

## 실습 2-4: 루트파일시스템

### 1. 준비

커널 속에 모듈들이 여러개 있다.  
모듈은 커널에 속한 디바이스들을 컴파일 하는 것들이다.

장비: USB-TTL Serial 3.3V 케이블, SD card, SD card reader기

\$HOME/ESP2018/chap02/rootfs 디렉토리를 만들고 이동한다.

### 2. BBB용으로 빌드된 루트파일시스템 다운로드

```
ohheum@ubuntu:~/ESP2018/chap02/rootfs$ wget -c https://rcn-ee.com/rootfs/eewiki/minfs/debian-8.4-minimal-armhf-2016-04-22.tar.xz
```

```
ohheum@ubuntu:~/ESP2018/chap02/rootfs$ tar xf debian-8.4-minimal-armhf-2016-04-22.tar.xz
```

```
ohheum@ubuntu:~/ESP2018/chap02/rootfs$ mkdir target
```

```
ohheum@ubuntu:~/ESP2018/chap02/rootfs$ sudo tar xfv debian-8.4-minimal-armhf-2016-04-22/armhf-rootfs-debian-jessie.tar -C target/
```

```
ohheum@ubuntu:~/ESP2018/chap02/rootfs$ sudo chown -R ohheum:ohheum target/
[sudo] password for ohheum: /* 실습을 조금 편하게 하기위한 편법임. */
```

```
ohheum@ubuntu:~/ESP2018/chap02/rootfs$ cd ~/ESP2018/chap02/kernel/linux-stable
```

```
ohheum@ubuntu:~/ESP2018/chap02/kernel/linux-stable$ echo $ARCH
arm
```

```
ohheum@ubuntu:~/ESP2018/chap02/kernel/linux-stable$ echo $CROSS_COMPILE
arm-linux-gnueabihf-
```

지금은 타겟 디렉토리 안에 필요한 모듈을 넣어야한다.

```
ohheum@ubuntu:~/ESP2018/chap02/kernel/linux-stable$ make INSTALL_MOD_PATH=~/ESP2018/chap02/rootfs/target/ modules_install
```

zImage와 별개로 컴파일된 모듈들을 모아서 인스톨 해준다  
각각을 찾기가 힘들기 때문이다.

### 3. 커널과 dtb 파일을 tftp 디렉토리에 복사

```
ohheum@ubuntu:~/ESP2018/chap02/kernel/linux-stable$ sudo apt-get install tftpd-hpa
```

```
ohheum@ubuntu:~/ESP2018/chap02/kernel/linux-stable$ sudo cp arch/arm/boot/zImage /var/lib/tftpboot/
[sudo] password for ohheum:
```

```
ohheum@ubuntu:~/ESP2018/chap02/kernel/linux-stable$ sudo cp arch/arm/boot/dts/am335x-boneblack.dtb /var/lib/tftpboot/
```

b는 컴파일된 바이너리 파일. s는 소스파일

### 4. NFS 서버를 설치하고 준비

```
ohheum@ubuntu:~/ESP2018/chap02/kernel/linux-stable$ sudo apt-get install nfs-kernel-server
```

```
ohheum@ubuntu:~/ESP2018/chap02/kernel/linux-stable$ sudo gedit /etc/exports
```

**/etc/exports** 파일을 다음과 같이 수정한다.

etc : 각종 설정값을 저장해두는 곳이다.

```
ohheum@ubuntu:~/ESP2018/chap02/kernel/linux-stable$ cat /etc/exports
```

```
# /etc/exports: the access control list for filesystems which may be exported
# to NFS clients. See exports(5).
```

```
# Example for NFSv2 and NFSv3:
```

```
# /srv/homes hostname1(rw,sync,no_subtree_check)
hostname2(ro,sync,no_subtree_check)
```

```
# Example for NFSv4:
```

```
# /srv/nfs4 gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes gss/krb5i(rw,sync,no_subtree_check)
```

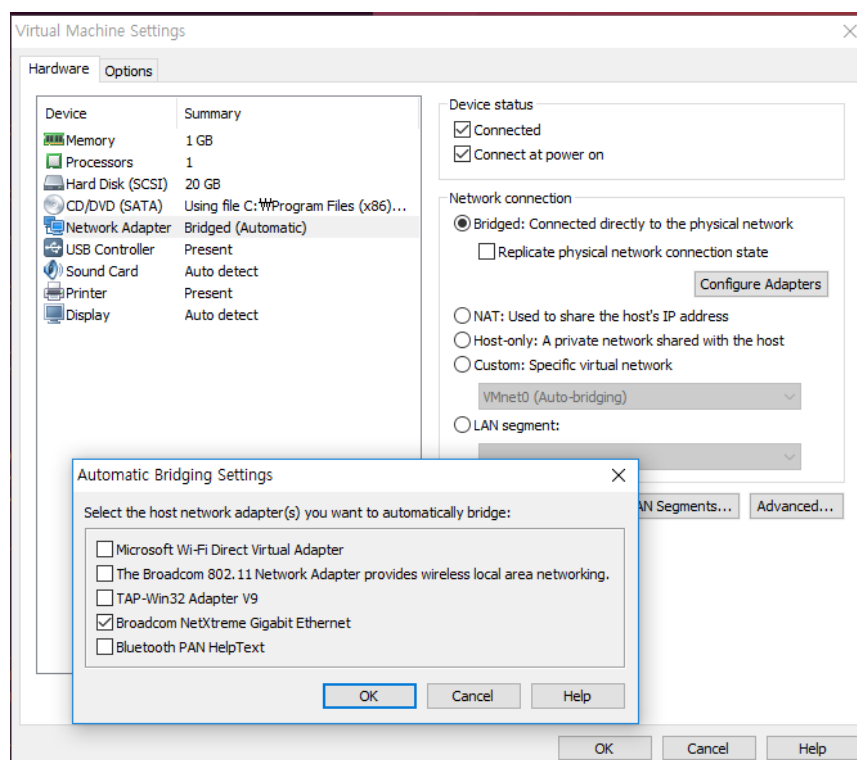
```
#
/home/ohheum/ESP2018/chap02/rootfs/target
192.168.0.100(rw,no_root_squash,no_subtree_check)
```

한 줄로 작성해야 함

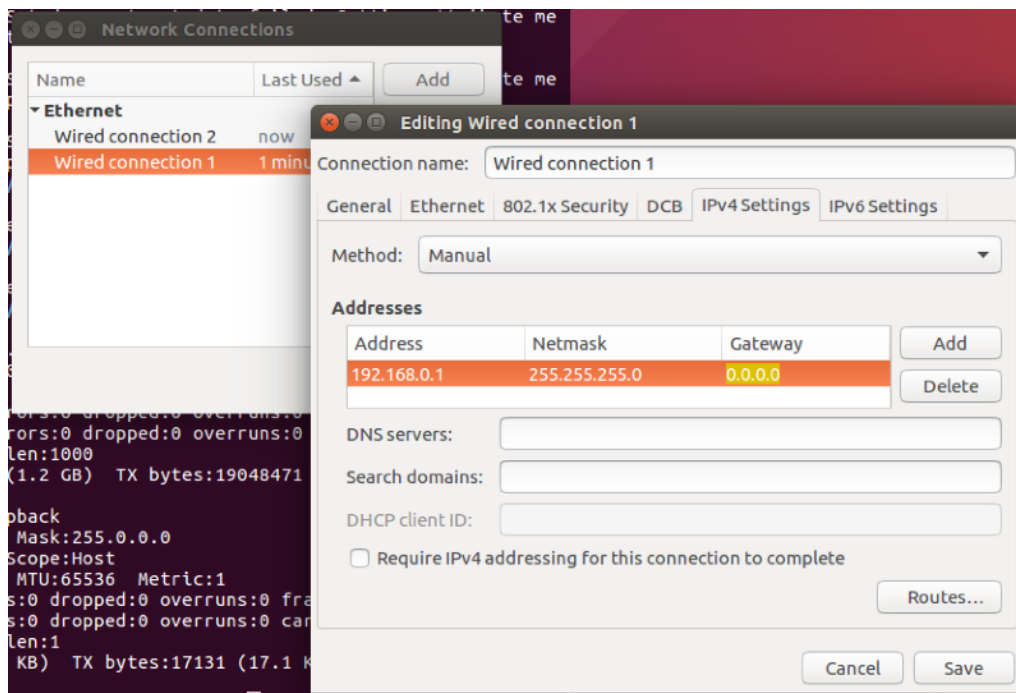
```
ohheum@ubuntu:~/ESP2018/chap02/kernel/linux-stable$ sudo systemctl restart nfs-kernel-server
```

## 5. BBB와 Ubuntu를 LAN 케이블로 연결하기

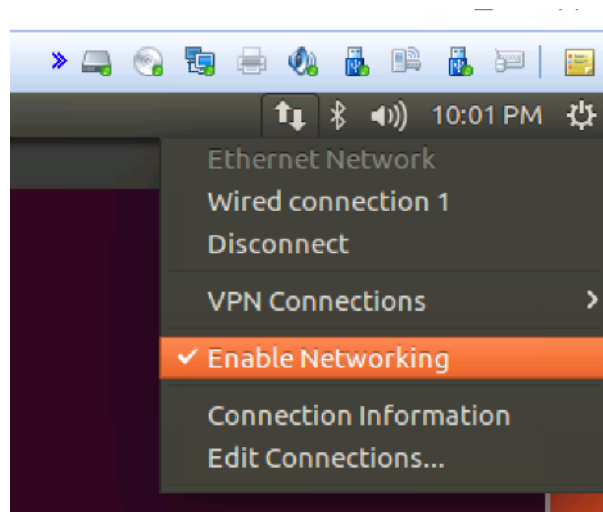
BBB와 Vmware가 설치된 노트북을 랜 케이블로 직접 서로 연결한다. Vmware의 Player 메뉴에서 “Removable Devices -> Network Adapter -> Setting”을 선택하고 다음 그림과 같이 Bridged 모드를 선택한다. 그런 다음 Configure Adapters를 누르고 BBB와의 연결에 사용된 Ethernet Adapter를 제외한 나머지 모든 항목의 체크를 해제한다. (아래 그림은 PC의 내장 Ethernet Adapter가 Broadcom Adapter인 경우이다.)



그런 다음 Ubuntu의 IPv4 세팅을 아래 그림과 같이 세팅한다 (IP Address: 192.168.0.1, Netmask: 255.255.255.0, Gateway: 0.0.0.0).



즉시 연결이 설정되지 않으면 아래 그림에서 Enable Networking을 한 번 해제 했다가 다시 켜면 연결이 수립된다.



## 6. NFS를 통한 BBB 부팅

Serial-To-USB 케이블로 BBB와 Ubuntu를 연결하고 다음과 같이 picocom을 실행한다.

```
ohheum@ubuntu:~$ sudo picocom -b 115200 /dev/ttyUSB0
picocom v1.7
```

```
port is : /dev/ttyUSB0
```

```
flowcontrol      : none
baudrate is      : 115200
parity is        : none
databits are     : 8
escape is        : C-a
local echo is    : no
noinit is        : no
noreset is       : no
nolock is        : no
send_cmd is      : sz -vv
receive_cmd is   : rz -vv
imap is          :
omap is          :
emap is          : crcrlf,delbs,
```

Terminal ready

// BBB에 SD카드를 삽입하고 boot 버튼을 누른 상태로 power on한다.  
// 재빨리 space bar를 누른다.

```
U-Boot SPL 2017.01-dirty (Feb 12 2017 - 13:57:57)
Trying to boot from MMC
reading args
spl_load_image_fat_os: error reading image args, err - -1
reading u-boot.img
reading u-boot.img
```

U-Boot 2017.01-dirty (Feb 12 2017 - 13:57:57 +0900)

```
Watchdog enabled
I2C:  ready
DRAM:  512 MiB
Reset Source: Power-on reset has occurred.
MMC:  OMAP SD/MMC: 0, OMAP SD/MMC: 1
Using default environment
```

```
Net:  <ethaddr> not set. Validating first E-fuse MAC
cpsw, usb_ether
Press SPACE to abort autoboot in 2 seconds
```

=> **setenv ipaddr 192.168.0.100**

=> **setenv serverip 192.168.0.1**

=> **tftp 0x81000000 zImage** tftp 프로토콜로 zImage(커널)를 읽어온다..  
저장할 위치

```
link up on port 0, speed 100, full duplex
Using cpsw device
TFTP from server 192.168.0.1; our IP address is 192.168.0.100
Filename 'zImage'.
Load address: 0x81000000
Loading: #####
#####
#####
#####
1.4 MiB/s
done
```

Bytes transferred = 3455144 (34b8a8 hex)

=> **tftp 0x82000000 am335x-boneblack.dtb**

link up on port 0, speed 100, full duplex

Using cpsw device

TFTP from server 192.168.0.1; our IP address is 192.168.0.100

Filename 'am335x-boneblack.dtb'.

Load address: 0x82000000

Loading: ###

1.2 MiB/s

done

Bytes transferred = 30149 (75c5 hex)

=> **setenv bootargs root=/dev/nfs rw ip=192.168.0.100 console=tty00,115200  
nfsroot=192.168.0.1:/home/ohheum/ESP2018/chap02/rootfs/target**

=> **bootz 0x81000000 - 0x82000000**

Starting Serial Getty on tty00...

[ OK ] Started Serial Getty on tty00.

[ OK ] Reached target Login Prompts.

[ OK ] Stopped OpenBSD Secure Shell server.

[ OK ] Started LSB: set CPUFreq kernel parameters.

[ OK ] Started LSB: Apache2 web server.

Debian GNU/Linux 8 arm tty00

default username:password is [debian:temppwd]

arm login: **debian** // root로 login할 수도 있음. 패스워드는 root

Password:

Linux arm 4.1.10 #1 SMP Sat Feb 13 11:28:35 KST 2017 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.

debian@arm:~\$

// ubuntu의 ~/ESP2018/chap02/rootfs/target/home/debian 디렉토리에 아무 파일이나 만들고 BBB에서  
확인한다.

// toolchain할 때 build했던 test프로그램의 실행파일을 BBB에서 실행해본다.

debian@arm:~\$ logout

root로 login한 후 shutdown -h now로 BBB를 끈다.