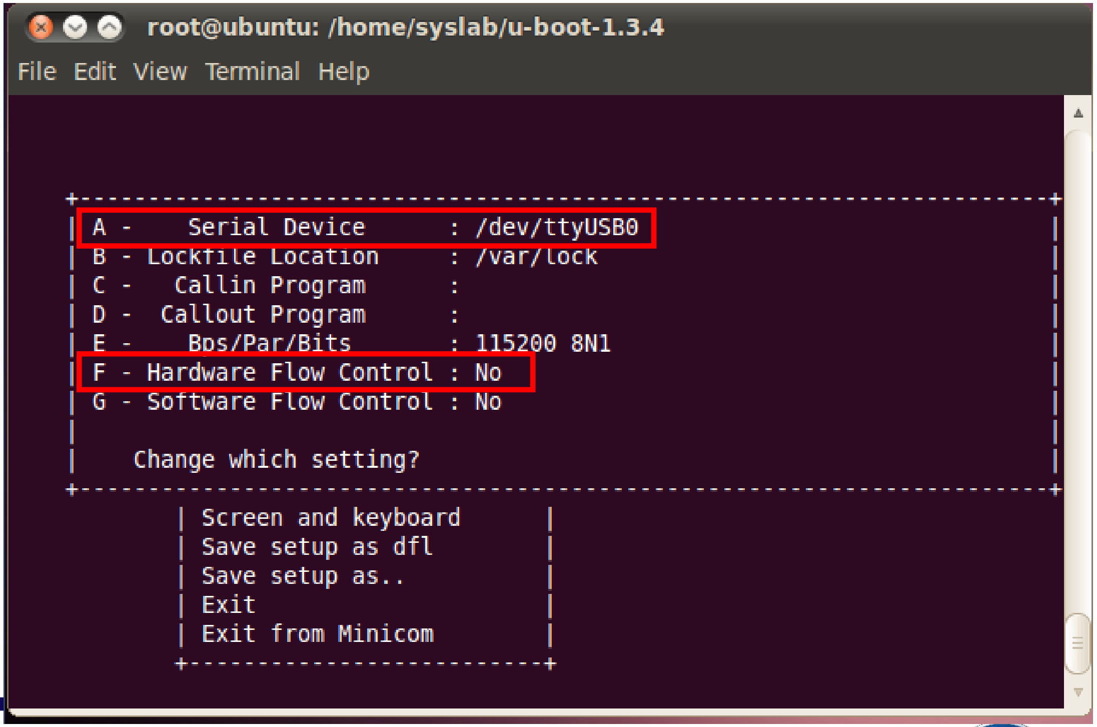
- **Minicom settings (be sure to use 'root' account)**

  >> minicom -s

  – In Serial port setup, change the 'Serial Device' and 'Hardware Flow Control' as follows (go to the corresponding item by pressing the keyboard 'a' and 'f' keys)


root@hanyang-desktop: /home/hanyang
파일(F) 편집(E) 보기(V) 터미널(T) 도움말(H)

```
+----[configuration]------+
| Filenames and paths     |
| File transfer protocols |
| Serial port setup       |
| Modem and dialing       |
| Screen and keyboard     |
| Save setup as dfl       |
| Save setup as..         |
| Exit                    |
| Exit from Minicom       |
+-------------------------+
```

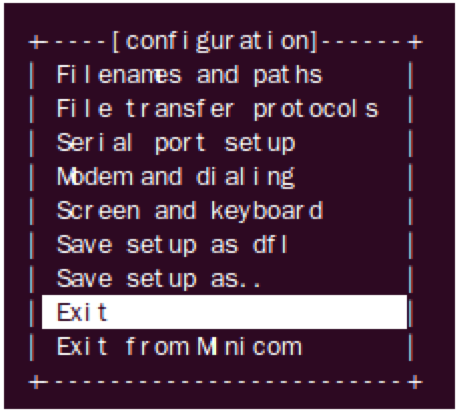

root@ubuntu: /home/syslab/u-boot-1.3.4
File Edit View Terminal Help

```
+-----------------------------------------------------------------------+
| A -    Serial Device      : /dev/ttyUSB0                              |
| B - Lockfile Location     : /var/lock                                 |
| C -   Callin Program      :                                           |
| D -  Callout Program      :                                           |
| E -    Bps/Par/Bits       : 115200 8N1                                |
| F - Hardware Flow Control : No                                        |
| G - Software Flow Control : No                                        |
|                                                                       |
|    Change which setting?                                              |
+-----------------------------------------------------------------------+
        | Screen and keyboard      |
        | Save setup as dfl        |
        | Save setup as..          |
        | Exit                     |
        | Exit from Minicom        |
        +--------------------------+
```

Minicom

- **Minicom settings**
  - Save minicom settings with ‘Save setup as dfl’

- **Connect to board with minicom**
  (1) >> minicom
  (2) >> minicom –s and ‘Exit’

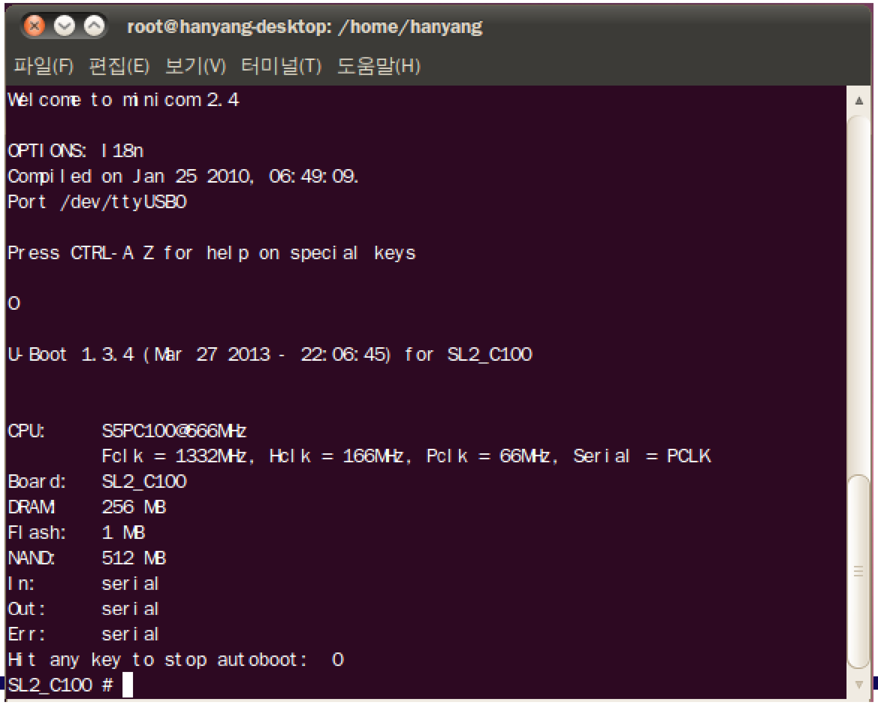


```
+----[configuration]------+
| Filenames and paths      |
| File transfer protocols  |
| Serial port setup        |
| Modem and dialing        |
| Screen and keyboard      |
| Save setup as dfl        |
| Save setup as..          |
| Exit                     |
| Exit from Minicom        |
+--------------------------+
```

U-boot

- **Power on the board and press the ‘Enter’ key**

- **If the following screen appears, the board connection succeeds.**
  - If the screen is not displayed, press the reset button to reset



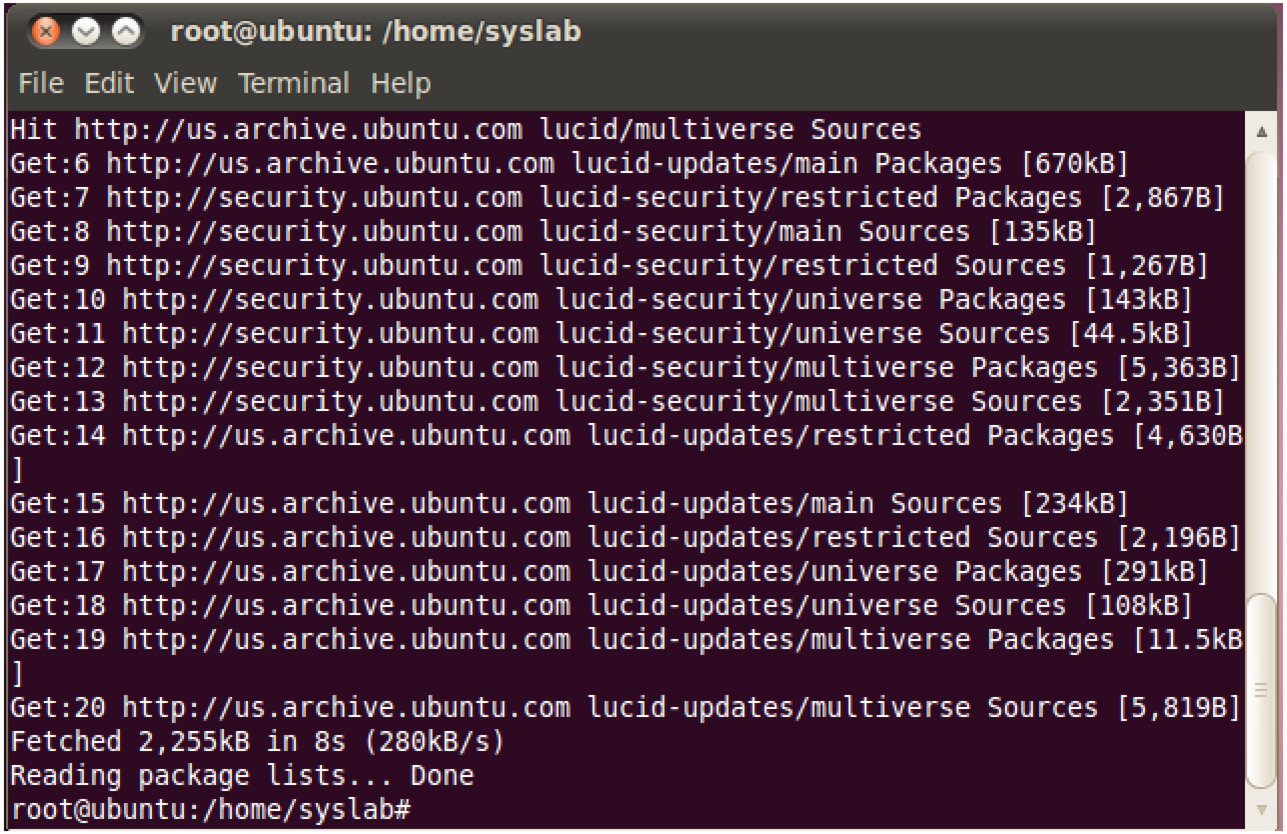
root@hanyang-desktop: /home/hanyang
파일(F) 편집(E) 보기(V) 터미널(T) 도움말(H)

```
Welcome to minicom 2.4

OPTIONS: I18n
Compiled on Jan 25 2010, 06:49:09.
Port /dev/ttyUSB0

Press CTRL-A Z for help on special keys

O

U-Boot 1.3.4 (Mar 27 2013 - 22:06:45) for SL2_C100


CPU:     S5PC100@666MHz
         Fclk = 1332MHz, Hclk = 166MHz, Pclk = 66MHz, Serial = PCLK
Board:   SL2_C100
DRAM     256 MB
Flash:   1 MB
NAND:    512 MB
In:      serial
Out:     serial
Err:     serial
Hit any key to stop autoboot:  0
SL2_C100 #
```

INSTALL TFTP, ARM CROSS COMPILER

Get Update Files

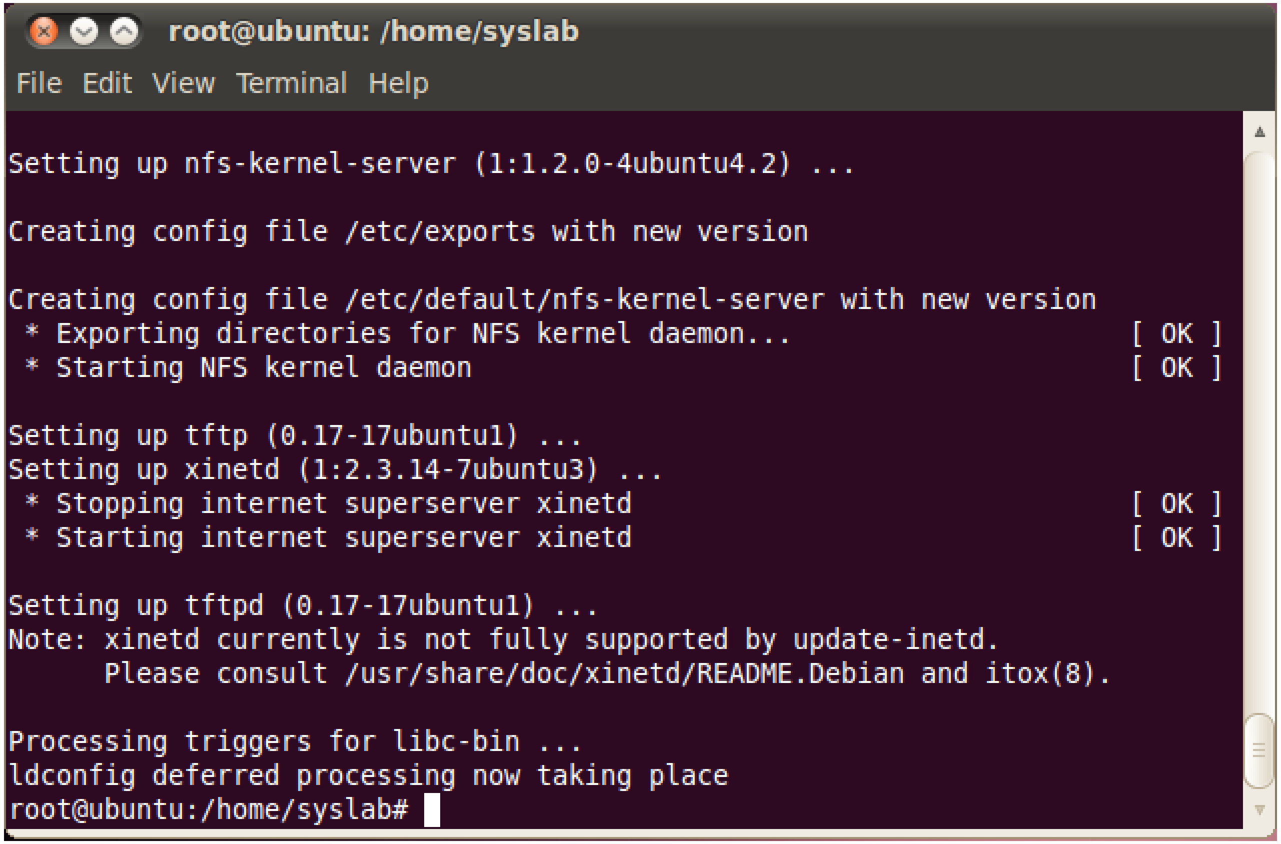
>> apt-get update



```
Hit http://us.archive.ubuntu.com lucid/multiverse Sources
Get:6 http://us.archive.ubuntu.com lucid-updates/main Packages [670kB]
Get:7 http://security.ubuntu.com lucid-security/restricted Packages [2,867B]
Get:8 http://security.ubuntu.com lucid-security/main Sources [135kB]
Get:9 http://security.ubuntu.com lucid-security/restricted Sources [1,267B]
Get:10 http://security.ubuntu.com lucid-security/universe Packages [143kB]
Get:11 http://security.ubuntu.com lucid-security/universe Sources [44.5kB]
Get:12 http://security.ubuntu.com lucid-security/multiverse Packages [5,363B]
Get:13 http://security.ubuntu.com lucid-security/multiverse Sources [2,351B]
Get:14 http://us.archive.ubuntu.com lucid-updates/restricted Packages [4,630B
]
Get:15 http://us.archive.ubuntu.com lucid-updates/main Sources [234kB]
Get:16 http://us.archive.ubuntu.com lucid-updates/restricted Sources [2,196B]
Get:17 http://us.archive.ubuntu.com lucid-updates/universe Packages [291kB]
Get:18 http://us.archive.ubuntu.com lucid-updates/universe Sources [108kB]
Get:19 http://us.archive.ubuntu.com lucid-updates/multiverse Packages [11.5kB
]
Get:20 http://us.archive.ubuntu.com lucid-updates/multiverse Sources [5,819B]
Fetched 2,255kB in 8s (280kB/s)
Reading package lists... Done
root@ubuntu:/home/syslab#
```

Install and Set up a Linux-based Development Environment

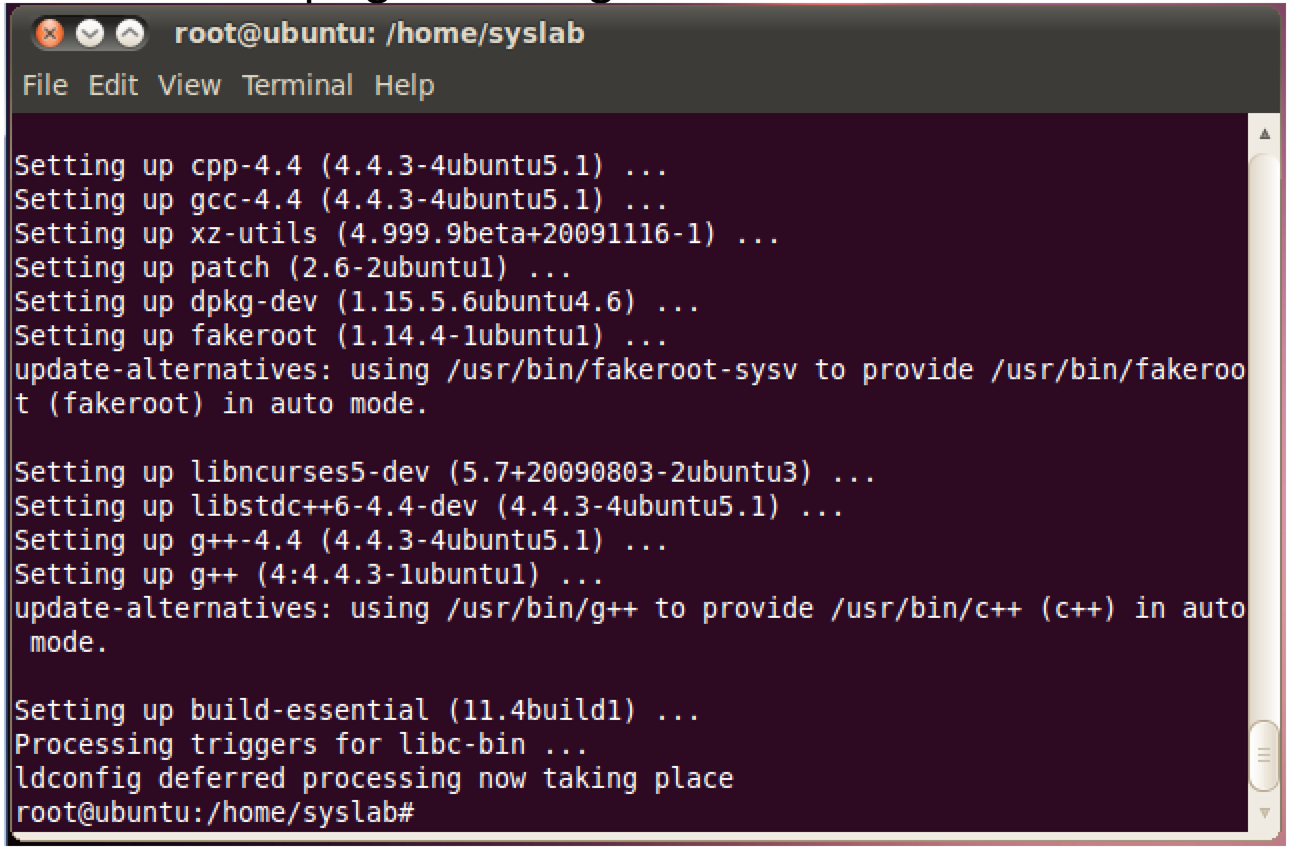
- **Install package**
  - >> apt-get install nfs-kernel-server tftp tftpd xinetd



```
Setting up nfs-kernel-server (1:1.2.0-4ubuntu4.2) ...

Creating config file /etc/exports with new version

Creating config file /etc/default/nfs-kernel-server with new version
 * Exporting directories for NFS kernel daemon...                      [ OK ]
 * Starting NFS kernel daemon                                          [ OK ]

Setting up tftp (0.17-17ubuntu1) ...
Setting up xinetd (1:2.3.14-7ubuntu3) ...
 * Stopping internet superserver xinetd                                [ OK ]
 * Starting internet superserver xinetd                                [ OK ]

Setting up tftpd (0.17-17ubuntu1) ...
Note: xinetd currently is not fully supported by update-inetd.
      Please consult /usr/share/doc/xinetd/README.Debian and itox(8).

Processing triggers for libc-bin ...
ldconfig deferred processing now taking place
root@ubuntu:/home/syslab# 
```

Install and Set up a Linux-based Development Environment

- >> apt-get install libncurses5 libncurses5-dev build-essential
- For 64bit >> apt-get install gcc-multilib



```
Setting up cpp-4.4 (4.4.3-4ubuntu5.1) ...
Setting up gcc-4.4 (4.4.3-4ubuntu5.1) ...
Setting up xz-utils (4.999.9beta+20091116-1) ...
Setting up patch (2.6-2ubuntu1) ...
Setting up dpkg-dev (1.15.5.6ubuntu4.6) ...
Setting up fakeroot (1.14.4-1ubuntu1) ...
update-alternatives: using /usr/bin/fakeroot-sysv to provide /usr/bin/fakeroo
t (fakeroot) in auto mode.

Setting up libncurses5-dev (5.7+20090803-2ubuntu3) ...
Setting up libstdc++6-4.4-dev (4.4.3-4ubuntu5.1) ...
Setting up g++-4.4 (4.4.3-4ubuntu5.1) ...
Setting up g++ (4:4.4.3-1ubuntu1) ...
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto
 mode.

Setting up build-essential (11.4build1) ...
Processing triggers for libc-bin ...
ldconfig deferred processing now taking place
root@ubuntu:/home/syslab#
```

TFTP

- **What is TFTP?**
  - Trivial File Transfer Protocol
  - Protocols for transferring files like FTP
    - Transfer in a simpler way than FTP
  - Mainly used as operating system upload on embedded systems

+ Easy to implement

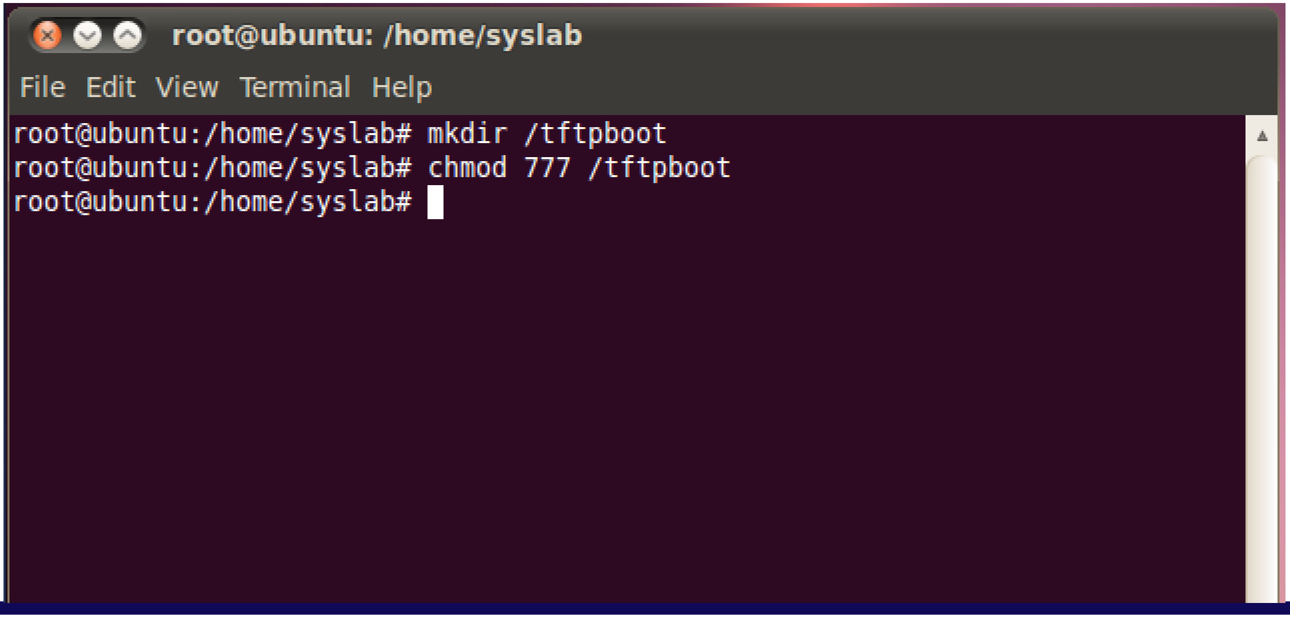
- Data may be lost during data transfer

TFTP Setting

- **TFTP Setting**
  - Make tftpboot folder

    >> mkdir /tftpboot
  - Setting tftpboot folder permissions

    >> chmod 777 /tftpboot



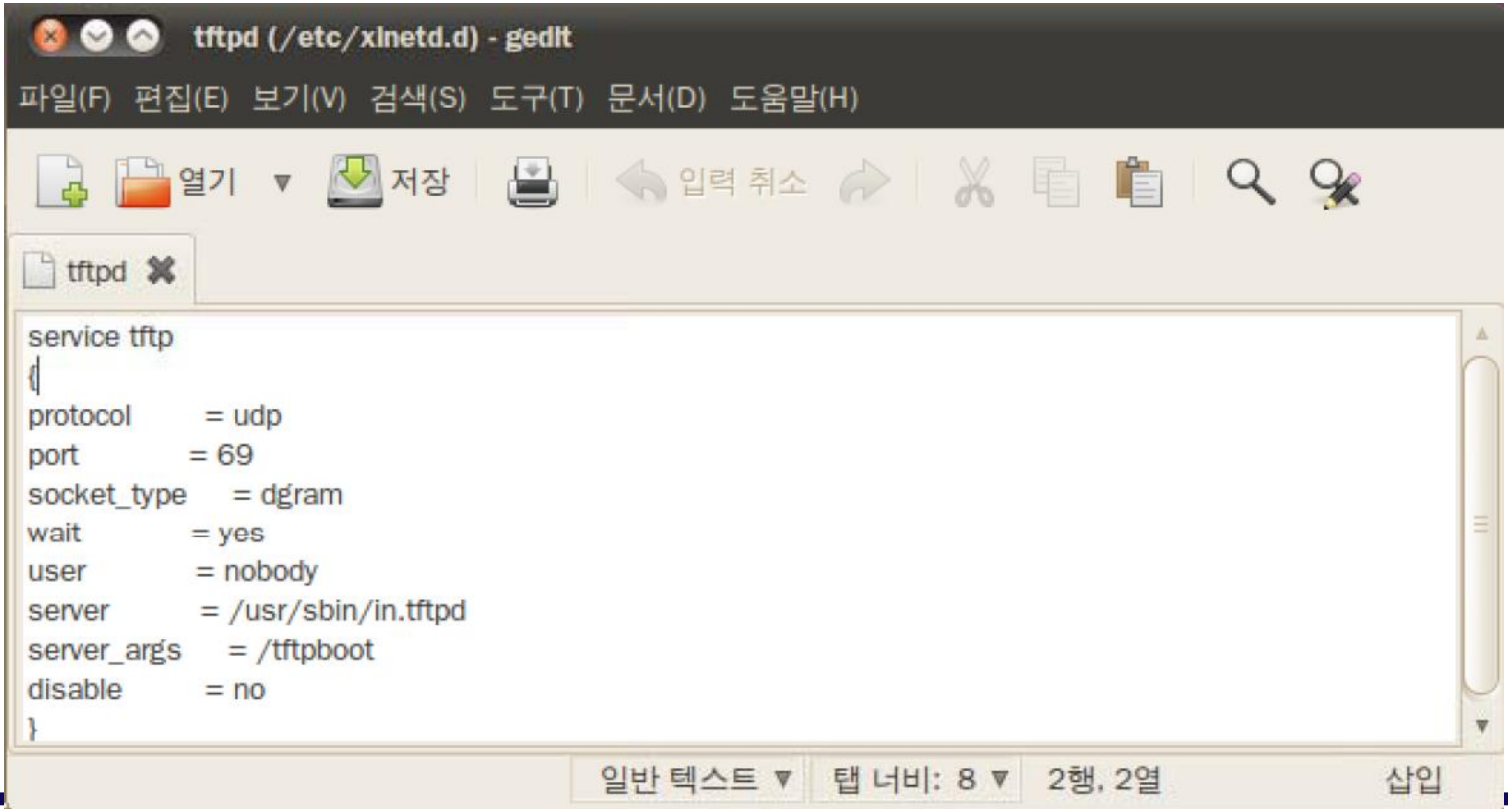
```
root@ubuntu:/home/syslab# mkdir /tftpboot
root@ubuntu:/home/syslab# chmod 777 /tftpboot
root@ubuntu:/home/syslab# 
```

TFTP Setting

- **TFTP Preferences**

  `>> gedit /etc/xinetd.d/tftpd`

  – Edit and save as shown below



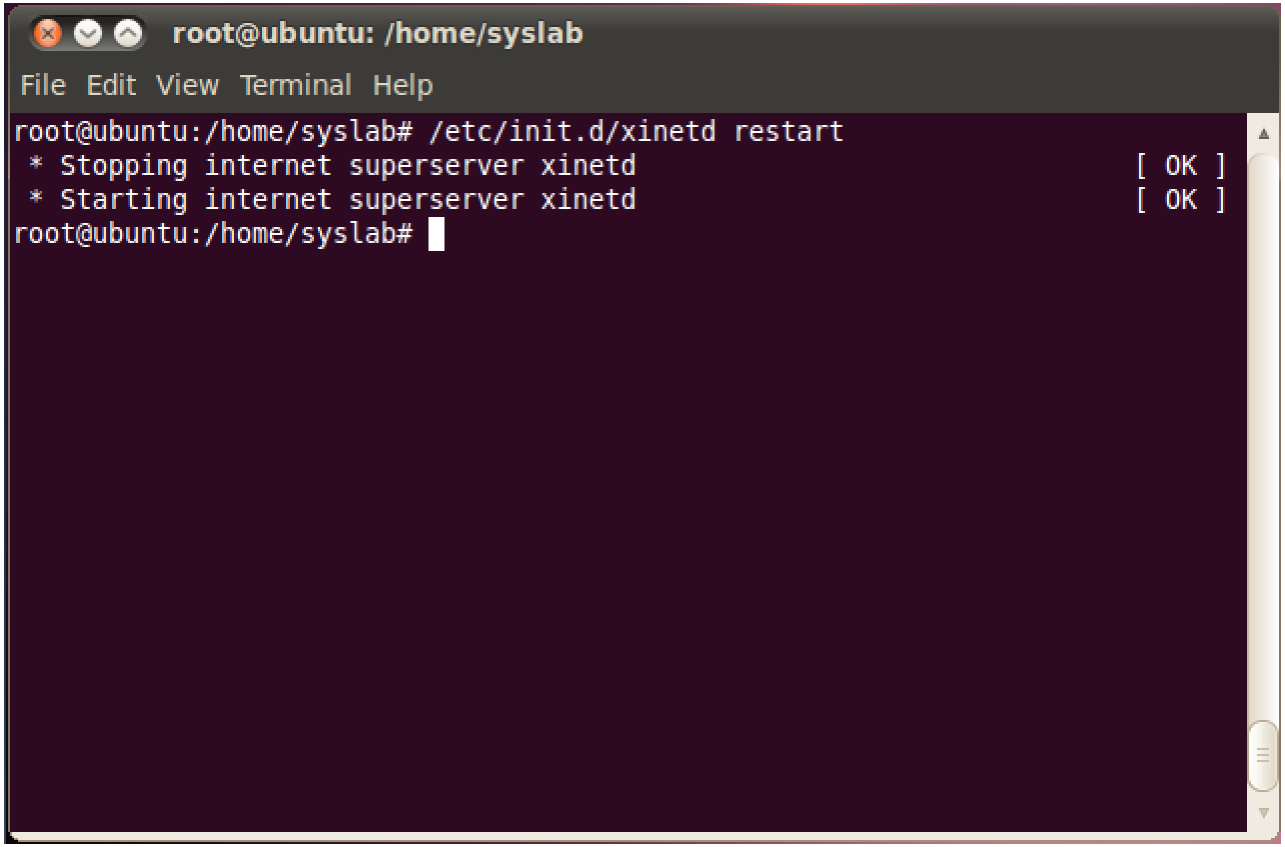
tftpd (/etc/xinetd.d) - gedit
파일(F) 편집(E) 보기(V) 검색(S) 도구(T) 문서(D) 도움말(H)
열기
저장
입력 취소
tftpd
일반 텍스트
탭 너비: 8
2행, 2열
삽입

```
service tftp
{
protocol        = udp
port            = 69
socket_type     = dgram
wait            = yes
user            = nobody
server          = /usr/sbin/in.tftpd
server_args     = /tftpboot
disable         = no
}
```

TFTP Setting

- **Xinetd restart**

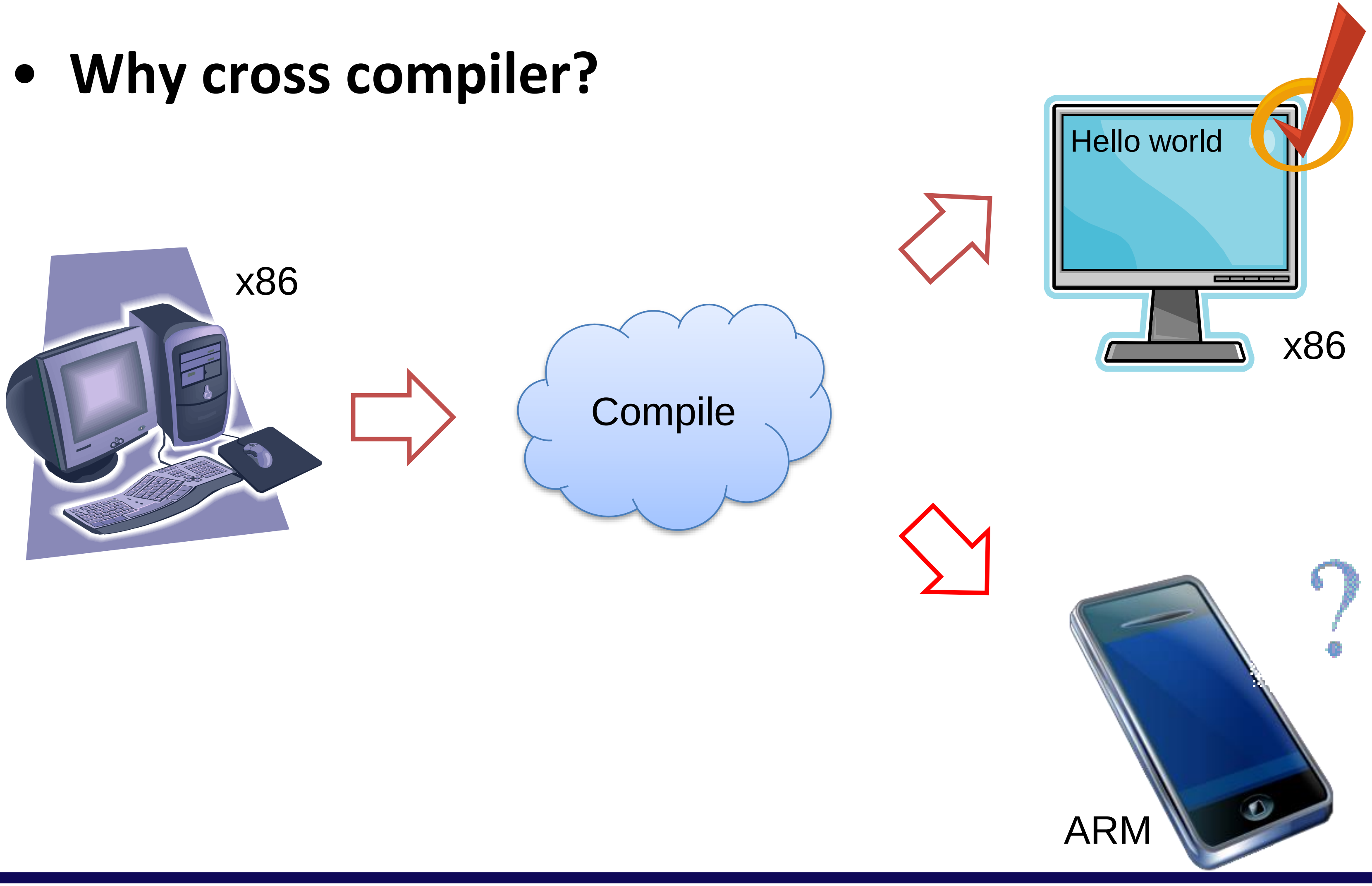
  >> /etc/init.d/xinetd restart



```
root@ubuntu:/home/syslab# /etc/init.d/xinetd restart
 * Stopping internet superserver xinetd                              [ OK ]
 * Starting internet superserver xinetd                              [ OK ]
root@ubuntu:/home/syslab# 
```

Cross Compiler

- **Why cross compiler?**



x86

Compile

Hello world

x86

ARM

Cross Compiler

- **Need for cross compiler**
  - On normal computer systems (x86), compiling and execution are done on the same machine (x86)
  - On embedded systems (ARM), compilation is performed on the host computer (x86) and executed in ARM systems.
    - Common compiler translates original codes to x86 machine language and can not run on ARM
- **What is cross compiler?**
  - Rather than translating the program into the machine language of the computer on which the compiler is running, compiling into machine language for different models
- **Toolchain**
  - Compilation environment of host computer for software development of desired embedded system
  - Includes cross compiler

Install ToolChain

- **Install ToolChain**

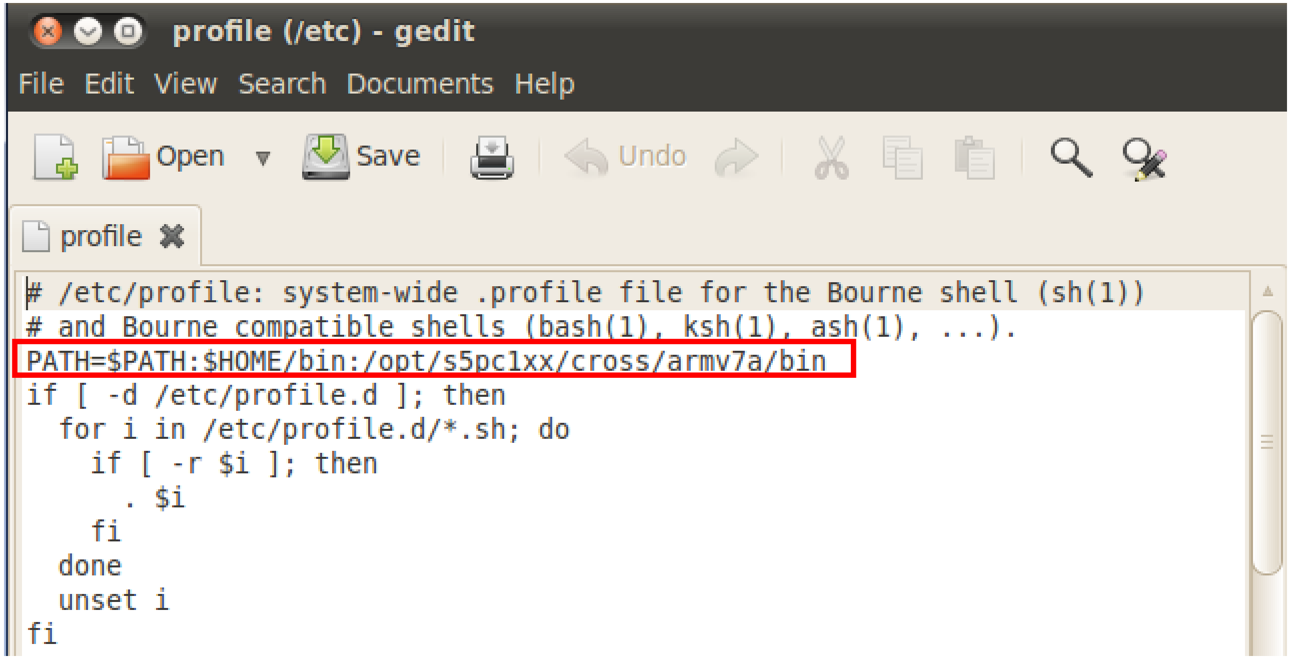
```
>> mv toolchain-s5pc1xx.tar.gz /opt
>> tar -xvf toolchain-s5pc1xx.tar.gz
```

Install ToolChain

- **Set cross compiler environment variables**

  >> gedit /etc/profile

  – Add PATH=$PATH:$HOME/bin:/opt/s5pc1xx/cross/armv7a/bin



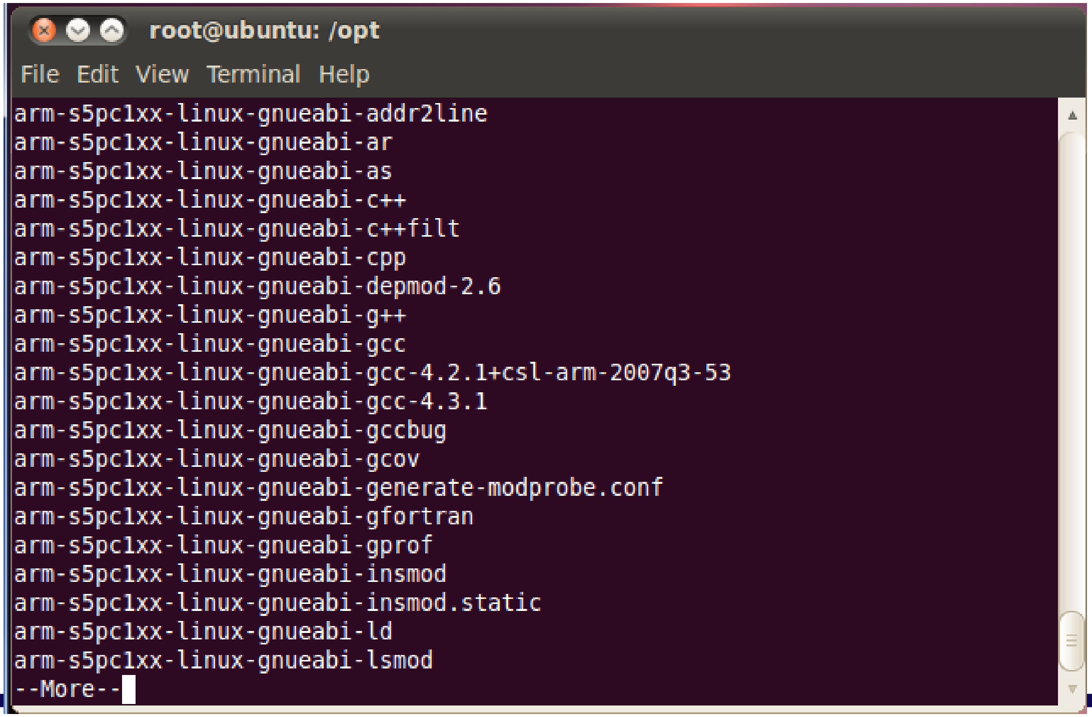
```
# /etc/profile: system-wide .profile file for the Bourne shell (sh(1))
# and Bourne compatible shells (bash(1), ksh(1), ash(1), ...).
PATH=$PATH:$HOME/bin:/opt/s5pc1xx/cross/armv7a/bin
if [ -d /etc/profile.d ]; then
  for i in /etc/profile.d/*.sh; do
    if [ -r $i ]; then
      . $i
    fi
  done
  unset i
fi
```

Install ToolChain

- **Apply the modification**

  >> source /etc/profile

- **Confirm cross compiler is applied**
  - Type ‘arm’ in the shell and press the Tap key twice



```
arm-s5pc1xx-linux-gnueabi-addr2line
arm-s5pc1xx-linux-gnueabi-ar
arm-s5pc1xx-linux-gnueabi-as
arm-s5pc1xx-linux-gnueabi-c++
arm-s5pc1xx-linux-gnueabi-c++filt
arm-s5pc1xx-linux-gnueabi-cpp
arm-s5pc1xx-linux-gnueabi-depmod-2.6
arm-s5pc1xx-linux-gnueabi-g++
arm-s5pc1xx-linux-gnueabi-gcc
arm-s5pc1xx-linux-gnueabi-gcc-4.2.1+csl-arm-2007q3-53
arm-s5pc1xx-linux-gnueabi-gcc-4.3.1
arm-s5pc1xx-linux-gnueabi-gccbug
arm-s5pc1xx-linux-gnueabi-gcov
arm-s5pc1xx-linux-gnueabi-generate-modprobe.conf
arm-s5pc1xx-linux-gnueabi-gfortran
arm-s5pc1xx-linux-gnueabi-gprof
arm-s5pc1xx-linux-gnueabi-insmod
arm-s5pc1xx-linux-gnueabi-insmod.static
arm-s5pc1xx-linux-gnueabi-ld
arm-s5pc1xx-linux-gnueabi-lsmod
--More--
```

U-BOOT PORTING

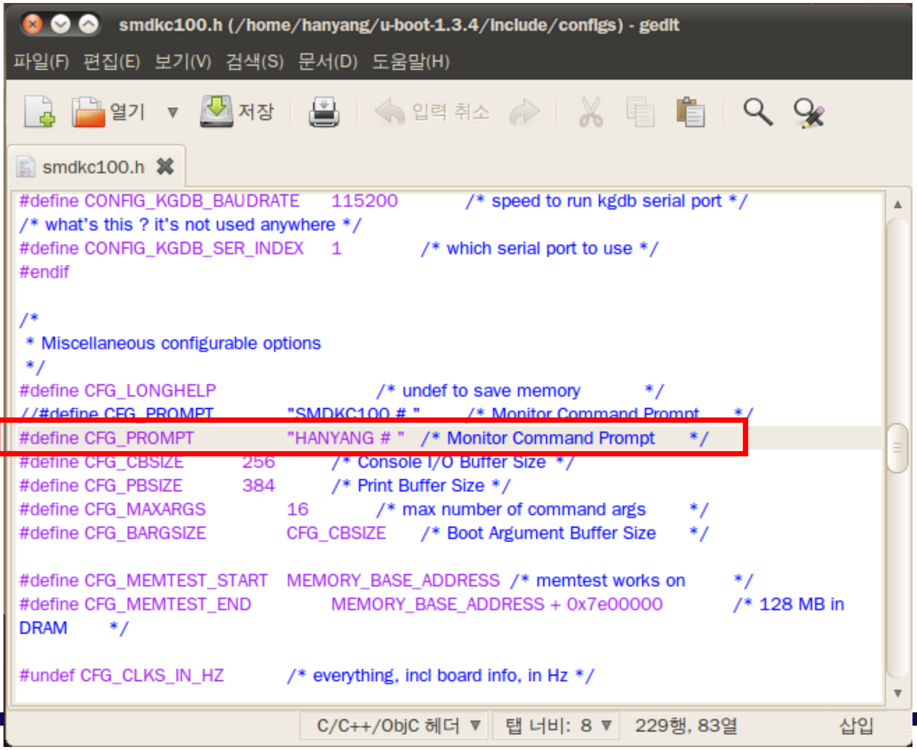
U-boot Compilation

- **Modify the code to make sure U-boot is installed correctly**

  >> gedit /home/hanyang/u-boot-1.3.4/include/configs/smdkc100.h

- **Modify the string of CFG_PROMPT on line 229**

  # define CFG_PROMPT “SL2_C100 #” ➔ “Student ID #”


smdkc100.h (/home/hanyang/u-boot-1.3.4/include/configs) - gedit
파일(F) 편집(E) 보기(V) 검색(S) 문서(D) 도움말(H)
열기
저장
입력 취소
smdkc100.h
C/C++/ObjC 헤더
탭 너비: 8
229행, 83열
삽입

```
#define CONFIG_KGDB_BAUDRATE     115200          /* speed to run kgdb serial port */
/* what's this ? it's not used anywhere */
#define CONFIG_KGDB_SER_INDEX    1           /* which serial port to use */
#endif

/*
 * Miscellaneous configurable options
 */
#define CFG_LONGHELP                        /* undef to save memory         */
//#define CFG_PROMPT            "SMDKC100 # "      /* Monitor Command Prompt      */
#define CFG_PROMPT              "HANYANG # "   /* Monitor Command Prompt      */
#define CFG_CBSIZE        256         /* Console I/O Buffer Size  */
#define CFG_PBSIZE        384         /* Print Buffer Size */
#define CFG_MAXARGS             16          /* max number of command args       */
#define CFG_BARGSIZE            CFG_CBSIZE      /* Boot Argument Buffer Size      */

#define CFG_MEMTEST_START   MEMORY_BASE_ADDRESS  /* memtest works on        */
#define CFG_MEMTEST_END         MEMORY_BASE_ADDRESS + 0x7e00000         /* 128 MB in
DRAM      */

#undef CFG_CLKS_IN_HZ        /* everything, incl board info, in Hz */
```

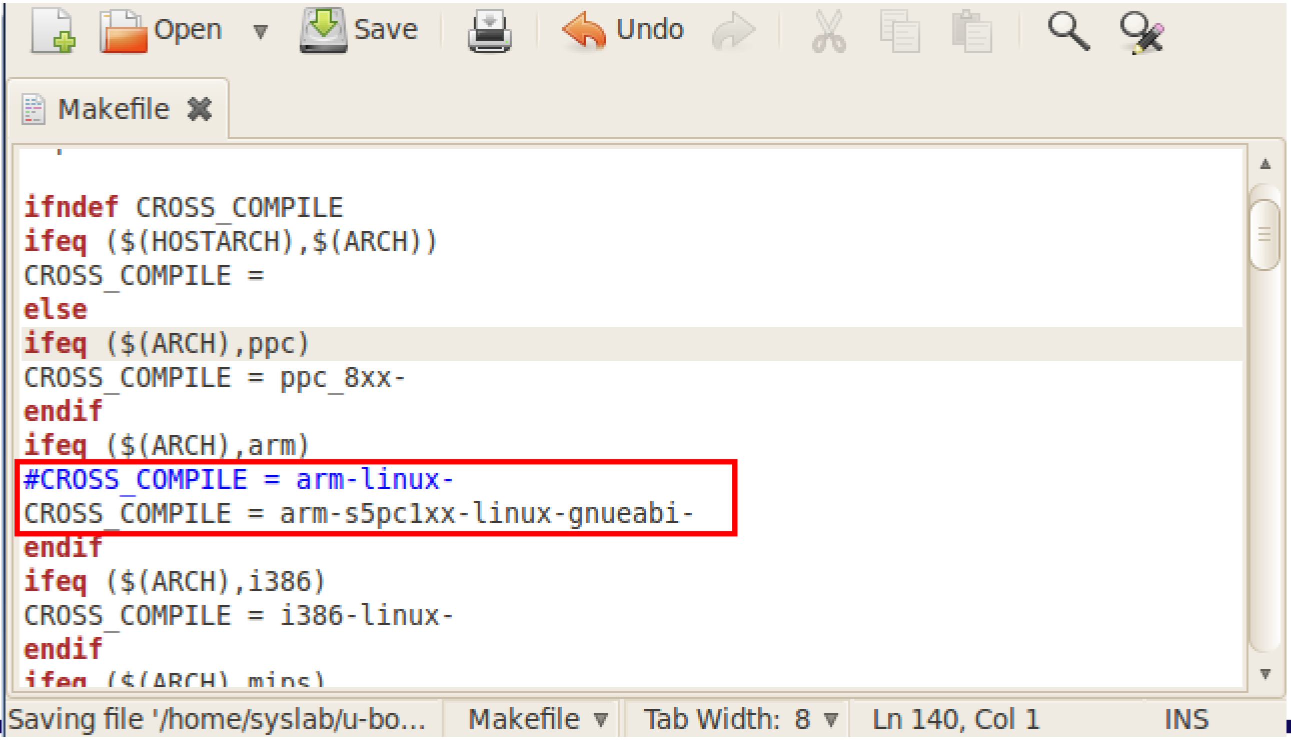
U-boot Compilation

- **Confirm Makefile**

  >> gedit /home/hanyang/u-boot-1.3.4/Makefile

  – Comment #CROSS_COMPILE = arm-linux-   on line 144

    [using ‘arm-s5pc1xx-linux-gnueabi-’ cross compiler]



```
ifndef CROSS_COMPILE
ifeq ($(HOSTARCH),$(ARCH))
CROSS_COMPILE =
else
ifeq ($(ARCH),ppc)
CROSS_COMPILE = ppc_8xx-
endif
ifeq ($(ARCH),arm)
#CROSS_COMPILE = arm-linux-
CROSS_COMPILE = arm-s5pc1xx-linux-gnueabi-
endif
ifeq ($(ARCH),i386)
CROSS_COMPILE = i386-linux-
endif
ifeq ($(ARCH),mips)
```

U-boot Compilation

- **U-boot make**

  >> cd /home/hanyang/u-boot-1.3.4

  >> source /etc/profile

  >> make clean

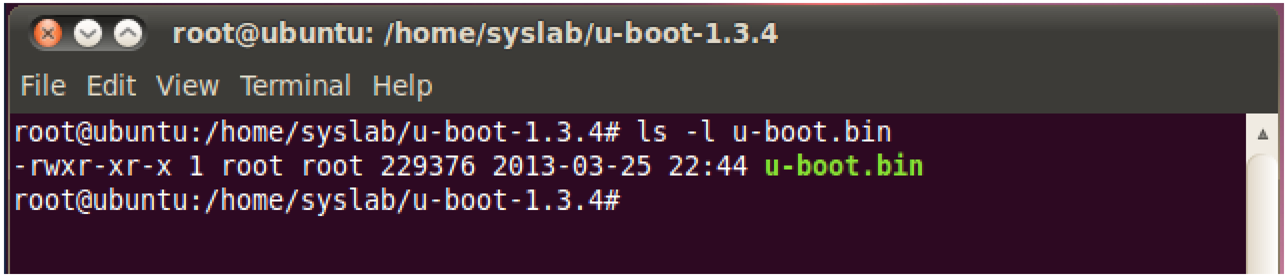
  >> make clobber

  >> make smdkc100_config

  >> make

- **Ensure the the U-boot.bin file is created**

  >> ls –l u-boot.bin


root@ubuntu: /home/syslab/u-boot-1.3.4
File Edit View Terminal Help

```
root@ubuntu:/home/syslab/u-boot-1.3.4# ls -l u-boot.bin
-rwxr-xr-x 1 root root 229376 2013-03-25 22:44 u-boot.bin
root@ubuntu:/home/syslab/u-boot-1.3.4#
```

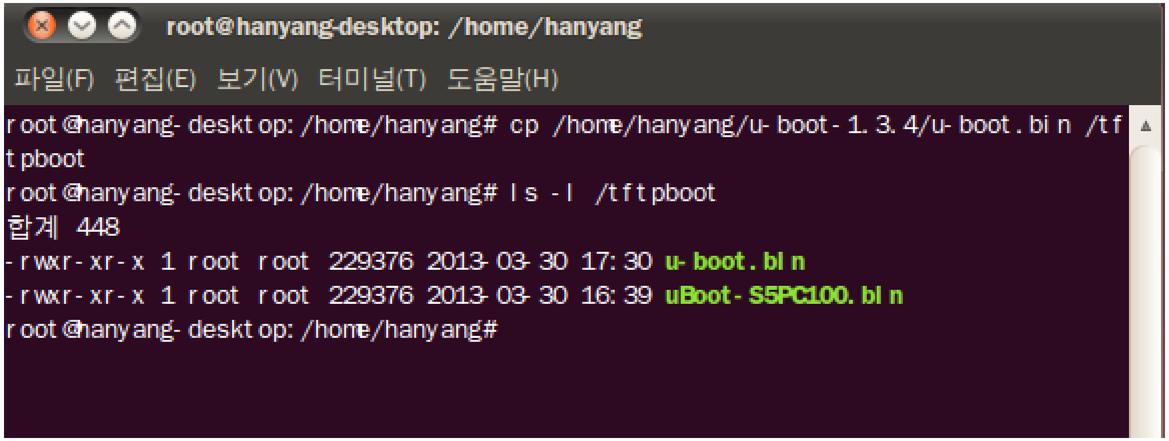
U-boot Compilation

- **Copy u-boot.bin file to the /tftpboot folder**

  >> cp /home/hanyang/u-boot-1.3.4/u-boot.bin /tftpboot

- **Verify that the u-boot.bin file has been copied**

  >> ls –l /tftpboot

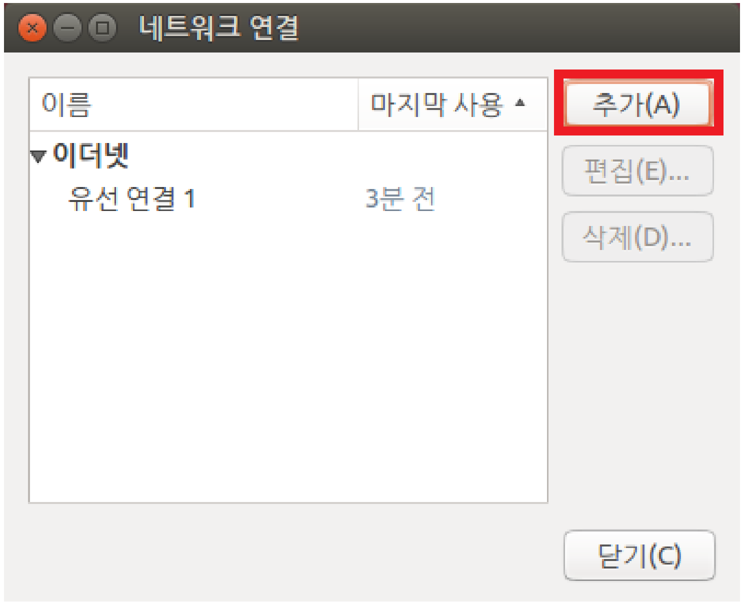

root@hanyang-desktop: /home/hanyang
파일(F) 편집(E) 보기(V) 터미널(T) 도움말(H)

```
root@hanyang-desktop:/home/hanyang# cp /home/hanyang/u-boot-1.3.4/u-boot.bin /tf
tpboot
root@hanyang-desktop:/home/hanyang# ls -l /tftpboot
합계 448
-rwxr-xr-x 1 root root 229376 2013-03-30 17:30 u-boot.bin
-rwxr-xr-x 1 root root 229376 2013-03-30 16:39 uBoot-S5PC100.bin
root@hanyang-desktop:/home/hanyang#
```

Network Settings

- **Network settings**
  - Linux IP settings

- **Network Connections**
  - Add -> Ethernet

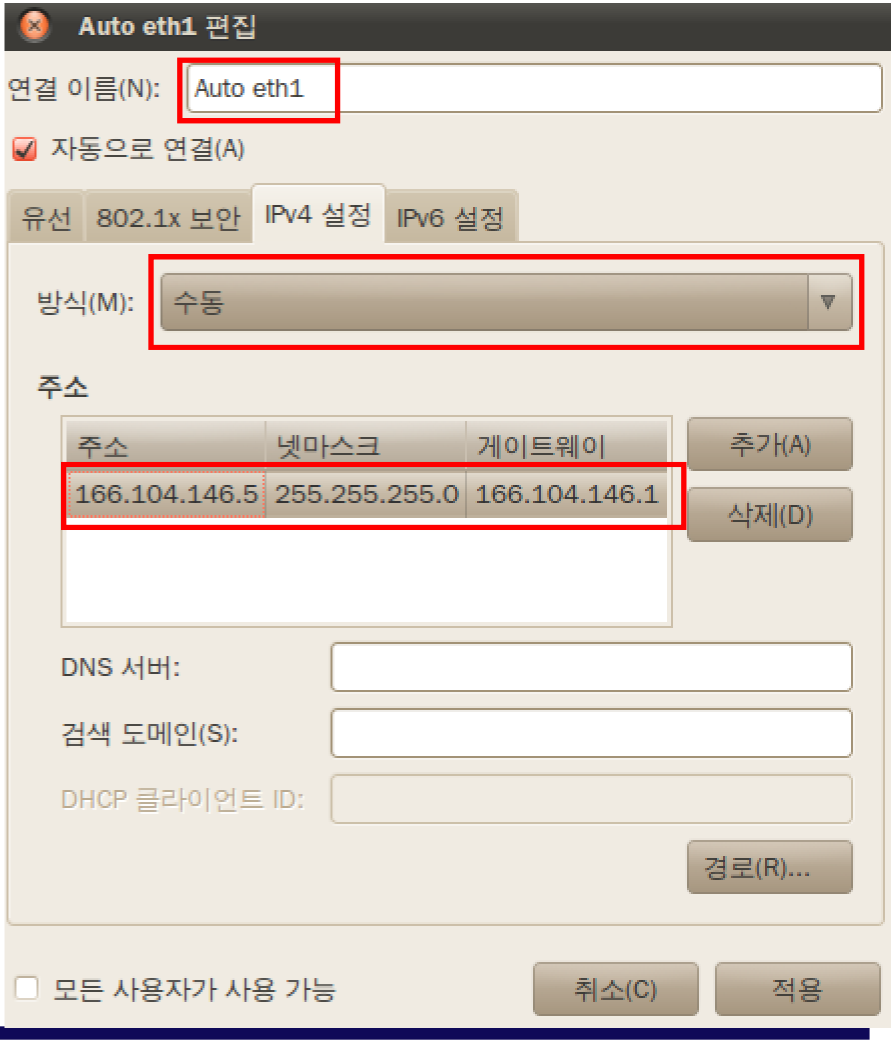


네트워크 연결

이름	마지막 사용 ▲
▾ **이더넷**	
유선 연결 1	3분 전

Network Settings

- **Edit Auto eth1**
  - IPv4 Settings
    - Method : Manual
    - Address : 166.104.146.5
    - Netmask : 255.255.255.0
    - Gateway : 166.104.146.1
  - Click ‘Save…’



Auto eth1 편집

연결 이름(N): Auto eth1

☑ 자동으로 연결(A)

방식(M): 수동

주소

주소	넷마스크	게이트웨이
166.104.146.5	255.255.255.0	166.104.146.1

DNS 서버:

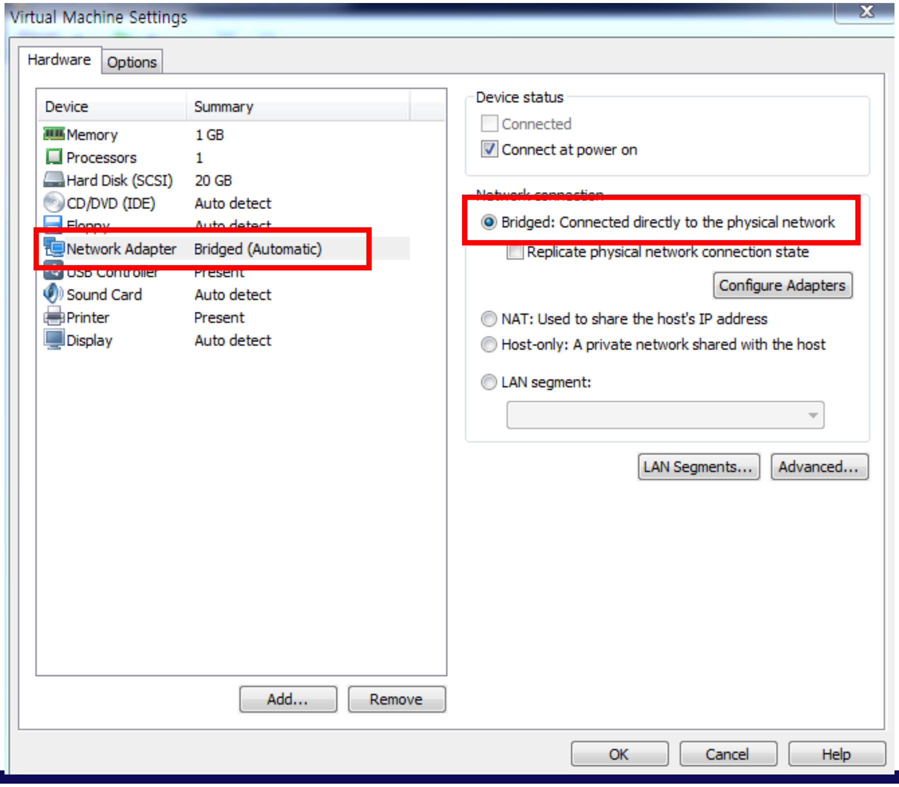
검색 도메인(S):

DHCP 클라이언트 ID:

□ 모든 사용자가 사용 가능

Network Settings

- **Network settings of virtual machine**
  - Shutdown linux
  - Hardware tab→ Network Adapter → Bridged



Virtual Machine Settings

Hardware | Options

Device	Summary
Memory	1 GB
Processors	1
Hard Disk (SCSI)	20 GB
CD/DVD (IDE)	Auto detect
Floppy	Auto detect
Network Adapter	Bridged (Automatic)
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Device status

□ Connected

☑ Connect at power on

Network connection

● Bridged: Connected directly to the physical network

□ Replicate physical network connection state

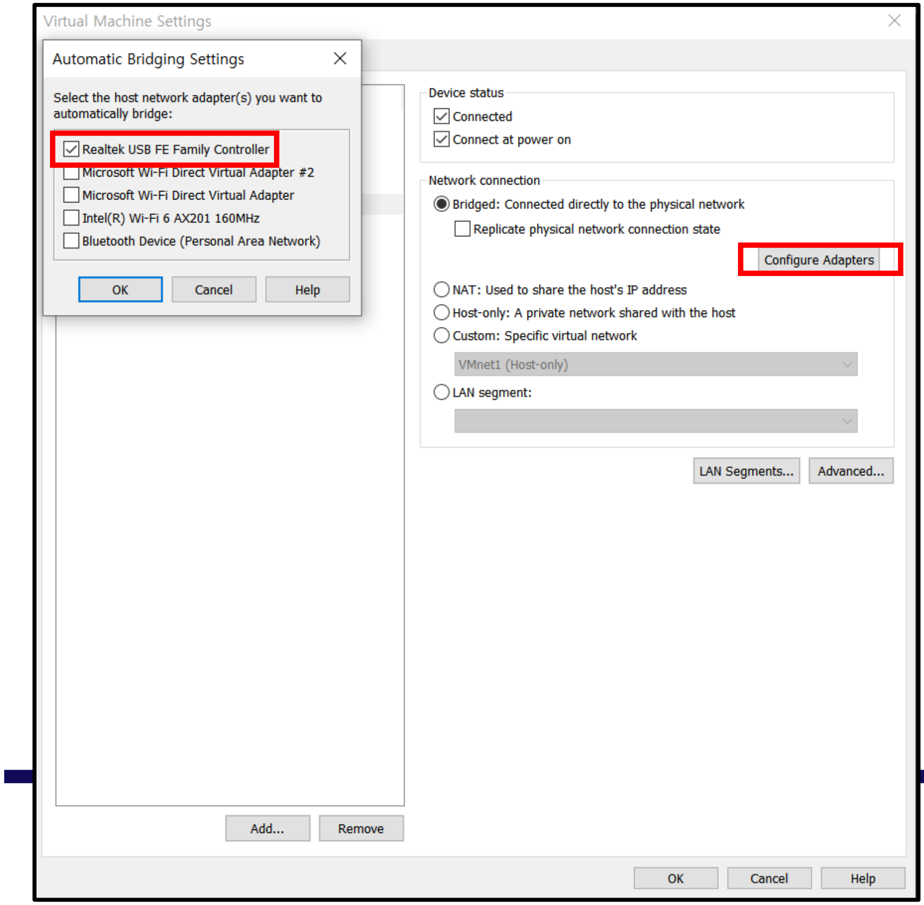
○ NAT: Used to share the host's IP address

○ Host-only: A private network shared with the host

○ LAN segment:

Network Settings

- **Network settings of virtual machine**
  - Configure Adapters
  - Check only ethernet adapters (not WIFI, Bluetooth)



Virtual Machine Settings

Automatic Bridging Settings

Select the host network adapter(s) you want to automatically bridge:

☑ Realtek USB FE Family Controller
□ Microsoft Wi-Fi Direct Virtual Adapter #2
□ Microsoft Wi-Fi Direct Virtual Adapter
□ Intel(R) Wi-Fi 6 AX201 160MHz
□ Bluetooth Device (Personal Area Network)

Device status

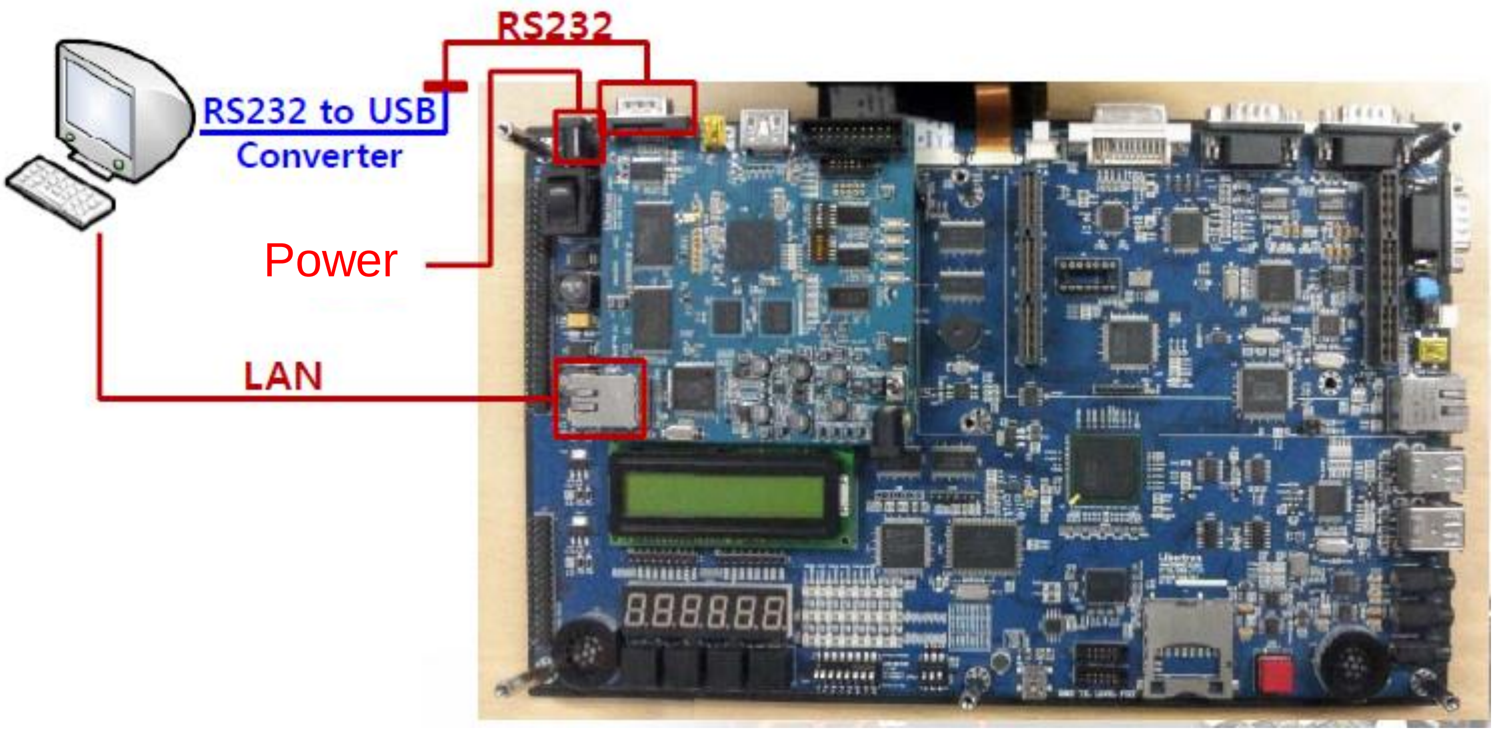
☑ Connected
☑ Connect at power on

Network connection

● Bridged: Connected directly to the physical network
□ Replicate physical network connection state
○ NAT: Used to share the host's IP address
○ Host-only: A private network shared with the host
○ Custom: Specific virtual network
VMnet1 (Host-only)
○ LAN segment:

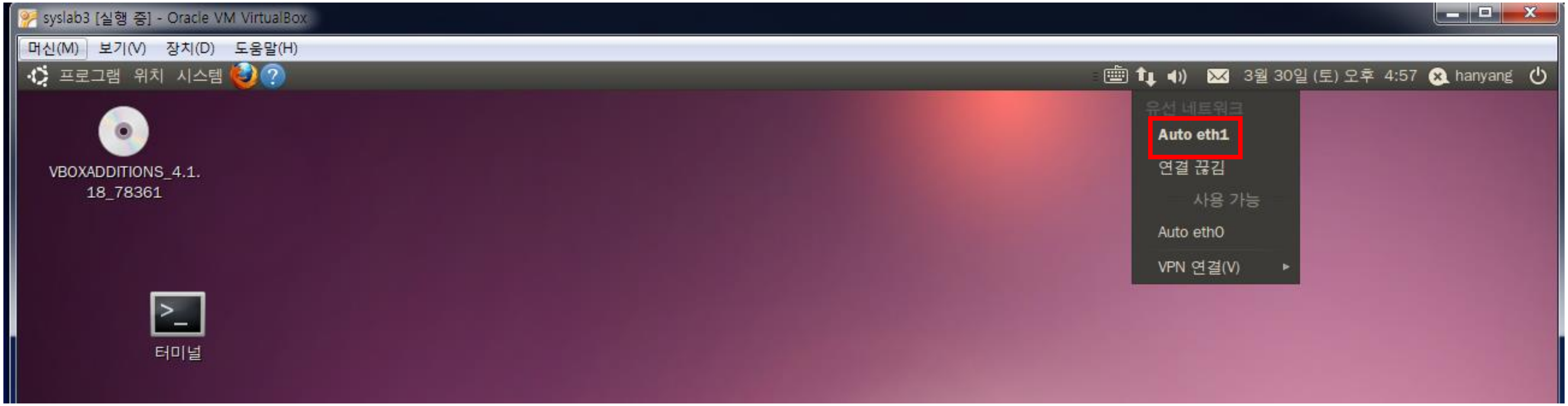
Network Connection

- **Network connection**
  - Connect LAN to pc



Network Connection

- Connect the network to 'Auto eth1' after reboot Linux



U-boot Porting

- **Network preferences**
  - Use the following U-boot command to set the IP of the host PC and the IP of the target board

    # setenv ipaddr 166.104.146.10 : save target board IP address

    # setenv serverip 166.104.146.5 : linux IP address

    # setenv gatewayip 166.104.146.1 : gateway address

    # setenv netmask 255.255.255.0 : netmask address

- **Save preferences**

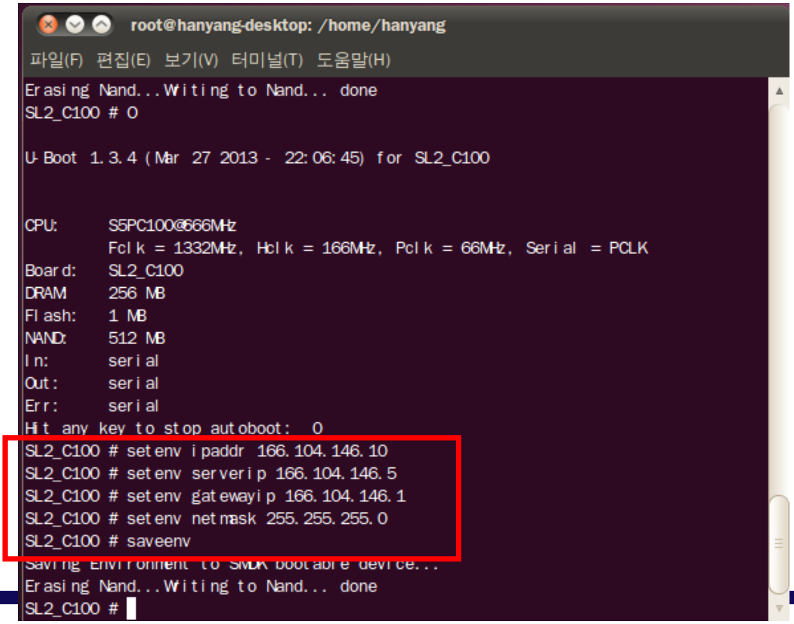
  # saveenv

- **Check network settings**

  # printenv

- **Check connection**

  # ping 166.104.146.5



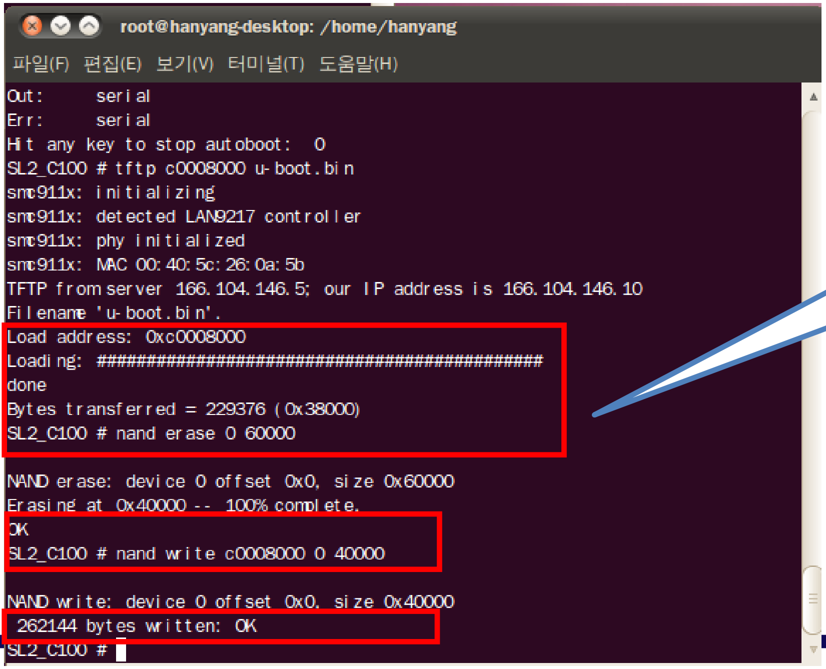
root@hanyang-desktop: /home/hanyang
파일(F) 편집(E) 보기(V) 터미널(T) 도움말(H)

```
Erasing Nand...Writing to Nand... done
SL2_C100 # 0

U-Boot 1.3.4 (Mar 27 2013 - 22:06:45) for SL2_C100


CPU:     S5PC100@666MHz
         Fclk = 1332MHz, Hclk = 166MHz, Pclk = 66MHz, Serial = PCLK
Board:   SL2_C100
DRAM     256 MB
Flash:   1 MB
NAND:    512 MB
In:      serial
Out:     serial
Err:     serial
Hit any key to stop autoboot:  0
SL2_C100 # setenv ipaddr 166.104.146.10
SL2_C100 # setenv serverip 166.104.146.5
SL2_C100 # setenv gatewayip 166.104.146.1
SL2_C100 # setenv netmask 255.255.255.0
SL2_C100 # saveenv
Saving Environment to SMDK bootable device...
Erasing Nand...Writing to Nand... done
SL2_C100 #
```

U-boot Porting

- **U-boot write using TFTP**
  - Write u-boot to nand with tftp using Network

```
# tftp c0008000 u-boot.bin
# nand erase 0 60000
# nand write c0008000 0 40000
```



root@hanyang-desktop: /home/hanyang
파일(F) 편집(E) 보기(V) 터미널(T) 도움말(H)

```
Out:      serial
Err:      serial
Hit any key to stop autoboot:  0
SL2_C100 # tftp c0008000 u-boot.bin
smc911x: initializing
smc911x: detected LAN9217 controller
smc911x: phy initialized
smc911x: MAC 00:40:5c:26:0a:5b
TFTP from server 166.104.146.5; our IP address is 166.104.146.10
Filename 'u-boot.bin'.
Load address: 0xc0008000
Loading: #############################################
done
Bytes transferred = 229376 (0x38000)
SL2_C100 # nand erase 0 60000

NAND erase: device 0 offset 0x0, size 0x60000
Erasing at 0x40000 --  100% complete.
OK
SL2_C100 # nand write c0008000 0 40000

NAND write: device 0 offset 0x0, size 0x40000
 262144 bytes written: OK
SL2_C100 #
```

Make sure that
Load, Erase, and Write
operations are correct!

U-boot Porting

- **Reset board and press ‘Enter’**
  - Check if it has been changed to “Student ID #“

- **U-boot write complete**

VPOS BOOTING

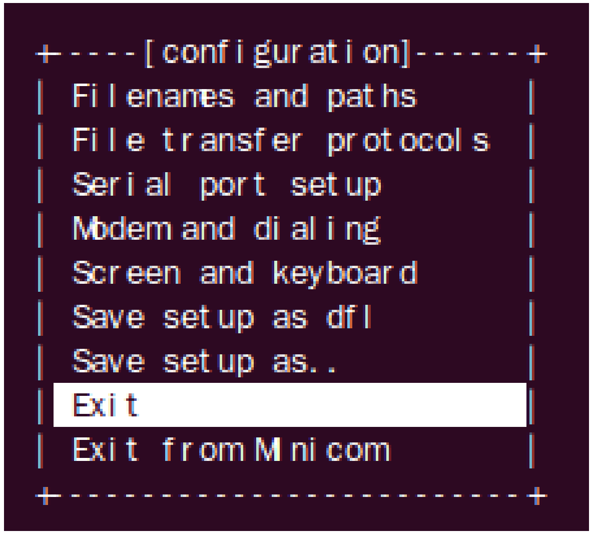
Prepare vpos Booting

- **Copy vpos.bin to /tftpboot**
  - Copy vpos.bin to /tftpboot

```
>> cp vpos.bin /tftpboot
```

Minicom

- **Connect to Board with minicom**
  (1) >> minicom
  (2) >> minicom –s and ‘Exit’



```
+----[configuration]------+
| Filenames and paths     |
| File transfer protocols |
| Serial port setup       |
| Modem and dialing       |
| Screen and keyboard     |
| Save setup as dfl       |
| Save setup as..         |
| Exit                    |
| Exit from Minicom       |
+-------------------------+
```

Download VPOS Kernel (not using NAND)

- **In this practice, we write it directly to SDRAM without being stored in NAND**

- **Kernel write using TFTP**
  - Write vpos kernel to SDRAM with tftp using network

    ```
    # tftp c0008000 vpos.bin
    ```

  - Booting command

    ```
    # bootm c0008000
    ```

Download VPOS kernel (using NAND)

- **Kernel write using TFTP**
  - Write vpos kernel to nand with tftp using network

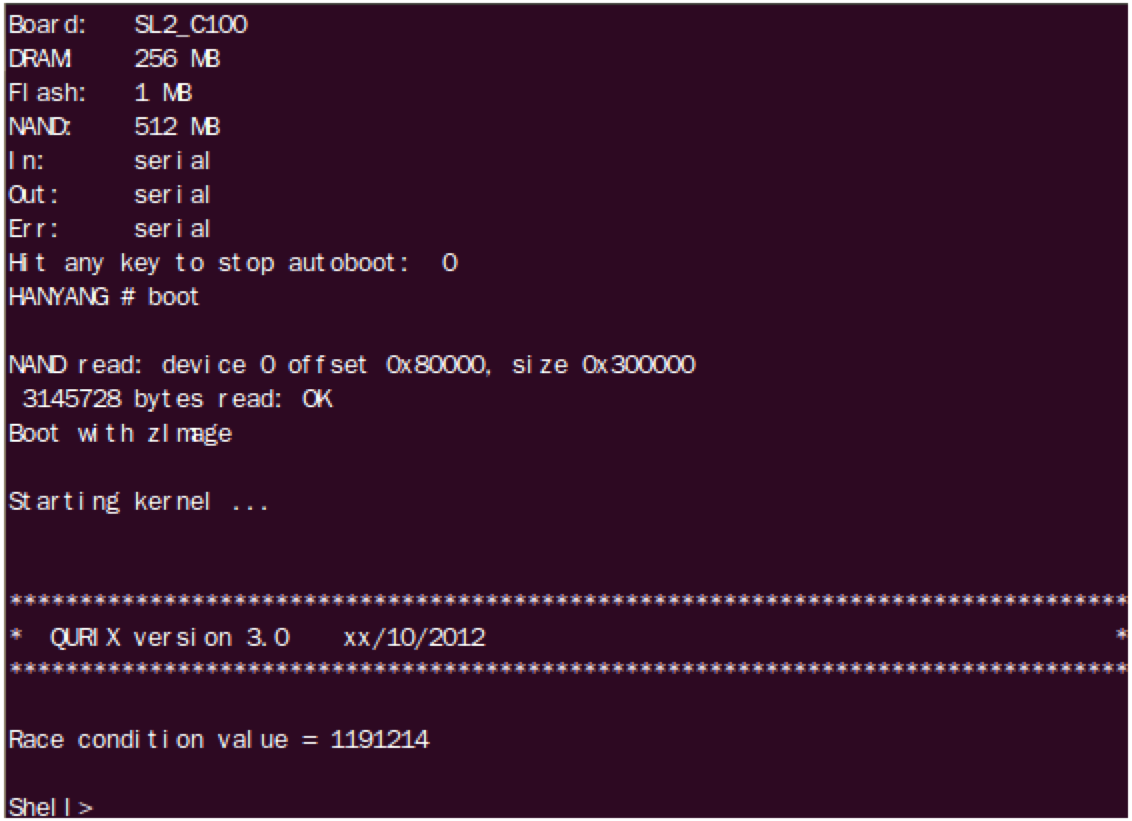
```
# tftp c0008000 vpos.bin
# nand erase 80000 400000
# nand write c0008000 80000 400000
# setenv bootcmd nand read c0008000 80000 300000\;bootm
c0008000
# saveenv
```

  - Booting command

```
# boot
```

VPOS Booting

- VPOS boot screen

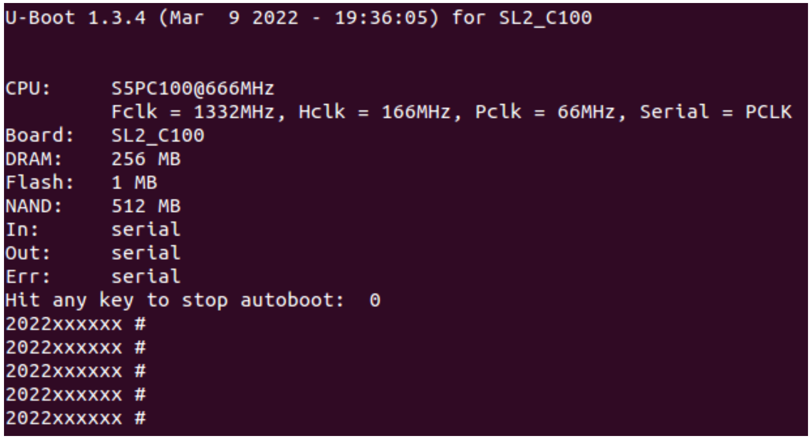


```
Board:    SL2_C100
DRAM      256 MB
Flash:    1 MB
NAND:     512 MB
In:       serial
Out:      serial
Err:      serial
Hit any key to stop autoboot:  0
HANYANG # boot

NAND read: device 0 offset 0x80000, size 0x300000
 3145728 bytes read: OK
Boot with zImage

Starting kernel ...


********************************************************************************
*  QURIX version 3.0    xx/10/2012                                             *
********************************************************************************

Race condition value = 1191214

Shell>
```

ASSIGNMENT

Assignment

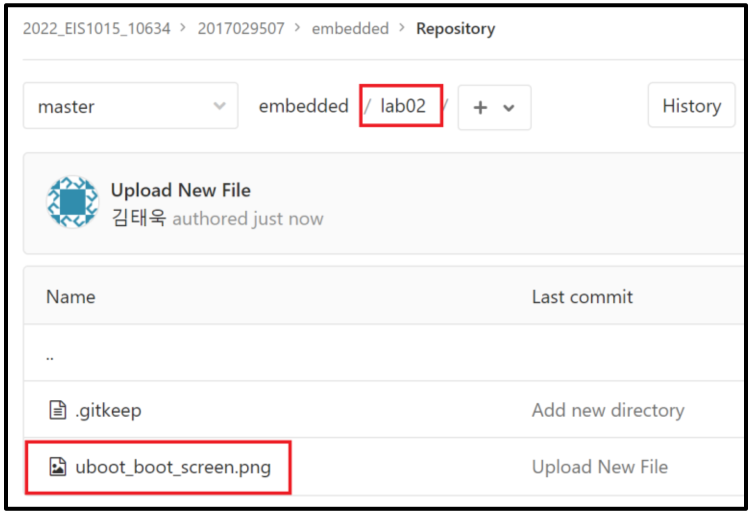
Upload modified u-boot boot screen on your gitlab



```
U-Boot 1.3.4 (Mar  9 2022 - 19:36:05) for SL2_C100


CPU:     S5PC100@666MHz
         Fclk = 1332MHz, Hclk = 166MHz, Pclk = 66MHz, Serial = PCLK
Board:   SL2_C100
DRAM:    256 MB
Flash:   1 MB
NAND:    512 MB
In:      serial
Out:     serial
Err:     serial
Hit any key to stop autoboot:  0
2022xxxxxx #
2022xxxxxx #
2022xxxxxx #
2022xxxxxx #
2022xxxxxx #
```

example image file to upload

(Student ID must be shown)



2022_EIS1015_10634 › 2017029507 › embedded › **Repository**

master ⌄ embedded / lab02 / + ⌄ History

Upload New File
김태욱 authored just now

Name	Last commit
..	
.gitkeep	Add new directory
uboot_boot_screen.png	Upload New File

create “lab02” directory and

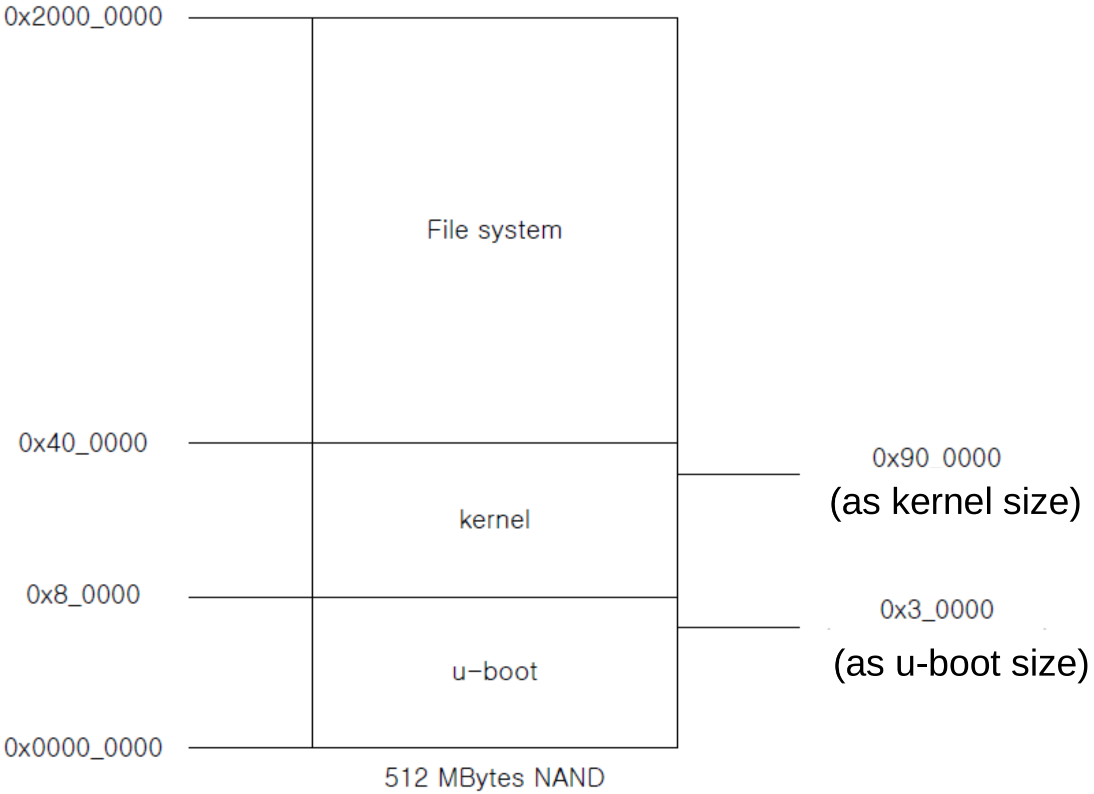
upload uboot screen file

Deadline : 3/18 11:59pm

APPENDIX

Location of U-boot and Kernel in NAND Flash

- NAND Flash Address Map of SYS-LAB II board



0x2000_0000

File system

0x40_0000

kernel

0x8_0000

u−boot

0x0000_0000

512 MBytes NAND

0x90_0000
(as kernel size)

0x3_0000
(as u-boot size)

Location of U-boot and Kernel in mDDR

- **What is mDDR?**
  - Double Data rate synchronous DRAM for mobile computers
  - Low power consumption than conventional DDR SDRAM

- **U-boot**
  - Physical address : 0x27e00000
  - Virtual address : 0xc7e00000

- **Kernel**
  - Physical address : 0x20008000
  - Virtual address : 0xc0008000

U-Boot Command

- **What is the meaning of "tftp c0008000 file_name"**
  - Write a file at 0xc0008000 of SDRAM
  - If only u-boot is booted, kernel memory space is empty
    So we can temporarily write data to that empty memory space
  - Since u-boot uses MMU, it accesses memory with virtual address

- **What is the meaning of "nand write c0008000 0 40000"**
  - Write the data in 0xc0008000 of SDRAM to 0x0 of NAND Flash
  - U-boot or Vpos are temporarily stored in 0xc0008000 of SDRAM
    So, we write data to the NAND Flash to store data permanently

Thank you