

---

# 리눅스 커널 및 파일 시스템

- Chapter 07 -

---

# Contents

---

- I. 리눅스 커널 개요
- II. 리눅스 커널 컴파일
- III. 리눅스 파일 시스템
- IV. 리눅스 파일 시스템 설치

# 리눅스 커널 개요

## ▶ 리눅스 커널 (Linux Kernel)

### ▶ 운영체제에서 가장 기초적이고 핵심적인 기능을 담당

▷ 프로세스의 관리, CPU 스케줄링, 입출력제어

▷ 메모리, 파일, 주변장치와 같은 시스템 관리

### ▶ 부트로더가 하드웨어 초기화를 담당했다면 커널은 시스템 자원을 관리해 주는 기능

# 리눅스 커널 개요

## ▶ 리눅스 커널의 종류

### ▶ 모놀리틱(Monolithic) 커널

- ▶ 하나의 커널에 필요한 모든 기능이 통합
- ▶ 같은 메모리 공간에 필요한 기능이 존재
- ▶ 함수 호출 방식으로 커널에서 제공하는 기능에 접근
- ▶ 구현이 간편하고 효율적이지만 포팅과 확장이 어려움



### ▶ 마이크로(Micro) 커널

- ▶ 필요한 기능들은 작은 서버 모듈로 나뉘어 설계
- ▶ 서버를 관리할 수 있는 최소의 크기
- ▶ 하드웨어 환경에 따라 기능의 확장과 기능 재구성이 용이
- ▶ 서비스를 사용하는 과정에서 여러 번의 메시지 전송과 컨텍스트 스위칭이 발생하는 단점
  - 높은 자원 사용율

# 리눅스 커널 개요

## ▶ 리눅스 커널의 버전 표기법

### ▶ 커널버전 : Linux-2.6.18

- ▷ 위 커널 버전의 첫 번째 번호 '2'는 Major number로 주 버전을 의미
- ▷ 두 번째 번호 '6'은 Minor number이며 해당 번호가 홀수/짝수일 때의 의미가 다름
  - 짝수는 안정화된 버전이며, 홀수는 개발 버전(3.0 이상부터는 의미 없음)
- ▷ 마지막 세 번째 해당하는 부분은 Patch된 횟수
  - 예제에서는 18번 Patch되었음

### ▶ 커널 버전 확인법

- ▷ `uname -r` 명령어를 이용해서 커널의 버전 확인
  - Linux ubuntu 3.13
- ▷ 리눅스 커널 최신 버전: 4.19

# 리눅스 커널 개요

---

## ▶ 리눅스 커널의 역할

- ▶ 프로세스 관리(Process Management)
- ▶ 메모리 관리(Memory Management)
- ▶ 파일 시스템 관리(File System Management)
- ▶ 디바이스 관리(Device Management)
- ▶ 네트워크 관리(Network Management)

# 리눅스 커널 개요

## ▶ 리눅스 커널의 역할

### ▶ 프로세스 관리

- ▷ 하나의 프로그램은 하나 이상의 프로세스를 가짐
  - Thread 지원
- ▷ 커널은 프로세스에게 시스템 자원을 분배하고, 다른 자원을 침해할 수 없도록 관리
- ▷ 프로세스의 생성과 소멸에 대한 전반적인 내용을 관리

### ▶ 메모리 관리

- ▷ 선형 구조의 메모리 관리
- ▷ 시스템에서 구동되는 프로세스는 프로세스만의 독립적인 메모리를 소유
- ▷ 리눅스는 프로세스가 물리메모리에 직접적으로 데이터를 기록하는 것을 허용하지 않음
  - 가상 메모리 공간을 부여
- ▷ 반환된 메모리의 관리 등과 같은 기능을 수행

# 리눅스 커널 개요

## ▶ 리눅스 커널의 역할

### ▶ 파일시스템 관리

- ▶ 리눅스는 다양한 파일시스템을 지원하며 연결된 모든 디바이스는 파일로 표현
- ▶ YAFFS, FAT16/32, EXT2/3/4 등을 지원

### ▶ 디바이스 관리

- ▶ 부트로더에서 초기화된 이후 모든 디바이스는 커널의 관리를 받음
- ▶ 응용 프로그램에서 디바이스를 이용해야 하는 경우 커널을 통해야만 제어가 가능
- ▶ 사용하고자 하는 장치 드라이버에 따라서 문자, 블록, 네트워크 방식으로 분류

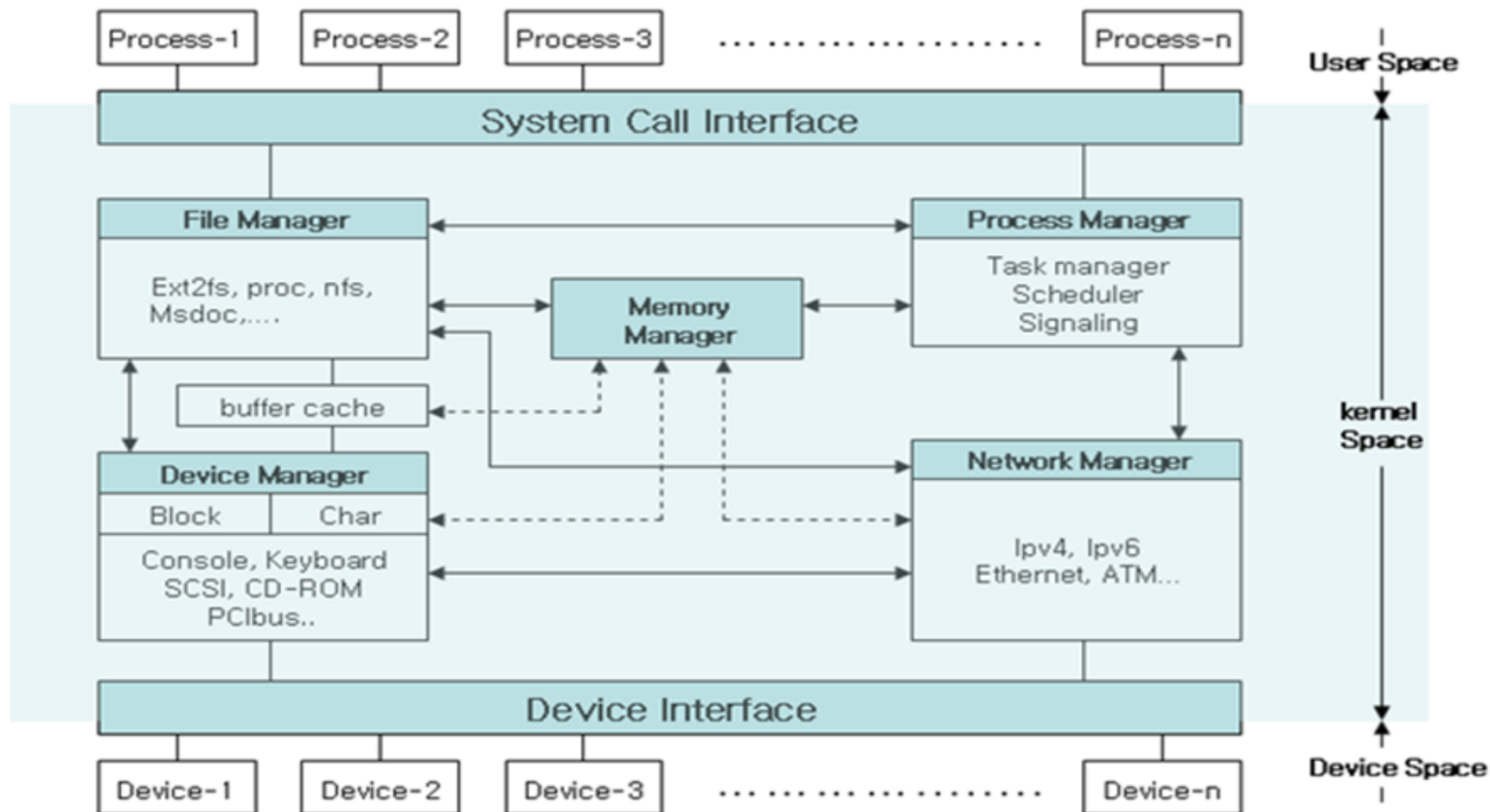
### ▶ 네트워크 관리

- ▶ 커널은 시스템 간의 정상적인 네트워크를 통해서 통신할 수 있도록 관리
- ▶ IPv4, IPv6, Ethernet, ATM, CDMA등이 지원



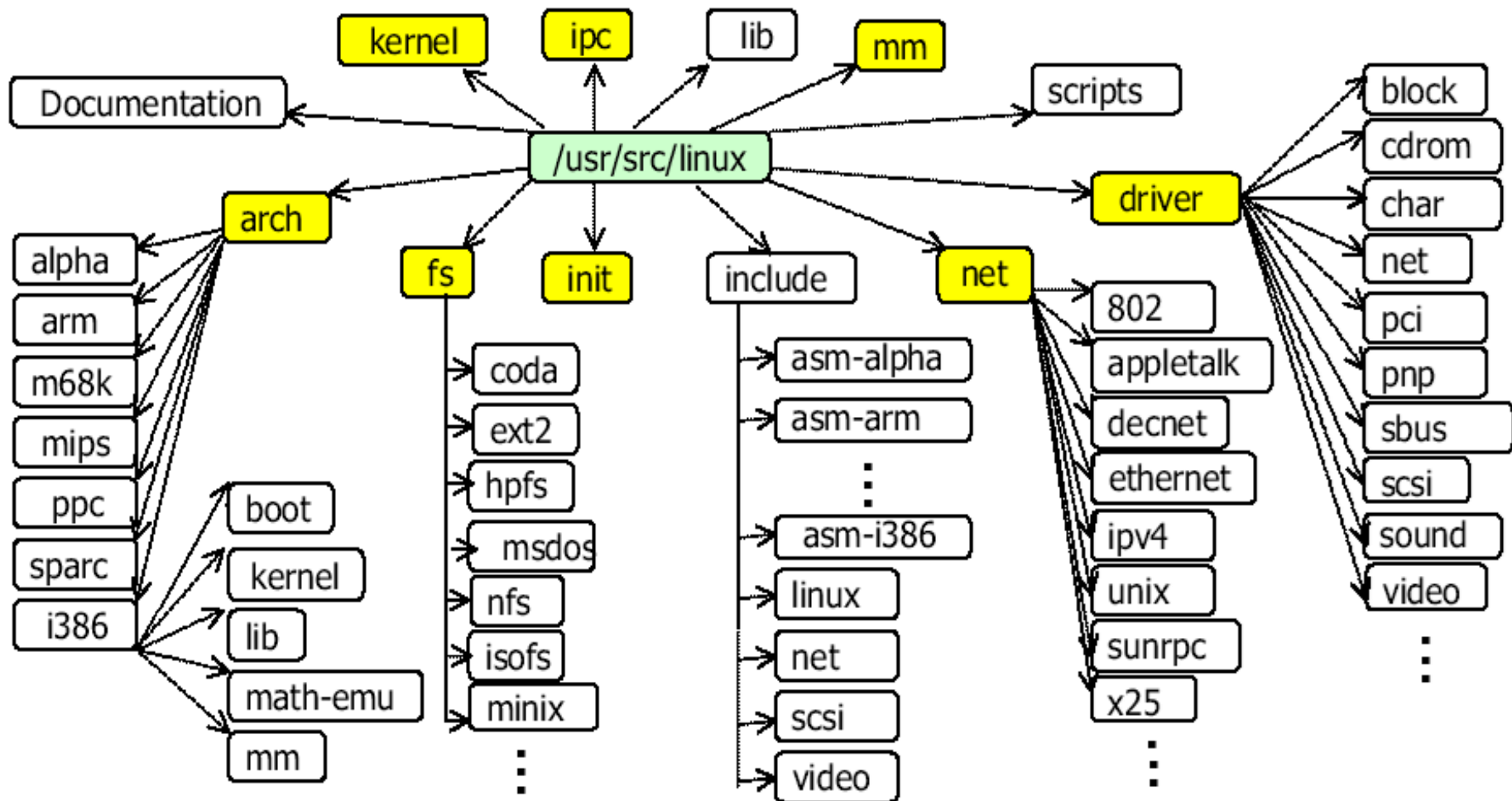
# 리눅스 커널 개요

## ▶ 리눅스 커널 Diagram



# 리눅스 커널 개요

## ▶ 커널 소스 디렉터리 구조



# 리눅스 커널 개요

## ▶ 커널 소스 디렉터리 구조

- ▶ arch: 프로세서 종류에 의존적인 부분(arch/i386, arch/alpha, arch/arm... )
  - ▶ arch/arm/boot/ : 부트 스트랩 코드의 내용
  - ▶ arch/arm/kernel/ : 하드웨어 의존적인 커널 관리 루틴, Trap과 인터럽트 처리 루틴, 문맥전환 (Context Switching) 루틴, 장치 설정과 초기화 루틴
  - ▶ arch/arm/mm/ : 하드웨어 의존적인 메모리 관리 루틴
- ▶ init: 하드웨어 비의존적 커널 초기화 (start\_kernel)
- ▶ kernel: 리눅스 커널의 중요 부분
  - ▶ 하드웨어 비의존적인 커널 관리 루틴
  - ▶ 태스크(fork, exit) 관련 시스템 호출 처리 루틴
  - ▶ 스케줄러, 시그널 핸들링, 시간 관리 루틴
- ▶ mm: 하드웨어 비의존적인 메모리 관리 루틴
  - ▶ 가상 메모리 관리
  - ▶ 페이징 기법, 스와핑 기법

# 리눅스 커널 개요

---

## ▶ 커널 소스 디렉터리 구조

### ▶ fs: 가상 파일 시스템 관리

▷ 파일시스템 관련 시스템 콜(open, read...) 루틴

### ▶ ipc: 프로세스간 통신(Inter Process Communication)

▷ 세마포어, 공유메모리, 메시지 큐

### ▶ lib: 커널 라이브러리 루틴

# 리눅스 커널 개요

## ▶ 커널 소스 디렉터리 구조

### ▶ driver: 디바이스 드라이버 루틴

- ▷ driver/block/ : 블록 디바이스
- ▷ driver/char/ : 문자 디바이스
- ▷ driver/net : 네트워크 디바이스
- ▷ driver/pci/ : PCI 버스 컨트롤러
- ▷ driver/sound/ : 사운드 카드
- ▷ driver/cdrom/ : CD-ROM 드라이버
- ▷ driver/scsi/ : SCSI 장치 드라이버

# 리눅스 커널 개요

## ▶ 커널 소스 디렉터리 구조

### ▶ net: 네트워크 관련 프로토콜과 드라이버

- ▷ Network protocols : TCP/IP, ARP등(device driver는 driver/net에 존재)

- ▷ Socket interface

### ▶ include: 커널을 위한 헤더파일

- ▷ include/linux/ : 하드웨어 비의존적인 헤더 파일

- ▷ include/asm-\*\*\*/ : 하드웨어 의존적인 헤더 파일

### ▶ modules: 커널 모듈 루틴

- ▷ 모듈의 동적 등록과 제거를 위한 부분 (insmod, modprobe, rmmod등)

### ▶ doc: 커널 관련 문서

# 리눅스 커널 컴파일

## ▶ 커널 소스 준비

### ▶ CD에서 커널 소스를 복사한다.

```
# cd /work/achro5250  
# cp -a /media/[Achro-Disc]/kernel/kernel_140806.tar.gz .  
# tar zxvf kernel_140806.tar.gz
```

### ▶ 커널 소스 디렉터리로 이동

```
# cd /work/achro5250/kernel
```

### ▶ 커널 소스 확인

```
# ls
```

```
root@ubuntu:/work/achro5250/kernel# ls  
arch          fk.sh         lib            README         usr  
block         fs            MAINTAINERS    REPORTING-BUGS virt  
COPYING       include       Makefile       samples        vmlinux  
CREDITS       init          mm             scripts        vmlinux.o  
crypto        ipc           modules.builtin security  
Documentation Kbuild       modules.order  sound  
drivers       Kconfig      Module.symvers System.map  
firmware     kernel       net            tools  
root@ubuntu:/work/achro5250/kernel#
```

# 리눅스 커널 컴파일

## ▶ 커널 컴파일

### ▶ 컴파일을 위한 환경변수 설정

```
# make distclean
```

### ▶ 커널 기본 설정 적용 (기본 설정에 관련된 설정 파일은, arch/arm/config 디렉터리에 위치)

```
# make achro5250_defconfig
```

```
root@ubuntu:/work/achro5250/kernel# ls arch/arm/configs/achro5250_defconfig
arch/arm/configs/achro5250_defconfig
root@ubuntu:/work/achro5250/kernel# make distclean
root@ubuntu:/work/achro5250/kernel# make achro5250_defconfig
HOSTCC  scripts/basic/fixdep
HOSTCC  scripts/kconfig/conf.o
SHIPPED scripts/kconfig/zconf.tab.c
SHIPPED scripts/kconfig/lex.zconf.c
SHIPPED scripts/kconfig/zconf.hash.c
HOSTCC  scripts/kconfig/zconf.tab.o
HOSTLD  scripts/kconfig/conf
#
# configuration written to .config
#
root@ubuntu:/work/achro5250/kernel#
```



# 리눅스 커널 컴파일

## ▶ 커널 컴파일

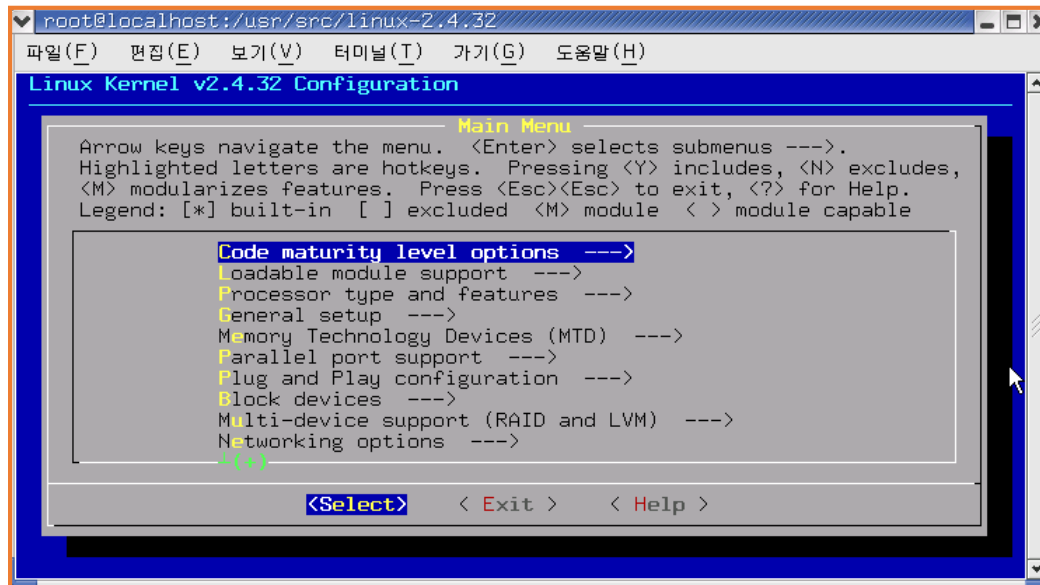
- ▶ 컴파일 하여 생성된 커널 이미지는 arch/arm/boot 디렉터리에 위치 (십여분 소요)

```
# make
```

## ▶ 커널 옵션 변경

- ▶ 커널을 컴파일 하기 이전에 커널에서 지원하는 드라이버나 속성을 변경하려면 옵션을 선택

```
# make menuconfig
```



# 리눅스 커널 컴파일

## ▶ 커널 기록을 위한 실습 툴 환경 설정

### ▶ 실습장비와 USB-OTG 케이블을 연결한다

▷ 커넥터는 휴대 모듈의 옆면에 있음

### ▶ 실습장비의 시리얼(Console) 포트와 PC를 USB-Serial 케이블 연결

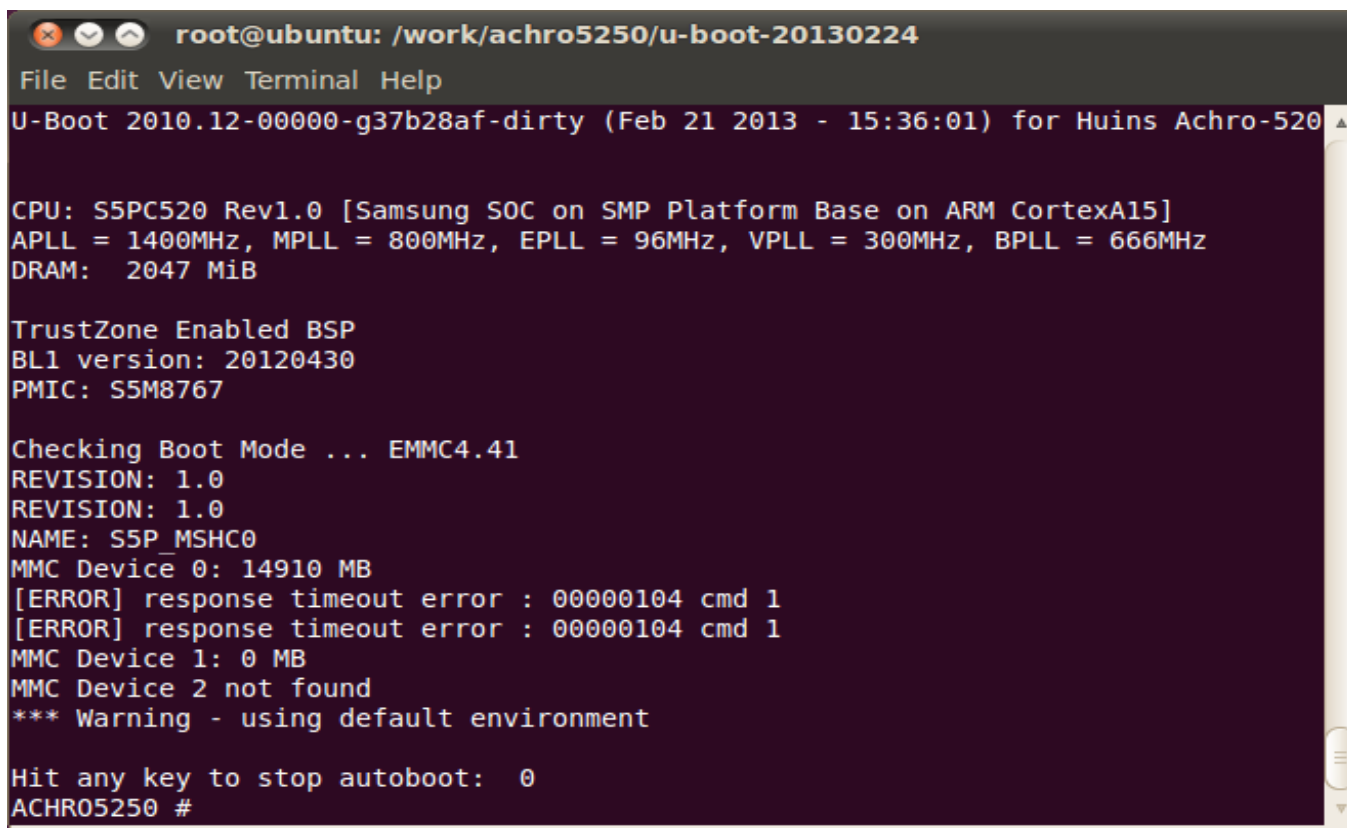


# 리눅스 커널 컴파일

## ▶ 커널 기록

- ▶ 현재 사용중인 터미널(리눅스 호스트용) 이외에 추가로 또다른 터미널(Ctrl+Alt+t)을 열고 여기에  
서 미니컴 실행 (타겟 보드용) - 자동 부팅 전에 엔터키 입력(u-boot 명령창 진입)

```
# minicom
```



```
root@ubuntu: /work/achro5250/u-boot-20130224
File Edit View Terminal Help
U-Boot 2010.12-00000-g37b28af-dirty (Feb 21 2013 - 15:36:01) for Huins Achro-520

CPU: S5PC520 Rev1.0 [Samsung SOC on SMP Platform Base on ARM CortexA15]
APLL = 1400MHz, MPLL = 800MHz, EPLL = 96MHz, VPLL = 300MHz, BPLL = 666MHz
DRAM: 2047 MiB

TrustZone Enabled BSP
BL1 version: 20120430
PMIC: S5M8767

Checking Boot Mode ... EMMC4.41
REVISION: 1.0
REVISION: 1.0
NAME: S5P_MSHC0
MMC Device 0: 14910 MB
[ERROR] response timeout error : 00000104 cmd 1
[ERROR] response timeout error : 00000104 cmd 1
MMC Device 1: 0 MB
MMC Device 2 not found
*** Warning - using default environment

Hit any key to stop autoboot: 0
ACHRO5250 #
```

# 리눅스 커널 컴파일

## ▶ 커널 기록

### ▶ 커널 이미지 전송 (타겟보드의 RAM으로 이미지 다운로드)

#### ▶ Achro5250(타겟 보드)에서 dnw를 이용 다운로드를 할 준비를 함

```
# dnw 0x40000000
```

#### ▶ 호스트의 커널 소스 디렉터리로 이동해서 빌드 된 zImage를 전송 (i는 대문자임)

```
# cd /work/achro5250/kernel  
# smdk-usbdl -f arch/arm/boot/zImage
```

```
root@ubuntu: ~  
File Edit View Terminal Help  
  
Board: achro4210  
POP type: POP_B  
DRAM: 1 GB  
NAND: 1024 MB  
MMC: 1910 MB  
In: serial  
Out: serial  
Err: serial  
  
Checking Boot Mode ... SDMMC  
Hit any key to stop autoboot: 0  
ACHRO4210 # dnw 0x40000000  
OTG cable Connected!  
Now, Waiting for DNW to transmit data  
Download Done!! Download Address: 0x40000000, Download Filesize:0x36e268  
Checksum is being calculated....  
Checksum O.K.  
ACHRO4210 #
```

```
root@ubuntu: /work/achro4210/kernel  
File Edit View Terminal Help  
  
CC drivers/net/wireless/bcm4329/bcm4329.mod.o  
LD [M] drivers/net/wireless/bcm4329/bcm4329.ko  
CC drivers/scsi/scsi_wait_scan.mod.o  
LD [M] drivers/scsi/scsi_wait_scan.ko  
CC drivers/video/backlight/lcd.mod.o  
LD [M] drivers/video/backlight/lcd.ko  
root@ubuntu: /work/achro4210/kernel# ls  
arch          drivers      Kbuild      mm           REPORTING-BUGS  tools  
block         firmware    kernel      modules.builtin  samples         usr  
COPYING       fs          lib         modules.order   scripts         virt  
CREDITS       include     MAINTAINERS Module.symvers  security        vmlinux  
crypto        init        Makefile    net            sound           vmlinux.o  
Documentation ipc          mkcscope.sh README        System.map  
root@ubuntu: /work/achro4210/kernel# smdk-usbdl -f arch/arm/boot/zImage  
SMDK42XX,S3C64XX USB Download Tool  
Version 0.20 (c) 2004,2005,2006 Ben Dooks <ben-linux@fluff.org>  
  
S3C64XX Detected!  
=> found device: bus 001, dev 008  
=> loaded 3596904 bytes from arch/arm/boot/zImage  
=> Downloading 3596914 bytes to 0x30000000  
=> Data checksum 1cb3  
=> usb bulk write() returned 3596914  
root@ubuntu: /work/achro4210/kernel#
```

# 리눅스 커널 컴파일

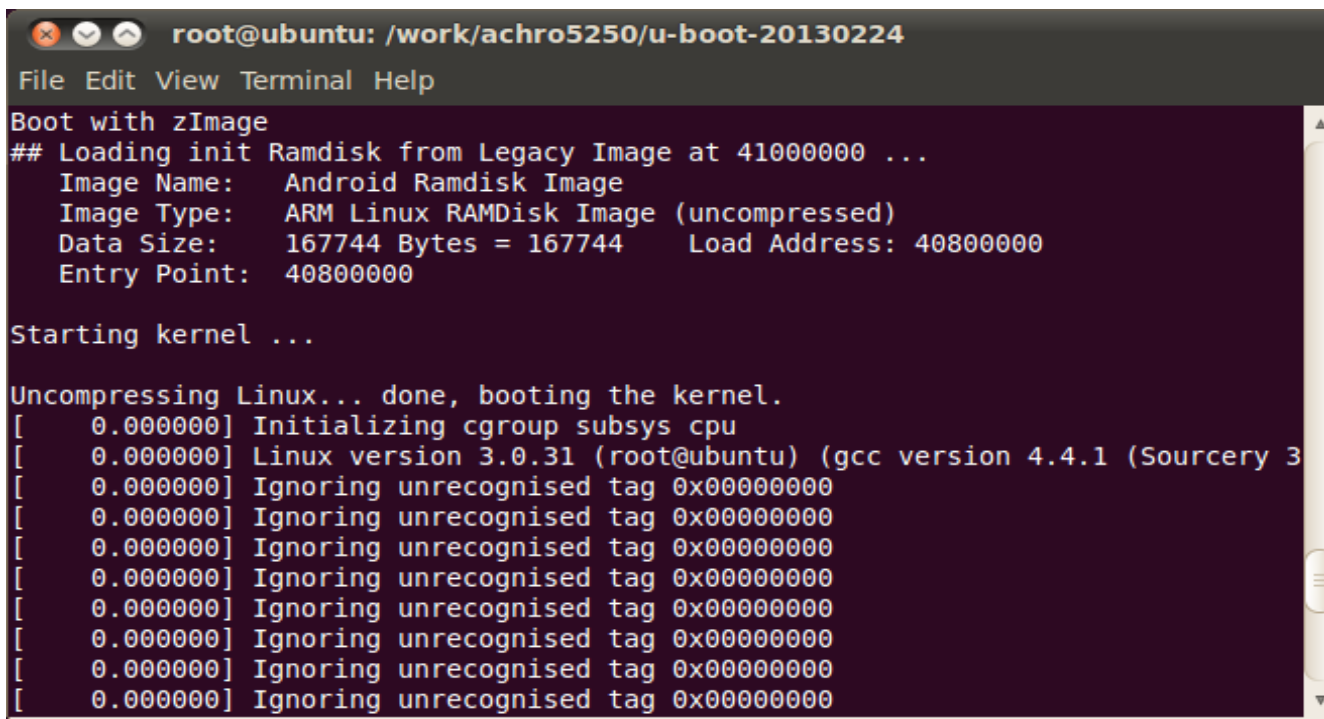
## ▶ 커널 기록

- ▶ Achro5250에서 이미지 기록 (타겟보드 RAM → 마이크로 SD 카드)

```
achro5250 # movi write kernel S5P_MSHC2 0x40000000
```

- ▶ Achro5250에서 리부팅 → 아직 파일시스템이 없어서 커널 패닉이 발생

```
# reset
```



```
root@ubuntu: /work/achro5250/u-boot-20130224
File Edit View Terminal Help
Boot with zImage
## Loading init Ramdisk from Legacy Image at 41000000 ...
Image Name:      Android Ramdisk Image
Image Type:      ARM Linux RAMDisk Image (uncompressed)
Data Size:       167744 Bytes = 167744      Load Address: 40800000
Entry Point:     40800000

Starting kernel ...

Uncompressing Linux... done, booting the kernel.
[ 0.000000] Initializing cgroup subsys cpu
[ 0.000000] Linux version 3.0.31 (root@ubuntu) (gcc version 4.4.1 (Sourcery 3
[ 0.000000] Ignoring unrecognised tag 0x00000000
[ 0.000000] Ignoring unrecognised tag 0x00000000
[ 0.000000] Ignoring unrecognised tag 0x00000000
[ 0.000000] Ignoring unrecognised tag 0x00000000
[ 0.000000] Ignoring unrecognised tag 0x00000000
[ 0.000000] Ignoring unrecognised tag 0x00000000
[ 0.000000] Ignoring unrecognised tag 0x00000000
[ 0.000000] Ignoring unrecognised tag 0x00000000
[ 0.000000] Ignoring unrecognised tag 0x00000000
```

# 리눅스 파일 시스템

## ▶ 파일 시스템의 개요

### ▶ 파일

- ▶ 저장 매체에 보관된 데이터 집합의 추상적 개념
- ▶ 용도에 따라 다른 파일시스템을 사용하는 것이 효율적
  - ▶ FAT(File Allocation Table): MS-DOS와 윈도우즈 95
  - ▶ NTFS(NT File System): 윈도우즈 XP
  - ▶ EXT2, EXT3: 리눅스
  - ▶ ISO 파일시스템: CD-ROM
  - ▶ NFS: 네트워크
  - ▶ EXT3, JFFS2: 저널링 파일시스템
- ▶ 루트 파일시스템으로는 램디스크(RAMDISK), JFFS2, NFS 등을 주로 사용

# 리눅스 파일 시스템

## ▶ 파일 시스템의 개요

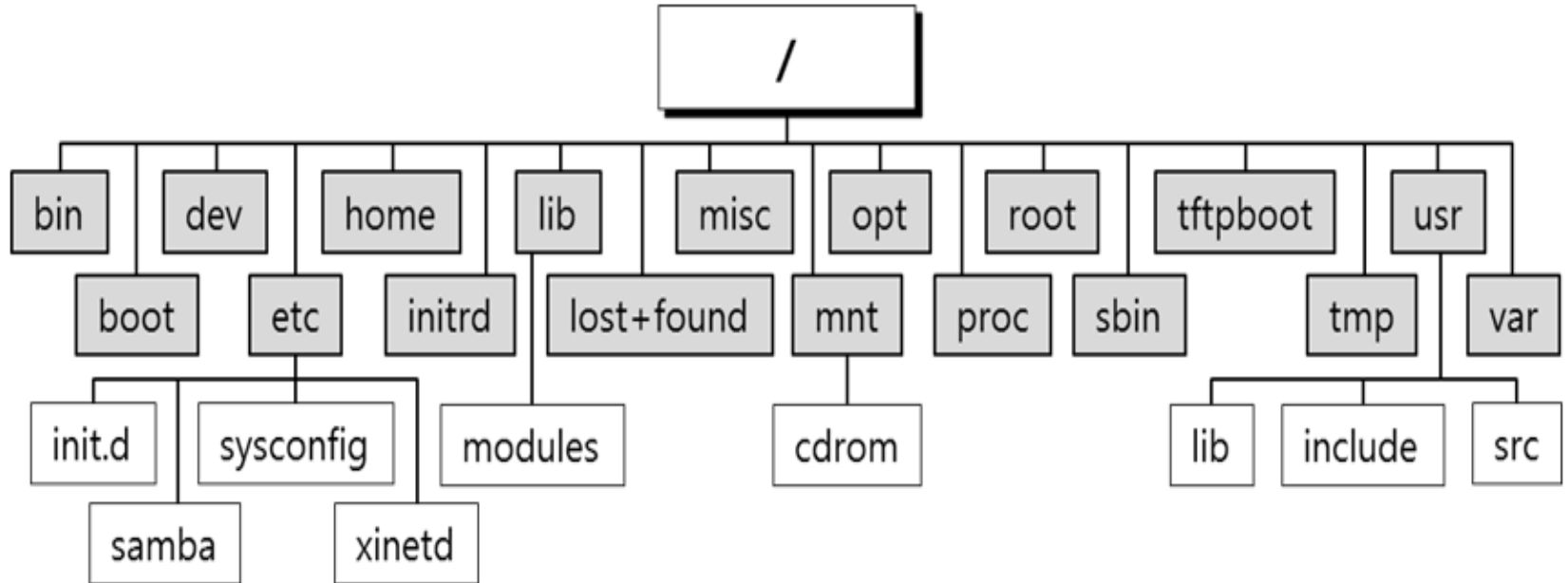
### ▶ 루트 파일 시스템

- ▷ “/”에 마운트되는 파일 시스템
- ▷ 리눅스에서 루트파일 시스템은 커널의 기본적인 운용 환경을 구성
- ▷ 이후 리눅스 시스템에서 사용하는 유틸리티, 라이브러리 등의 추가적인 요소들을 제공

# 리눅스 파일 시스템

## ▶ 리눅스 파일 시스템의 구성

### ▶ 리눅스 파일 시스템의 개략도





# 리눅스 파일 시스템

## ▶ 실제 Achro-5250의 파일 시스템

```
root@ubuntu: ~
파일(F) 편집(E) 보기(V) 터미널(T) 도움말(H)
[ root@Achro-5250 /]# ls -al
total 68
drwxr-xr-x  18 root    root    4096 Mar  6  2013 ./
drwxr-xr-x  18 root    root    4096 Mar  6  2013 ../
drwxrwxr-x   2 root    root    4096 Mar  6  2013 bin/
drwxr-xr-x   9 root    root    4500 Jan  1  00:00 dev/
drwxr-xr-x   8 root    root    4096 Jan  1  00:00 etc/
drwxrwxr-x   4 root    root    4096 Mar  6  2013 home/
drwxrwxr-x   2 root    root    4096 Mar  6  2013 lib/
lrwxrwxrwx   1 root    root      11 Mar  6  2013 linuxrc -> bin/busybox*
drwx-----  2 root    root   16384 Mar  6  2013 lost+found/
drwxrwxr-x   2 root    root    4096 Mar  6  2013 media/
drwxrwxr-x   2 root    root    4096 Mar  6  2013 mnt/
drwxrwxr-x   2 root    root    4096 Mar  6  2013 opt/
dr-xr-xr-x  68 root    root      0 Jan  1  1970 proc/
drwxrwxr-x   2 root    root    4096 Mar  6  2013 root/
lrwxrwxrwx   1 root    root      3 Aug 31  2012 run -> tmp/
drwxrwxr-x   2 root    root    4096 Mar  6  2013/sbin/
drwxr-xr-x  12 root    root      0 Jan  1  00:00 sys/
drwxrwxrwt   2 root    root    100 Jan  1  00:00 tmp/
drwxrwxr-x  10 root    root    4096 Mar  6  2013 usr/
drwxrwxr-x   3 root    root    4096 Aug 31  2012 var/
[ root@Achro-5250 /]#
```

CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.4 | VT102 | Offline

# 리눅스 파일 시스템

## ▶ 리눅스 파일 시스템의 디렉토리

- ▶ /
  - ▷ 리눅스의 최상위 디렉터리
- ▶ /root
  - ▷ 슈퍼 유저의 홈 디렉터리
- ▶ /bin
  - ▷ 시스템을 사용하기 위한 기본적인 명령어들이 위치
- ▶ /boot
  - ▷ 리눅스의 부팅에 필요한 파일들이 들어있음
- ▶ /dev
  - ▷ 리눅스에서 사용되는 모든 디바이스들의 연결에 필요한 노드 등이 위치. 여기 있는 노드들은 `mkmod`로 생성
- ▶ /etc
  - ▷ 리눅스 시스템에서 시스템 환경 설정파일과 응용 프로그램 등에서 사용되는 환경파일이 위치
- ▶ /usr
  - ▷ 기본적으로 시스템 혹은 사용자가 이용하는 프로그램, 유틸리티 등이 설치
- ▶ /lib
  - ▷ 시스템에서 사용되는 라이브러리 등이 위치

# 리눅스 파일 시스템

## ▶ 리눅스 파일 시스템의 디렉토리

### ▶ /mnt

- ▷ 외부 장치들을 시스템에서 연결하기 위한 디렉터리

### ▶ /opt

- ▷ 리눅스 배포본에 포함되지 않고 별도로 설치되는 패키지를 설치하는 디렉터리

### ▶ /proc

- ▷ 프로세스와 관련된 시스템 정보를 제공하기 위해 사용
- ▷ 메모리 정보 혹은 커널 관련 정보들이 위치하며 cat 명령으로 해당 상태를 확인

### ▶ /tmp

- ▷ 프로세스 생성과정 혹은 특정 응용 설치 시 임시로 사용되는 디렉터리

### ▶ /sbin

- ▷ 시스템 운영에 필요한 관리자용 명령 혹은 프로그램이 위치

### ▶ /var

- ▷ 시스템 운영 시 발생하는 로그들이 저장

### ▶ /home

- ▷ /home 디렉토리에는 사용자의 개인 디렉터리가 만들어짐
- ▷ 보통 만드는 규칙 등은 /etc/skel 디렉터리를 따름

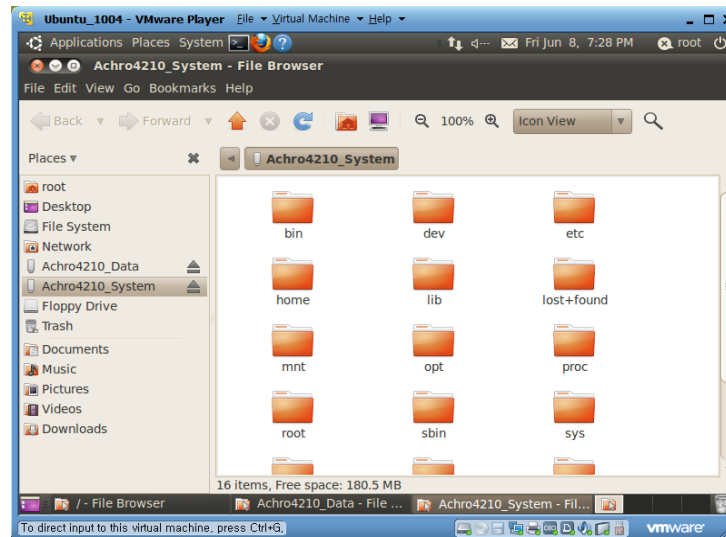
# 리눅스 파일 시스템 설치

## ▶ 파일 시스템 설치

- ▶ System MicroSD를 Achro-5250으로부터 분리 (반드시 전원을 끄고 진행해야 함)
- ▶ USB타입 MicroSD 젠더를 이용하여 호스트에 연결한다.



- ▶ 리눅스에서 마운트가 되면 다음과 같이 마운트 된 디렉터리가 표시



# 리눅스 파일 시스템 설치

## ▶ 파일 시스템 설치

### ▶ 리눅스 파일 시스템 준비

```
# cd /work/achro5250  
# cp -a /[Achro_Disc]/filesystem/linux_filesystem/linuxfs_20130312.tar.gz .  
# tar zxvf linux_20130312.tar.gz
```

### ▶ 파일시스템 복사

- ▶ 마운트가 확인 되면 터미널을 실행하여 만들어 둔 파일시스템을 MicroSD로 복사한다.  
그전에 이전에 파일 시스템이 들어 있다면 삭제해주고 복사한다.

```
# cd /media/Achro5250_System  
# rm -rf *  
# cd /work/achro5250/linuxfs  
# cp -a * /media/Achro5250_System  
# sync
```

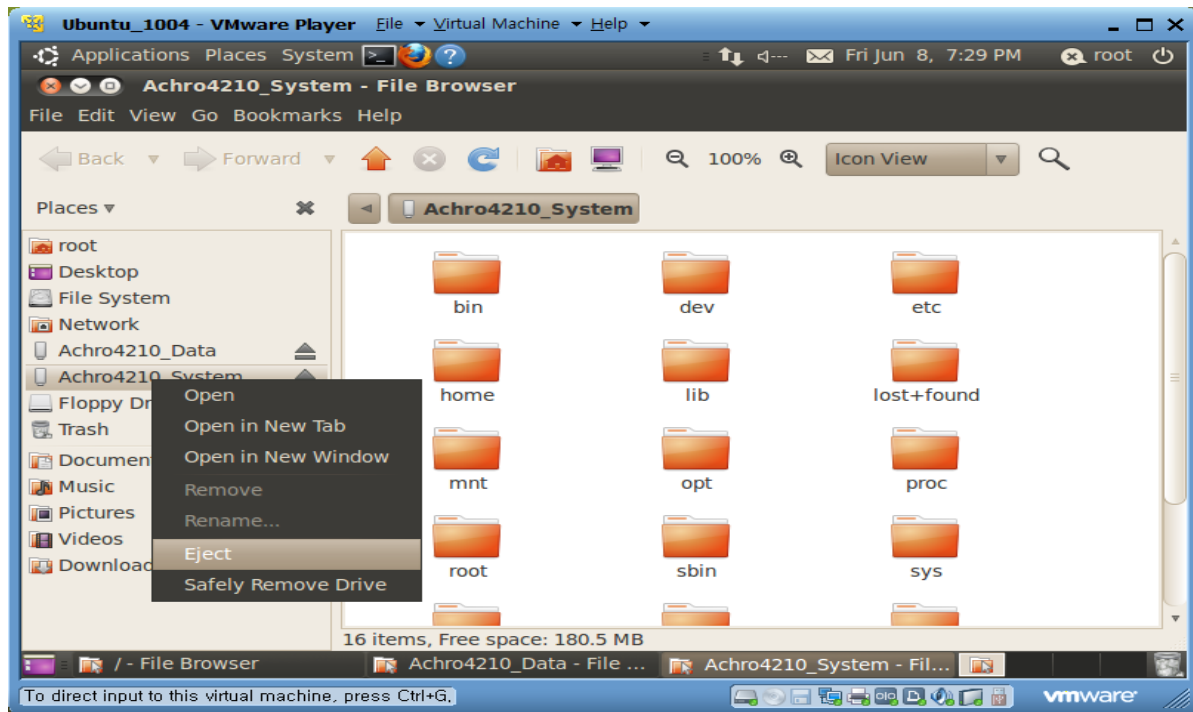
# 리눅스 파일 시스템 설치

## ▶ 파일 시스템 설치

- ▶ 복사가 완료되면 마운트를 해제하고 PC에서 MicroSD를 분리하여 Achro-5250의 System 소켓에 연결

```
# umount /media/achro5250_System  
# umount /media/achro5250_Data
```

- ▶ 다음과 같이 메뉴를 통해 해당 마운트된 영역에서 마우스 우측클릭 후 Eject로 분리가 가능



---

**Q & A**

---