



Operating System Practice– Lab 1: Developing Embedded Systems

Che-Wei Chang

chewei@mail.cgu.edu.tw

Department of Computer Science and Information
Engineering, Chang Gung University



Preparation

Notices

- ▶ No food, no drink
- ▶ The evaluation boards are quite expensive
- ▶ Do not do anything else to crash the PC
- ▶ Do not update the OS nor tools to keep the consistency
- ▶ Remember the number of your evaluation board
 - Check the items before you use them
 - Check the items before you return them
- ▶ No rubbish

What are We Going to Do?

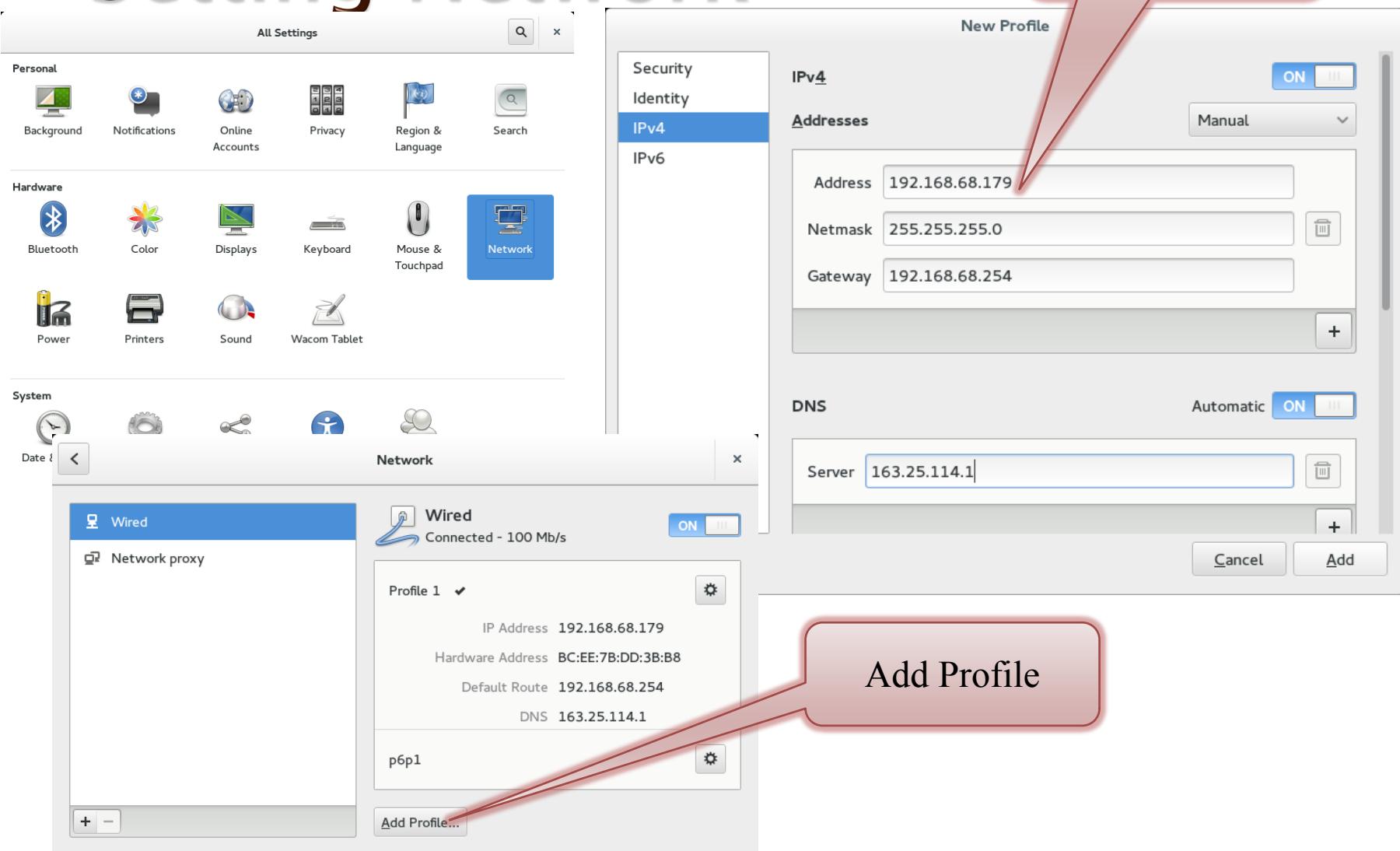
- ▶ Build Cross Development Toolchain
- ▶ Build Linux Kernel
 - ➔ Check Point 1: uImage
- ▶ Setup tftp Server
- ▶ Setup NFS Server
 - ➔ Check Point 2: Test the Services
- ▶ Setup Target Board
- ▶ Download Linux Kernel
 - ➔ Check Point 3: Try the Linux Kernel

Fedora Linux

- ▶ The Fedora Project was created in late 2003
- ▶ We are using the version 20
- ▶ Package manager: RPM
- ▶ Update method: Yum
- ▶ Default user interface: GNOME 3
 - Password: 123456
 - Select the language: Taiwan
 - WindowsKey+Space to change the input language
 - Activities → Search: terminal → to get the terminal
 - Edit → Profile Preferences → Colors → Uncheck “use colors from system theme”
 - Click the icon at the right-top corner for network setting



Setting Network



vi— A Screen-Oriented Text Editor

- ▶ vi is widely supported by Unix-like operating system
- ▶ Normal mode
 - Move, search, copy, paste, delete,...
 - Press i, I, a, A, o, O,... to change to the insert mode
 - Press : for the command mode
- ▶ Command mode
 - Save, quit, load, split,...
 - After enter the command, it will be back to the normal mode
- ▶ Insert mode
 - Move and input anything
 - Press ESC to go back to the normal mode

vi Commands

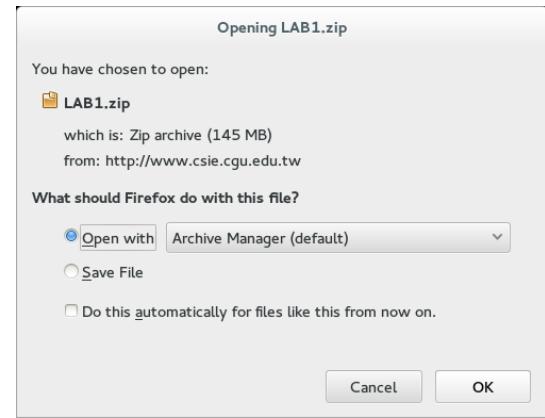
- ▶ Press ‘i’ to get the insert mode
- ▶ Key-in anything
- ▶ Press ‘ESC’ to go back the normal mode
- ▶ Press ‘:→w→q→ENTER” to save and quit
- ▶ Please search for some tutorial of vi and study by yourself



Lab1: Build the Linux Kernel for TI OMAP 5912

Download Files

- ▶ Download the tools from the course website and extract the files



Extract

Create Folder

/home/csie/LAB1

Extract

The diagram illustrates the workflow for extracting files:

- A red speech bubble labeled "Extract" points to a file manager window showing a list of files in a "LAB" folder.
- An arrow points from the file manager to an "Extract" dialog box. This dialog box shows a list of files to be extracted and various extraction options: "All files" (selected), "Keep directory structure" (checked), and "Do not overwrite newer files" (unchecked).
- An arrow points from the "Extract" dialog box to a folder path: "/home/csie/LAB1".
- An arrow points from the folder path to another "Extract" button, which is highlighted with a red speech bubble labeled "Extract".

Download Files

- ▶ You will need the following files
 - linux-2.6.12-omap.tgz → the kernel source code
 - gcc-3.3.2.tar → some gcc extension for this lab
 - mkimage → some script which is used when compiling kernel
 - rootfsosk.tar.bz2 → the content of the root filesystem
- ▶ You need the root privilege for the following actions
 - *su* (the password is 123456) → change to root
 - *cd /home/csie/LAB1*
 - *cp linux-2.6.12-omap.tgz /opt/linux-2.6.12-omap.tgz*
 - *cp gcc-3.3.2.tar /opt/gcc-3.3.2.tar*
 - *chmod +x mkimage*
 - *cd /opt*
 - *tar xvf gcc-3.3.2.tar*
 - *tar zxvf linux-2.6.12-omap.tgz*
 - *cp /home/csie/LAB1/mkimage /opt/usr/local/arm/3.3.2/bin/mkimage*

Prepare the Compiling Environment

- ▶ Set Path
 - *export PATH=\$PATH:/opt/usr/local/arm/3.3.2/bin* → for every terminal session, before you compile the kernel
 - *export LANG=en*
- ▶ Install Tools
 - *yum -y install gcc* → compiler tools
 - *yum -y install glibc.i686* → library for 32bit Linux kernel
 - *yum -y install minicom* → minicom is the utility for the serial port connection

Build the Linux Kernel

- ▶ Go to the kernel source directory (be the root)
 - *cd /opt/linux-2.6.12*
- ▶ Set the kernel configuration
 - *make omap_osk_5912_defconfig*
- ▶ Compile the kernel
 - *make ulmage*
- ▶ Prepare the root filesystem
 - *cp /home/csie/LAB1/rootfsosk.tar.bz2 /tmp/rootfsosk.tar.bz2*
 - *cd /tmp*
 - *tar jxvf rootfsosk.tar.bz2*

Check Point 1

- ▶ Now, you should have the compiled kernel
- ▶ The kernel image is at:
`/opt/linux-2.6.12/arch/arm/boot/uImage`
- ▶ The root filesystem for the evaluation board is at:
`/tmp/roorfs2.6`

Set the Network Services

- ▶ Disable the Firewall (it is not a good idea, only for this lab exercise)
 - *systemctl stop firewalld*
 - *systemctl disable firewalld*
- ▶ Set the TFTP Service
 - *yum -y install tftp-server tftp* → tftp is used to download kernel image
 - *vi /etc/xinetd.d/tftp*
 - Find `disable = yes`
 - Change it to `disable = no`
 - ~~*/sbin/chkconfig xinetd on*~~
 - *systemctl start tftp.socket*
 - ~~*/sbin/service xinetd start*~~
 - *systemctl enable tftp.socket*
- ▶ Set the NFS Service
 - *yum -y install nfs-utils* → nfs for the root filesystem
 - *vi /etc/exports*
 - Add the line `/tmp/rootfs2.6 *(rw,fsid=1,no_root_squash)`
 - *exportfs -rv*
 - *systemctl start rpcbind.service*
 - *systemctl start nfs-mountd.service*

Test the Network Services

- ▶ You need a friend for the following test
 - One be the server and the other be the client
 - Switch the roles and do it again
- ▶ Test TFTP
 - Server side:
 - *vi /var/lib/tftpboot/testfile* → and then key something
 - Client side:
 - *tftp 192.168.68.xxx* (xxx is for the server IP)
 - *get testfile*
 - *quit*
 - *cat testfile*
- ▶ Test NFS
 - Server side:
 - Client side:
 - *mkdir /home/csie/nfstest*
 - *mount -t nfs 192.168.68.xxx:/tmp/rootfs2.6 /home/csie/nfstest*
 - *cd /home/csie/nfstest*
 - *ls*
 - *cd /*
 - *umount /home/csie/nfstest*

Check Point 2

- ▶ Now, you have enabled the TFTP and NFS services on your PC
- ▶ TFTP and NFS are properly working now

Set the Minicom (1 / 3)

- ▶ Enter the setting menu
 - *minicom -s*

```
+----[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
+-----+
```

- ▶ Serial port setup → press the letter to change it

```
+-----+
| A -  Serial Device      : /dev/ttyS0
|
| 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?  |
+-----+
```

Set the Minicom (2/3)

► Modem and dialing

```
+-----[Modem and dialing parameter setup]-----+
| A - Init string .....
| B - Reset string .....
| C - Dialing prefix #1....
| D - Dialing suffix #1....
| E - Dialing prefix #2.... ATDP
| F - Dialing suffix #2.... ^M
| G - Dialing prefix #3.... ATX1DT
| H - Dialing suffix #3.... ;X4D^M
| I - Connect string ..... CONNECT
| J - No connect strings .. NO CARRIER           BUSY
|                               NO DIALTONE          VOICE
| K - Hang-up string ..... ~~~++~ATH^M
| L - Dial cancel string .. ^M
|
| M - Dial time ..... 45      Q - Auto bps detect ..... No
| N - Delay before redial . 2      R - Modem has DCD line .. Yes
| O - Number of tries ..... 10     S - Status line shows ... DTE speed
| P - DTR drop time (0=no). 1      T - Multi-line untag .... No
|
| Change which setting? [ ]  Return or Esc to exit. Edit A+B to get defaults.
+-----+
```

```
+-----[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
+-----+
```

Set the Minicom (3/3)

- ▶ Save and leave the setting interface

```
+----[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        |
+-----+
```

```
+----[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        |
+-----+
```

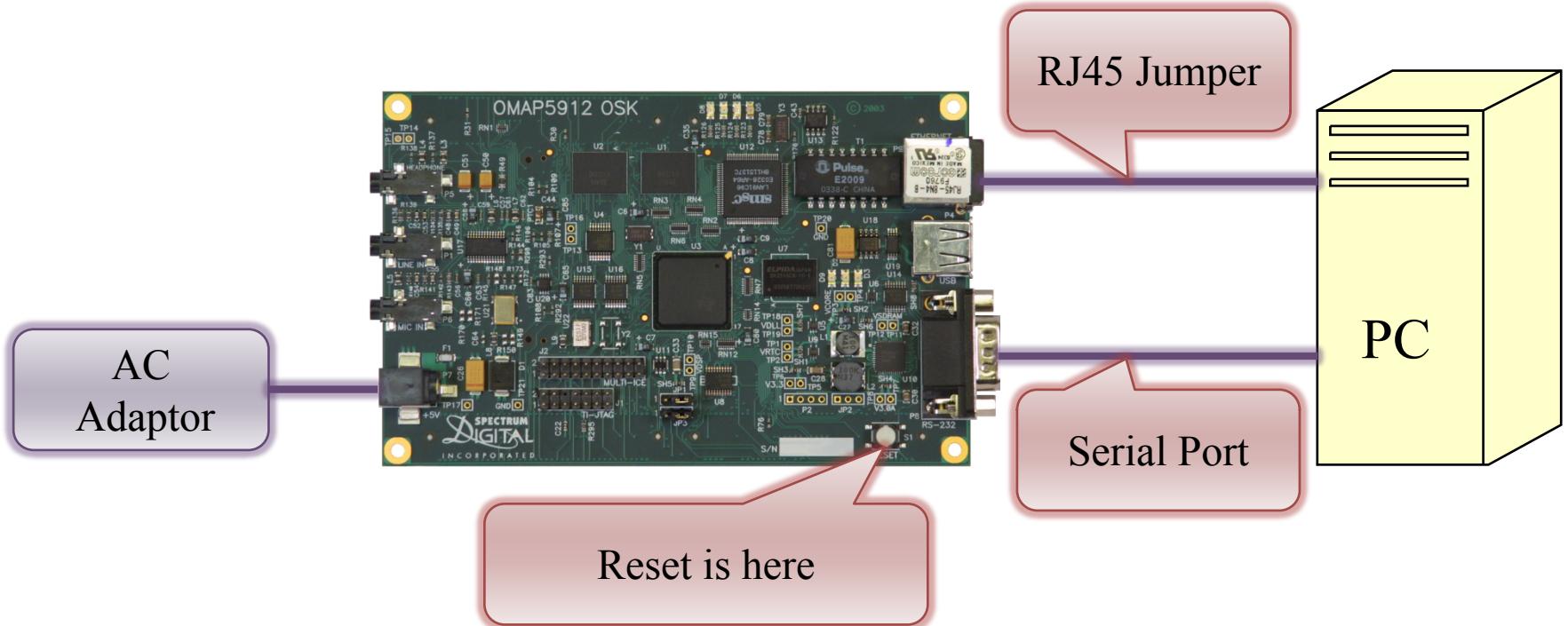
- ▶ Start and quit minicom
 - Start *minicom*
 - Quit *CTRL+A* → *Q*

```
Welcome to minicom 2.6.2
OPTIONS: I18n
Compiled on Aug  7 2013, 13:32:48.
Port /dev/ttyS0, 21:18:16

Press CTRL-A Z for help on special keys
```

Prepare for the Booting

- ▶ Copy the boot image for TFTP booting
 - `cp /opt/linux-2.6.12/arch/arm/boot/uImage /var/lib/tftpboot/uImage`
- ▶ Set the evaluation board as follows



Boot the Evaluation Board

- ▶ Start mimicom
 - *minicom*
- ▶ Press the reset button on the board
 - After the reset, immediately press any key on minicom terminal
 - You will get the following prompt

```
OMAP5912 OSK #
```

Download the New Kernel

- ▶ Set the boot configuration
 - *set ipaddr 192.168.68.yy* (evaluation board IP)
 - *set serverip 192.168.68.zz* (PC IP)
 - *set netmask 255.255.255.0*
 - *set gatewayip 192.168.68.254*
 - *set ethaddr 00-0e-99-xx-xx-xx*
 - *set bootargs console=ttyS0,115200n8 rw ip=192.168.68.yy root=/dev/nfs nfsroot=192.168.68.zz:/tmp/rootfs2.6,v3*
 - *printenv* → double check the setting

```
OMAP5912 OSK # printenv
bootdelay=3
baudrate=115200
bootfile="uImage"
bootcmd=bootm 0x100000
ipaddr=192.168.68.123
serverip=192.168.68.186
netmask=255.255.255.0
gatewayip=192.168.68.254
ethaddr=00-0e-99-02-0d-0b
stdin=serial
stdout=serial
stderr=serial
bootargs=console=ttyS0,115200n8 rw ip=192.168.68.123 root=/dev/nfs nfsroot=192.168.68.186:/tmp/rootfs2.6,v3

Environment size: 337/131068 bytes
OMAP5912 OSK #
```

- *saveenv* → if everything is correct → be careful, do not crash the entire system

Boot the New Kernel and Mount the NFS Root Filesystem

- ▶ Download the kernel: `tftpboot 0x10000000 ulimage`

- ▶ Boot the OS: *bootm 0x10000000*

```
Looking up port of RPC 100003/3 on 192.168.68.186
Looking up port of RPC 100005/3 on 192.168.68.186
VFS: Mounted root (nfs filesystem).
Freeing init memory: 112K
init started: BusyBox v1.00-pre8 (2004.03.05-22:18+0000) multi-call binary

*****
Starting System Init for OMAP59120SK
*****
```





Done!
Or Bugs!?

Common Mistakes

- ▶ *su* and *export* should be used whenever a new terminal is created
 - If you extract the root file system by the user csie, there will be an error when you boot the board to mount the NFS root file system
 - Reboot the computer and do everything again
 - If you do not export the path of the tools, you will get some error when you compile the kernel module
- ▶ Please read the error message if you type something wrong
- ▶ UART: it should be connected to the bottom port
- ▶ Ethernet: do check the IP is correct
- ▶ Some evaluation boards were tested to be good: 1, 7, 9, 10, 11, ~~12~~, 15, 19, 20

Grading this Exercise

- ▶ Attend and understand this exercise: 10%
- ▶ Check point 1: 10%
- ▶ Check point 2: 10%
- ▶ Final results: 20%
- ▶ Report before the exercise: 25%
- ▶ Report after the exercise: 25%
- ▶ Bonus: 30%

Report Requirements

- ▶ Report before the exercise: 25%
 - Only two pages, 12-pt font size
 - Deadline is 20:00, 2019/05/15
 - File name: OSP-Lab1-Study-StudentID
 - File type: PDF or Word
 - Send it to my email: kf200660306@gmail.com
 - Email title: OSP Lab1 Study StudentID
- ▶ Report after the exercise: 25%
 - Only two pages, 12-pt font size
 - Deadline is 20:00, 2019/05/22
 - File name: OSP-Lab1-Report-GroupID
 - File type: PDF or Word
 - Send it to my email: kf200660306@gmail.com
 - Email title: OSP Lab1 Report GroupID
 - Remember to list all student IDs of your group
- ▶ Bonus: 20%
 - Try NFS after the class on your PC. I might ask you to do some demonstration
 - Send another report: OSP Lab1 Bonus StudentID
 - Deadline is 20:00, 2019/05/22