

Smart Phone Hacking!

정구홍@BoB

2013-08-20

강의 일정

- 1차 : Inside of Smart Phone
 - Composition of Smart Phone
 - Building Linux & Android
- 2차 : Remote Attack
 - Remote Attack Vectors
 - Case study : Flash Vulnerability
- 3차 : Local Attack
 - Local Attack Vectors
 - Case study : Device Driver Vulnerability

오늘 배워볼 것들

- 스마트폰의 구성 이해하기
 - 하드웨어 구성
 - 소프트웨어 구성
- 리눅스 운영체제 빌드해보기
 - 크로스 컴파일
 - 커널 컴파일
 - 루트 파일 시스템 컴파일
 - 부트 로더 이해하기
- 안드로이드 운영체제 빌드 (구경)해보기

진행하면서 동시에..

- 필요한 프로그램들을 설치해주세요
 - apt-get update
 - apt-get install openssh-server
 - apt-get install xinit
 - apt-get install qemu
 - apt-get install libncurses5-dev
- VM 비밀번호
 - notroot / thoughtpolice
 - sudo -l (thoughtpolice)

연구용 스마트폰 구매하기

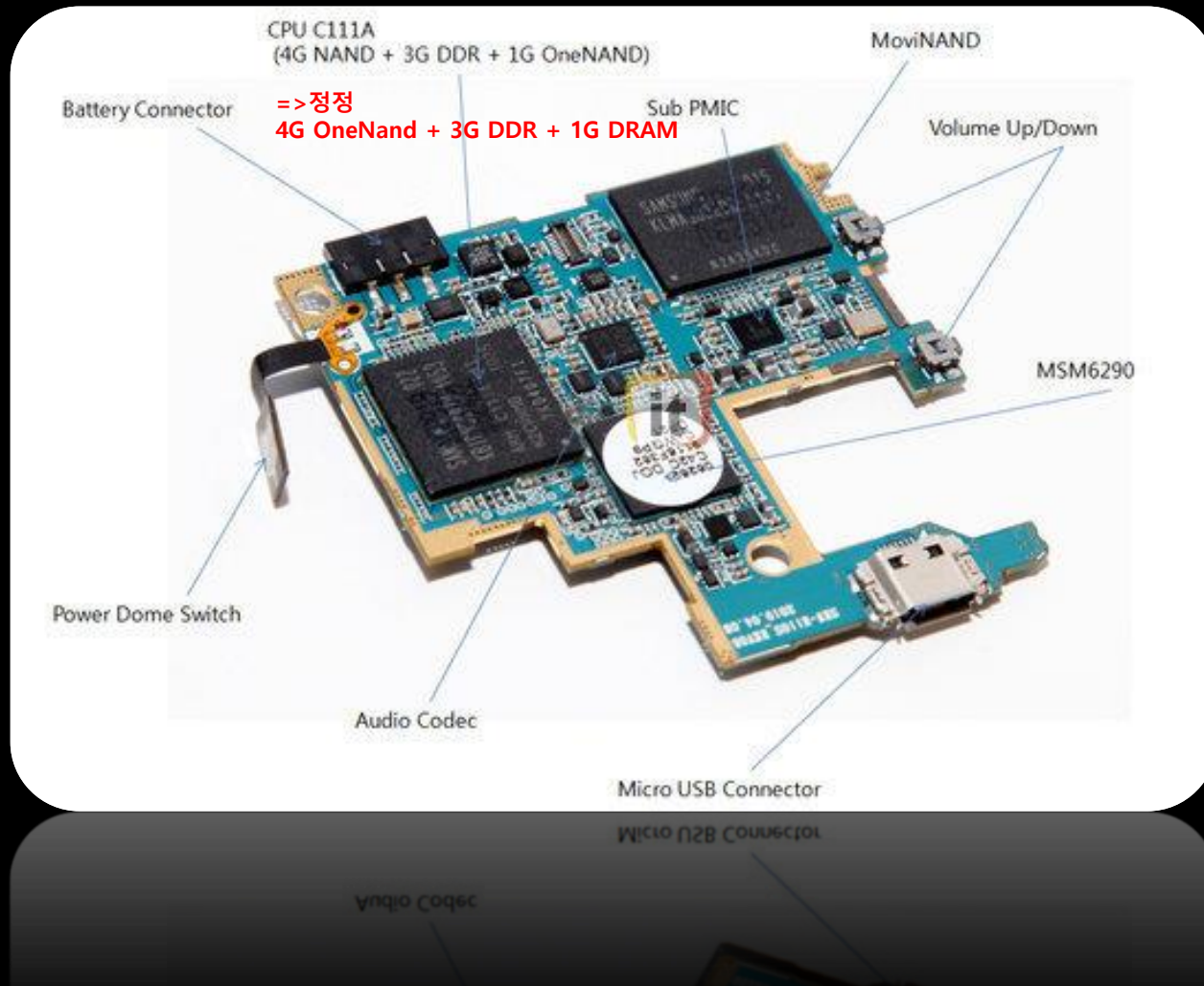
- 다음의 글들을 참고해 주세요

- http://www.hackersschool.org/Sub_Html/HS_Posting/?uid=17

- http://www.hackersschool.org/Sub_Html/HS_Posting/?uid=18

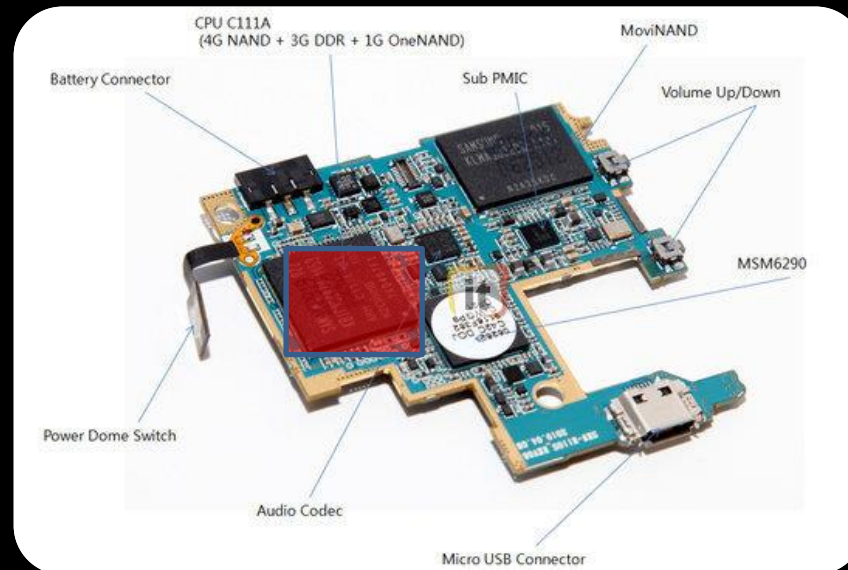
스마트폰 하드웨어의 구성

(갤럭시S 예제)



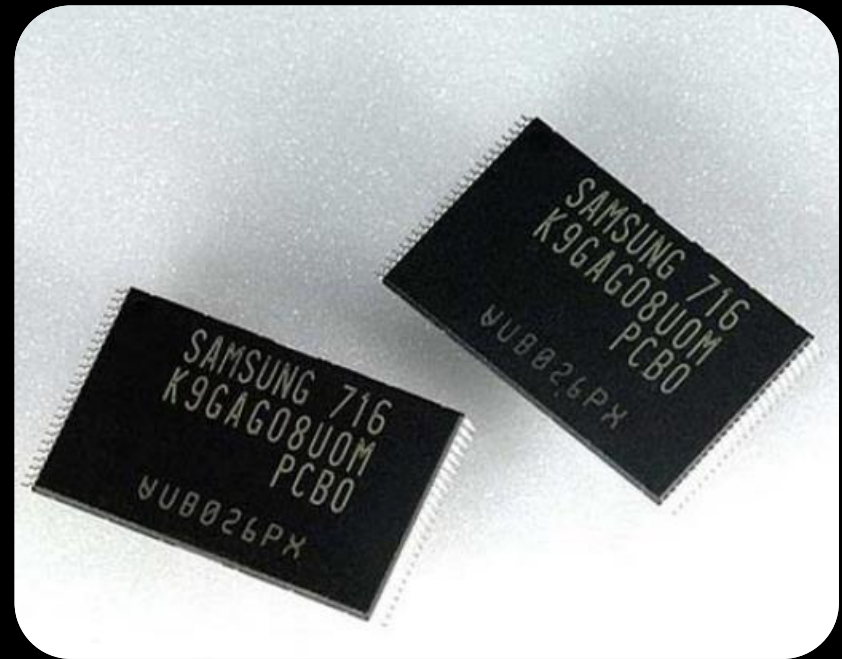
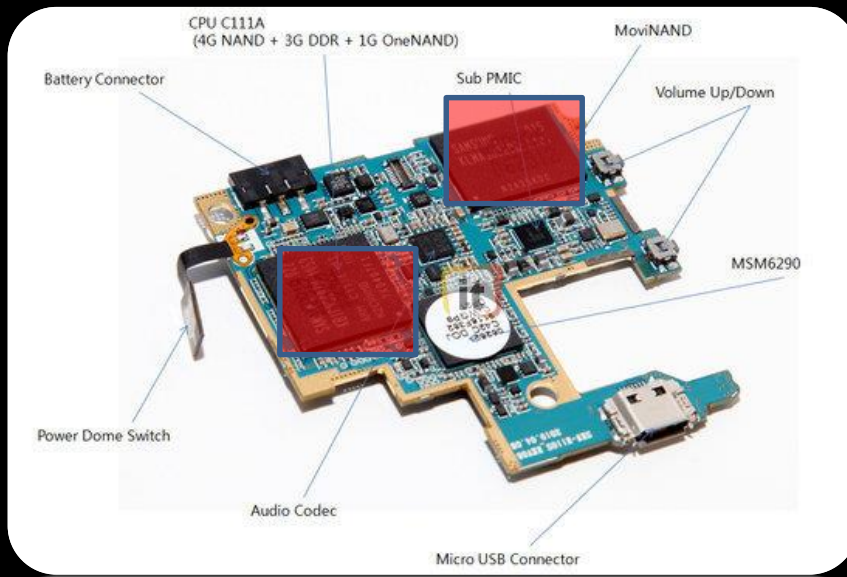
CPU

- S5PC111
 - 삼성 제작
 - Core : Cortex A8 (ARM)
 - CPU + OneNand Flash + DDR RAM + DRAM



Flash Memory

- 비휘발성 메모리 장치
- 커널, 운영체제 파일들, 사용자 데이터 등을 저장



NOR vs NAND

- NOR
 - 각 메모리 셀이 병렬 형태로 이루어짐
 - Read 속도가 빠름
 - Write 속도는 느림
 - 코드 영역으로 적합 (ex. 펌웨어)
- NAND
 - 셀이 직렬 형태로 이루어짐
 - Read는 느리고, Write는 빠름
 - 데이터 영역으로 적합 (ex. 이동식 장치)

NOR vs NAND

- NOR (Code)
 - 각 메모리 셀이 병렬 형태로 이루어짐
 - Read 속도가 빠름
 - Write 속도는 느림
 - 코드 영역으로 적합 (ex. 펌웨어)
- NAND (Data)
 - 셀이 직렬 형태로 이루어짐
 - Read는 느리고, Write는 빠름
 - 데이터 영역으로 적합 (ex. 이동식 장치)

OneNand란?

- 삼성전자 개발
- NOR Flash와 NAND Flash의 장점을 합함
 - 읽기/쓰기 모두 빠름
 - 퓨전 메모리 형태
- 즉, 코드 실행까지 가능한 NAND
- 최대 10배의 속도

MoviNand?

- 삼성전자 개발
- 모바일폰 전용 대용량 Nand Flash
- Nand Flash + Controller => 원칩화
- OneNand보다 느림

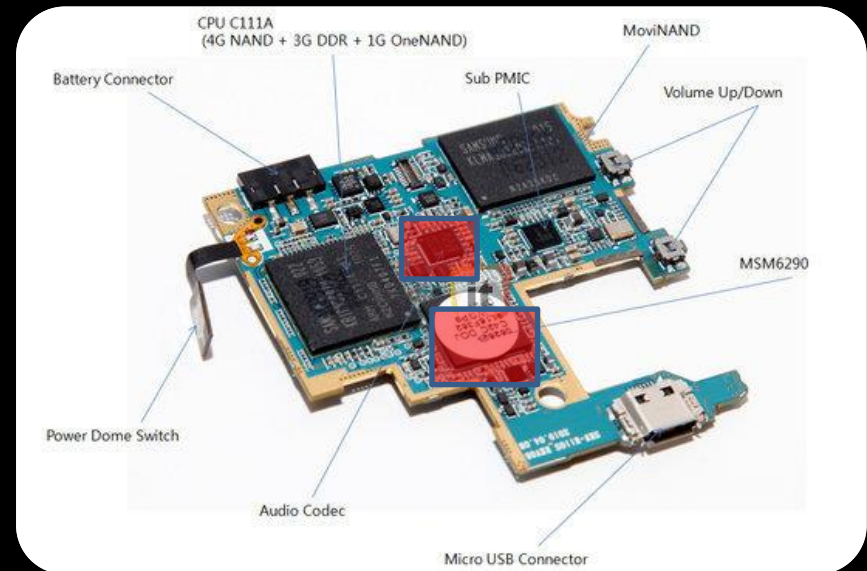


갤럭시S의 저장장치 구성

- oneNand
 - 여러 개의 파티션으로 분할
 - /system/ (300메가)
 - /dbdata/ (100메가)
 - /cache/ (30메가)
- moviNand
 - 2개의 파티션으로 분할
 - 내장 SD카드(14기가)
 - /data/ (2기가)
 - /dev/block/mmcXXX

그 외의 칩들

- MSM6290
 - 퀄컴 제작
 - 통신 전용 칩셋
 - WCDMA, HSDPA
- 오디오코덱
 - 오디오 데이터 처리



갤럭시S vs 아이폰4

이지이 IT참고



제품명	삼성 GALAXY S	iPhone 4
프로세서	S5PC111 (Cortex A8 1GHz)	Apple A4(Cortex A8 1GHz)
내장 GPU	PowerVR SGX540	PowerVR SGX535
운영체제	안드로이드 2.1 + 햅틱UI	iOS4
메모리(RAM/ROM)	512RAM/8~16GB ROM	512RAM/16~32GB ROM
외장 메모리	micro SD 확장가능	확장 불가
디스플레이	4인치 Super AM OLED(800x480)	3.5인치 레티나 LCD(960x640) IPS패널
터치방식	정전식	정전식
멀티터치	○	○
카메라	500만화소(AF) 720p/30fps 동영상 촬영	500만화소(AF, 플래쉬) 720p/30fps 동영상 촬영
영상통화	○	○ (Wi-Fi제한)
블루투스	3.0	2.1
Wi-Fi	802.11b/g/n	802.11b/g/n

비슷비슷..

스마트폰의 소프트웨어 이해하기

- ARM 바이너리 만들기 실습
- ARM 커널 만들기 실습
- 부트로더,
- 리눅스 커널
- 안드로이드 운영체제

ARM 바이너리 만들기 실습

Cross Compile란?

- 다른 아키텍처의 코드를 생성하는 컴파일
- 예
 - x86에서 x86코드 컴파일 => Not Cross Compiler
 - x86에서 ARM코드 컴파일 => Cross Compiler!
 - ARM에서 x86코드 컴파일 => Cross Compiler!
- Cross Compiler 설치 필요

Cross Compile 환경 구축하기

- OS
 - 우분투 13.04 (다른 버전도 상관없음)
 - <http://sourceforge.net/projects/thoughtpolicevm/files/Ubuntu/ubuntu-server-13.04/ubuntu-server-13.04-i386.zip/download>
- Cross Compiler 다운로드
 - CodeSourcery

Cross Compiler 설치

- CodeSourcery Cross Compiler 설치
 - http://sourcery.mentor.com/public/gnu_toolchain/arm-none-linux-gnueabi/
 - 최신 버전으로 down!
 - arm-2013.05-24-arm-none-linux-gnueabi.bin
 - 설치 now
 - `chmod +x "파일명"`
 - `./"파일명"`
 - 기본셸 변경 후 다시 실행 (중요)

```
root@ubuntu:~/CrossCompiler# rm -rf /bin/sh
root@ubuntu:~/CrossCompiler# ln -s /bin/bash /bin/sh
```

Cross Compiler 설치

- CodeSourcery Cross Compiler 설치
 - 엔터엔터 와이와이~

```
Install Folder:
  /root/CodeSourcery/Sourcery_CodeBench_Lite_for_ARM_GNU_Linux

Link Folder:
  /root/Sourcery_CodeBench_Lite_for_ARM_GNU_Linux

Disk Space Information (for Installation Target):
  Required: 387,119,237 bytes
  Available: 17,149,759,488 bytes

PRESS <ENTER> TO CONTINUE:

=====
Ready To Install
=====

InstallAnywhere is now ready to install Sourcery CodeBench Lite for ARM
GNU/Linux onto your system at the following location:

  /root/CodeSourcery/Sourcery_CodeBench_Lite_for_ARM_GNU_Linux

PRESS <ENTER> TO INSTALL:

=====
Installing...
=====

[=====|=====|=====|=====]
[
```

Cross Compiler 설치

- 대표적인 ARM용 Cross Compiler들
 - CodeSourcery에서 배포
 - http://sourcery.mentor.com/public/gnu_toolchain/arm-none-linux-gnueabi/
 - Android에서 배포
 - <http://developer.android.com/tools/sdk/ndk/index.html>
 - uCLibc에서 배포
 - <http://www.uclibc.org/downloads/binaries/>
 - Kegel에서 배포
 - <http://kegel.com/crosstool/>

설치 완료

```
root@ubuntu:~# cd /root/CodeSourcery/Sourcery_CodeBench_Lite_for_ARM_GNU_Linux
root@ubuntu:~/CodeSourcery/Sourcery_CodeBench_Lite_for_ARM_GNU_Linux# cd bin
root@ubuntu:~/CodeSourcery/Sourcery_CodeBench_Lite_for_ARM_GNU_Linux/bin#
root@ubuntu:~/CodeSourcery/Sourcery_CodeBench_Lite_for_ARM_GNU_Linux/bin#
root@ubuntu:~/CodeSourcery/Sourcery_CodeBench_Lite_for_ARM_GNU_Linux/bin# ./arm-
none-linux-gnueabi-gcc
arm-none-linux-gnueabi-gcc: fatal error: no input files
compilation terminated.
root@ubuntu:~/CodeSourcery/Sourcery_CodeBench_Lite_for_ARM_GNU_Linux/bin#
root@ubuntu:~/CodeSourcery/Sourcery_CodeBench_Lite_for_ARM_GNU_Linux/bin#
```

helloworld.c를 ARM용으로~

```
# cd /root/CodeSourcery/Sourcery_CodeBench_Lite_for_ARM_GNU_Linux
```

```
# vi hello.c
```

```
void main()
```

```
{
```

```
    printf("hello world\n");
```

```
}
```

```
#
```

```
# ./arm-none-linux-gnueabi-gcc -o hello hello.c
```

```
hello.c: In function 'main':
```

```
hello.c:3:2: warning: incompatible implicit declaration of built-in function 'printf' [enabled by default]
```

```
#
```

```
# file hello
```

```
hello: ELF 32-bit LSB executable, ARM, version 1 (SYSV), dynamically linked (uses shared libs), for GNU/Linux 2.6.16, not stripped
```

```
#
```

* Static 컴파일하기

```
# ./arm-none-linux-gnueabi-gcc -o hello hello.c -static
```

```
hello.c: In function 'main':
```

```
hello.c:3:2: warning: incompatible implicit declaration of built-in function 'printf' [enabled
```

```
#
```


EABI란?

- ABI(Application Binary Interface)
 - 어플리케이션과 OS 사이 혹은 어플리케이션과 라이브러리 사이에서 사용되는 Low-Level 인터페이스
 - API와의 차이점
 - API : 소스 코드 레벨에서의 관점
 - ABI : 바이너리 레벨에서의 관점
 - 즉, 컴파일 되어 나온 Object 파일들과의 연결 방식

EABI란?

- EABI(Embedded Application Binary Interface)
 - 임베디드 소프트웨어의 자료형, 레지스터 사용 등에 대한 표준 정의
 - 다른 Compiler를 사용해 만들어진 Object 들과의 연동에 필요
- None EABI
 - “어떤 OS에 특화된 EABI가 아니다”라는 뜻
 - 즉, 특정 OS 하나를 위한 바이너리가 아니다
- 출처
 - <http://blog.naver.com/dong880510/140156760753>

정적 컴파일하기

- 필요한 라이브러리가 대상 OS 안에 없을 수 있음
- 있더라도 경로가 달라서 로딩하지 못할 수 있음
- 로딩했더라도 호환되지 않을 수 있음

```
# ./arm-none-linux-gnueabi-gcc -o hello hello.c -static
```

내 폰에 올려서 실행해보잡

- 아이폰(iOS)
- 갤럭시(안드로이드)

리눅스 커널 컴파일하기

- 커널 소스코드 다운로드
 - <https://www.kernel.org>

```
root@ubuntu:~/Linux_Build# xz -d linux-3.10.6.tar.xz
root@ubuntu:~/Linux_Build#
root@ubuntu:~/Linux_Build# ls
linux-3.10.6.tar
root@ubuntu:~/Linux_Build# tar xvf linux-3.10.6.tar
...
```

리눅스 커널 컴파일하기

```
# make ARCH=arm versatile_defconfig  
# make ARCH=arm menuconfig  
# make ARCH=arm CROSS_COMPILE=  
  /root/CodeSourcery/Sourcery_CodeBench_Lite_for_ARM_GNU_Linu  
  x/bin/arm-none-linux-gnueabi- all
```

...

```
# find . -name zImage  
./arch/arm/boot/zImage  
#
```

Config 변경 (menuconfig)

kernel Features -->

Memory split (3G/1G user/kernel split) --->

☐ Preemptible Kernel (EXPERIMENTAL)

☒ Use the ARM EABI to compile the kernel

☒ Allow old ABI binaries to run with this kernel (EXPERIMENTAL)

☐ High Memory Support (EXPERIMENTAL)

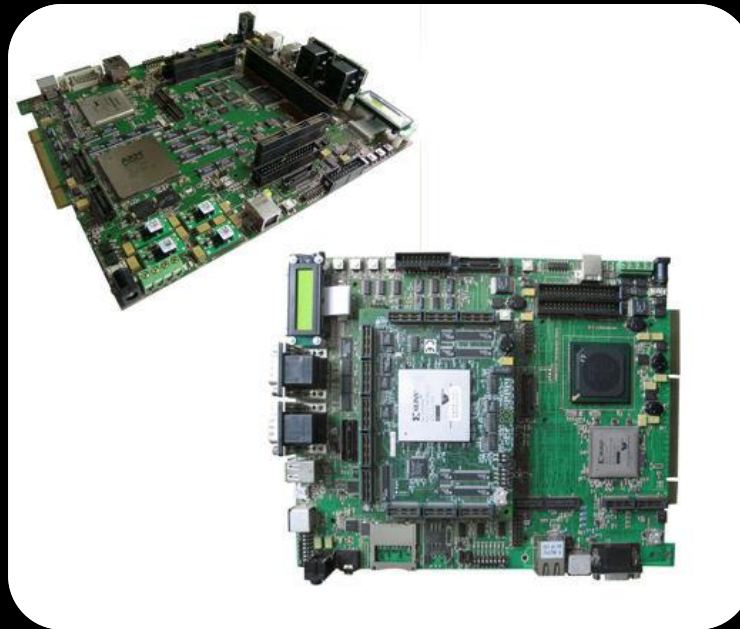
Memory model (Flat Memory) --->

☐ Add LRU list to track non-evictable pages

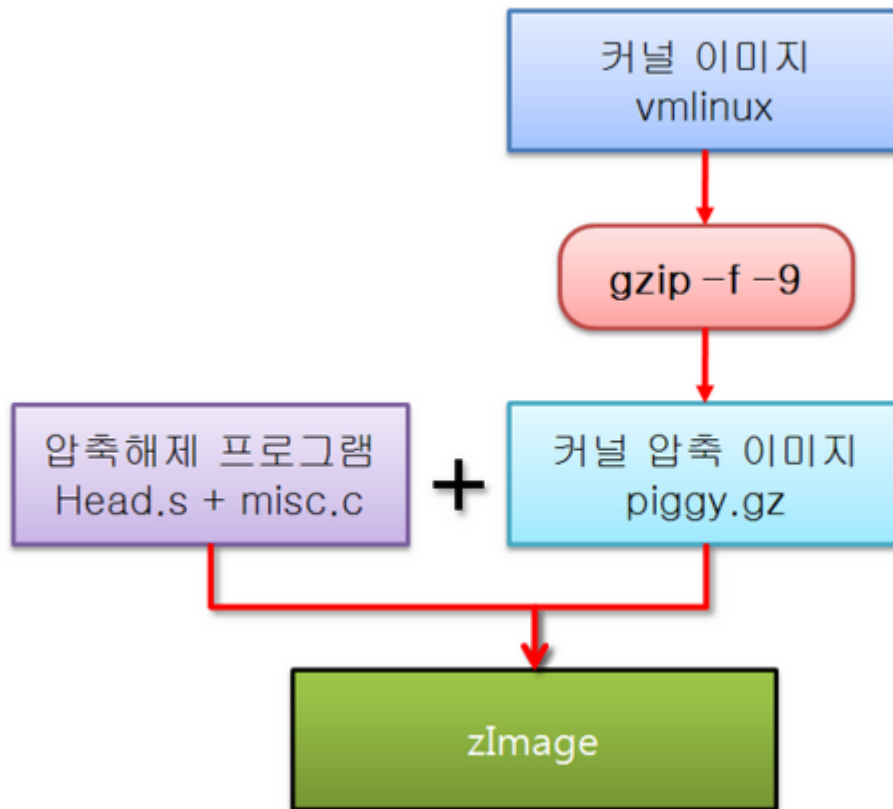
(4096) Low address space to protect from user allocation

Versatile?

- "The Versatile one is the most interesting as it includes a hard disk SCSI controller, an Ethernet card and a graphical display"



zImage의 구조



zImage의 구조

- vmlinux : 실제 커널
- piggy.gz : vmlinux를 압축한 파일
- misc.c : 압축 해제 수행
- head.s : 압축 해제된 코드로 jump

자, 이제 이 커널을 어디서?

- 어디서 실행 할 것인가?
 - 후보1 : 내 스마트폰? -_-
 - 후보2 : 개발용 보드
 - 후보3 : QEMU
 - # apt-get install qemu
 - Qemu가 지원하는 아키텍처들

```
...  
qemu-system qemu-system-arm qemu-system-  
common qemu-system-mips qemu-system-misc  
qemu-system-ppc  
    qemu-system-sparc qemu-system-x86 qemu-user  
qemu-utils seabios sharutils vgabios  
...
```

QEMU가 지원하는 보드 목록

```
root@ubuntu:~/Linux_Build/linux-3.10.6# qemu-system-arm -M xxx
```

Supported machines are:

none	empty machine
beagle	Beagle board (OMAP3530)
beaglexm	Beagle board XM (OMAP3630)
collie	Collie PDA (SA-1110)
nuri	Samsung NURI board (Exynos4210)
smdkc210	Samsung SMDKC210 board (Exynos4210)
connex	Gumstix Connex (PXA255)
verdex	Gumstix Verdex (PXA270)
highbank	Calxeda Highbank (ECX-1000)
integratorcp	ARM Integrator/CP (ARM926EJ-S) (default)
kzm	ARM KZM Emulation Baseboard (ARM1136)
mainstone	Mainstone II (PXA27x)
musicpal	Marvell 88w8618 / MusicPal (ARM926EJ-S)
n800	Nokia N800 tablet aka. RX-34 (OMAP2420)
n810	Nokia N810 tablet aka. RX-44 (OMAP2420)
...	

커널 부팅 성공~

- qemu-system-arm -M versatilepb -m 128M -kernel zImage

```
bio: create slab <bio-0> at 0
Switching to clocksource timer3
NET: Registered protocol family 2
TCP: established hash table entries: 512 (order: 0, 4096 bytes)
TCP: bind hash table entries: 512 (order: -1, 2048 bytes)
TCP: Hash tables configured (established 512 bind 512)
TCP: reno registered
UDP: hash table entries: 256 (order: 0, 4096 bytes)
UDP: Lite hash table entries: 256 (order: 0, 4096 bytes)
NET: Registered protocol family 1
RPC: Registered named UNIX socket transport module.
RPC: Registered udp transport module.
RPC: Registered tcp transport module.
RPC: Registered tcp NFSv4.1 backchannel transport module.
NetWinder Floating Point Emulator V0.97 (double precision)
Installing knfsd (copyright (C) 1996 okir@monad.swb.de).
ifconfig: version 2.2. (NAND) T-2001-2006 Red Hat, Inc.
ROMFS MTD (C) 2007 Red Hat, Inc.
msgmni has been set to 55
Block layer SCSI generic (bsg) driver version 0.4 loaded (major 254)
io scheduler noop registered
io scheduler deadline registered
io scheduler cfq registered (default)
clkdev: plllx dev:20: PLL10 rev0 at 0x10120000
clkdev: plllx dev:20: Versatile hardware, UGA display
Console: switching to colour frame buffer device 80x60
brd: module loaded
physmap platform flash device: 04000000 at 34000000
physmap-flash.0: Found 1 x32 devices at 0x0 in 32-bit bank. Manufacturer ID 0x00
0000 Chip ID 0x000000
Intel/Sharp Extended Query Table at 0x0031
Using buffer write method
smc91c11x.c: v1.1. Sep 22 2004 by Nicolas Pitre <nico@fluxnic.net>
eth0: SMC91C11x (rev 1) at c29c8000 IRQ 57 [nowait]
eth0: Ethernet addr: 52:54:00:12:34:56
mousedev: PS/2 mouse device common for all mice
TCP: cubic registered
NET: Registered protocol family 17
UFP support v0.3: implementor 41 architecture 1 part 18 variant 9 rev 0
input: BT Raw Set 2 keyboard as /devices/fpga:06/serial0/input0
input: ImExPS/2 Generic Explorer Mouse as /devices/fpga:07/serial1/input0/input1
UFS: Cannot open root device "1f03" or unknown-block(31,3): error -6
Please append a correct "root=" boot option; here are the available partitions:
1f00 65536 mtdblock0 (driver?)
Kernel panic - not syncing: UFS: Unable to mount root fs on unknown-block(31,3)
CPU: 0 [0] 1 Comm swapper Not tainted 3.10.6 #1
[[c0018b3c]] (unwind_backtrace+0x0/0xf0) from [[c00169a4]] (show_stack+0x10/0x14)
[[c00169a4]] (show_stack+0x10/0x14) from [[c0283c90]] (panic+0x80/0x1d0)
[[c0283c90]] (panic+0x80/0x1d0) from [[c034eeee4]] (mount_block_root+0x1a0/0x258)
[[c034eeee4]] (mount_block_root+0x1a0/0x258) from [[c034f188]] (mount_root+0xf0/0
x118)
[[c034f188]] (mount_root+0xf0/0x118) from [[c034f310]] (prepare_namespace+0x160/
0x1b4)
[[c034f310]] (prepare_namespace+0x160/0x1b4) from [[c034eb6c]] (kernel_init_free
able+0x16c/0x1b0)
[[c034eb6c]] (kernel_init_freeable+0x16c/0x1b0) from [[c0282d58]] (kernel_init+0
x0/0xe4)
[[c0282d58]] (kernel_init+0x0/0xe4) from [[c0013db0]] (ret_from_fork+0x14/0x24)
```

Root File System

- 루트 파일 시스템이란?
 - 커널 부팅 완료 후 만나게 되는 파일들
 - OS 인터페이스
 - Shell
 - X-Windows
 - 기본 프로그램들
 - Login, passwd, ls, id, ps, netstat 등등..
 - 라이브러리들
 - Glibc 등

BusyBox 소개

- 다양한 유틸리티, 프로그램들을 하나로 통합한 패키지 프로그램
- 중복되는 부분을 제거함으로써 용량 최소화
- 스마트폰에 없는 프로그램들 제공
- 임베디드 운영체제에서 많이 사용 됨
- 다운로드
 - <http://busybox.net/downloads/busybox-1.21.1.tar.bz2>

Busybox 컴파일

- make ARCH=arm CROSS_COMPILE=/root/xxx defconfig
- make ARCH=arm CROSS_COMPILE=/root/xxx menuconfig
- 컴파일 전에 옵션 변경
 - Static binary 체크

```
[*] Build BusyBox as a static binary (no shared libs)
[ ] Force NOMMU build
[*] Build with Large File Support (for accessing files > 2 GB)
() Cross Compiler prefix
() Additional CFLAGS
```

- make ARCH=arm CROSS_COMPILE=/root/xxx install

Busybox 컴파일

```
./_install//usr/sbin/rdev -> ../../bin/busybox
./_install//usr/sbin/readahead -> ../../bin/busybox
./_install//usr/sbin/readprofile -> ../../bin/busybox
./_install//usr/sbin/remove-shell -> ../../bin/busybox
./_install//usr/sbin/rtcwake -> ../../bin/busybox
./_install//usr/sbin/sendmail -> ../../bin/busybox
./_install//usr/sbin/setfont -> ../../bin/busybox
./_install//usr/sbin/setlogcons -> ../../bin/busybox
./_install//usr/sbin/svlogd -> ../../bin/busybox
./_install//usr/sbin/telnetd -> ../../bin/busybox
./_install//usr/sbin/tftpd -> ../../bin/busybox
./_install//usr/sbin/ubiattach -> ../../bin/busybox
./_install//usr/sbin/ubidetach -> ../../bin/busybox
./_install//usr/sbin/ubimkvol -> ../../bin/busybox
./_install//usr/sbin/ubirmvol -> ../../bin/busybox
./_install//usr/sbin/ubirsvol -> ../../bin/busybox
./_install//usr/sbin/ubiupdatevol -> ../../bin/busybox
./_install//usr/sbin/udhcpd -> ../../bin/busybox

-----
You will probably need to make your busybox binary
setuid root to ensure all configured applets will
work properly.
-----

root@ubuntu:~/Linux_Build/busybox/busybox-1.21.1#
```

기본 파일시스템 생성

```
# cd _install/
|# mkdir proc sys dev etc etc/init.d
#
# cd etc
# cd init.d
# vi rcS
    #!/bin/sh
    mount -t proc none /proc
    mount -t sysfs none /sys
    /sbin/mdev -s
# chmod +x rcS
# cd ..
# cd ..
# find . | cpio -o --format=newc > ../rootfs.img
    3994 blocks
#
```

OS 부팅

- `qemu-system-arm -M versatilepb -m 128M -kernel zImage -initrd rootfs.img.gz -append "root=/dev/ram rdinit=/sbin/init"`

와우 쉘이 떴다!

```
TCP bind hash table entries: 1024 (order: 0, 4096 bytes)
TCP: Hash tables configured (established 1024 bind 1024)
TCP: reno registered
UDP hash table entries: 256 (order: 0, 4096 bytes)
UDP-Lite hash table entries: 256 (order: 0, 4096 bytes)
NET: Registered protocol family 1
RPC: Registered named UNIX socket transport module.
RPC: Registered udp transport module.
RPC: Registered tcp transport module.
RPC: Registered tcp NFSv4.1 backchannel transport module.
Trying to unpack rootfs image as initramfs...
Freeing initrd memory: 1996K (c4000000 - c41f3000)
NetWinder Floating Point Emulator V0.97 (double precision)
Installing knfsd (copyright (C) 1996 okir@monad.swb.de).
jffs2: version 2.2. (NAND) 2001-2006 Red Hat, Inc.
ROMFS MTD (C) 2007 Red Hat, Inc.
msgmni has been set to 246
Block layer SCSI generic (bsg) driver version 0.4 loaded (major 254)
io scheduler noop registered
io scheduler deadline registered
io scheduler cfq registered (default)
clcd-pll1x dev:20: PLL10 rev0 at 0x10120000
clcd-pll1x dev:20: Versatile hardware, UGA display
Console: switching to colour frame buffer device 80x60
brd: module loaded
physmap-flash device: 04000000 at 34000000
physmap-flash.0: Found 1 x32 devices at 0x0 in 32-bit bank. Manufacturer ID 0x00
0000 Chip ID 0x000000
Intel/Sharp Extended Query Table at 0x0031
Using buffer write method
smc91x.c: v1.1, sep 22 2004 by Nicolas Pitre <nico@fluxnic.net>
eth0: SMC91C11x (rev 1) at c89c8000 IRQ 57 [nowait]
eth0: Ethernet addr: 52:54:00:12:34:56
mousedev: PS/2 mouse device common for all mice
TCP: cubic registered
NET: Registered protocol family 17
VFP support v0.3: implementor 41 architecture 1 part 10 variant 9 rev 0
Freeing unused kernel memory: 112K (c034e000 - c036a000)
input: AT Raw Set 2 keyboard as /devices/fpga:06/serio0/input/input0

Please press Enter to activate this console. input: ImExPS/2 Generic Explorer Mo
use as /devices/fpga:07/serio1/input/input1

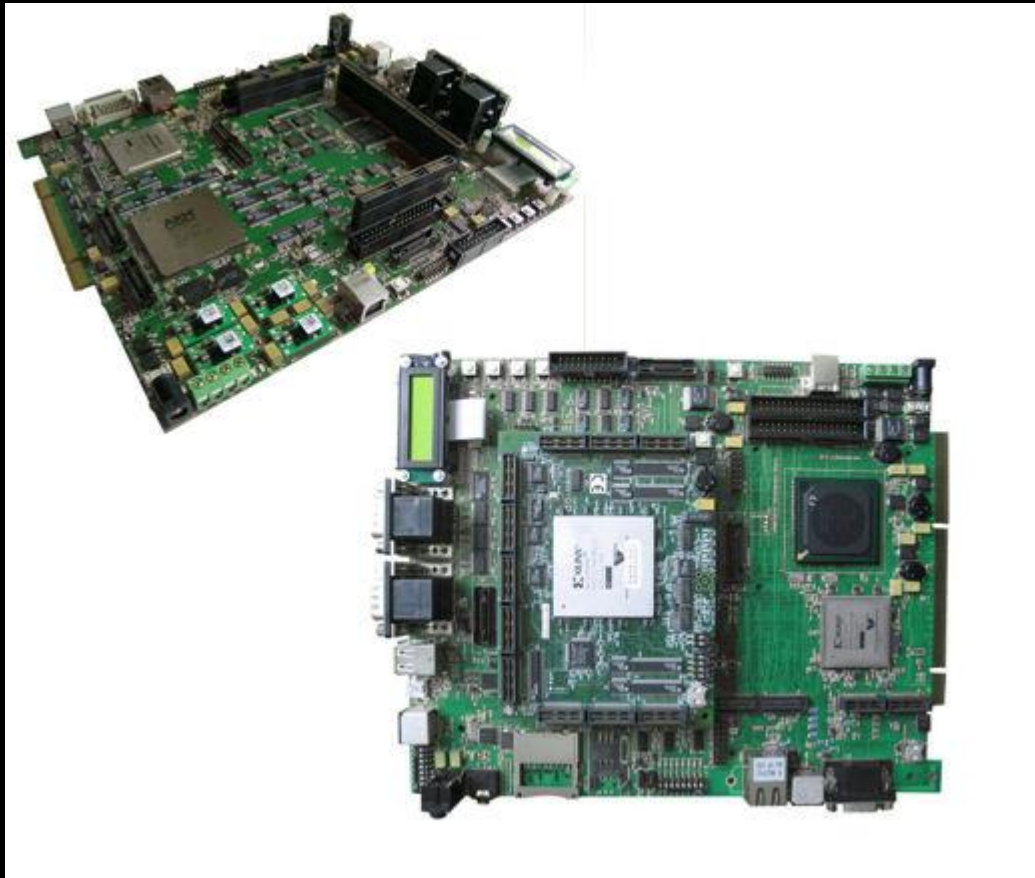
/ # id
uid=0 gid=0
/ # ls -al
total 4
drwxr-xr-x 10 0 0 0 Jan 1 00:00 .
drwxr-xr-x 10 0 0 0 Jan 1 00:00 ..
-rw-r--r-- 1 0 0 0 10 Jan 1 00:00 .ash_history
drwxr-xr-x 2 0 0 0 Aug 14 2013 bin
drwxr-xr-x 3 0 0 0 Jan 1 00:00 dev
drwxr-xr-x 3 0 0 0 Aug 14 2013 etc
lrwxrwxrwx 1 0 0 0 Aug 14 2013 linuxrc -> bin/busybox
dr-xr-xr-x 35 0 0 0 Jan 1 00:00 proc
drwx----- 2 0 0 0 Aug 14 2013 root
drwxr-xr-x 2 0 0 0 Aug 14 2013 sbin
dr-xr-xr-x 12 0 0 0 Jan 1 00:00 sys
drwxr-xr-x 4 0 0 0 Aug 14 2013 usr
/ #
```

리눅스 배포본이란?

- 기본 리눅스 커널을 기반 위에 어떤 Root File System 및 Interface를 구성하느냐에 따라 다른 배포본이 된다.
- Ubuntu Linux
- Fedora Linux
- Android Linux

실제 장비에 넣기

- <http://www.arm.com/products/tools/development-boards/versatile/platform-baseboards.php>



실제 장비에 넣기

- ROM Writer
 - Writing 전용 장비
 - Flash memory에 writing
- JTAG
 - 하드웨어 디버깅 장비
 - Flash memory에 writing



부트로더란?

- 초기 OS 개발 시 엄청난 시행착오를 겪으며 OS를 새로 업로드 해야한다.
- 매번 장비를 사용하기 너무 귀찮다!!
 - 네트워크로 전송할 수는 없을까?
- 부트로더 안에 펌웨어 업데이트 기능을 넣어서 진행

Custom Rom 만들기

- 커스텀롬
 - 소스코드를 기반으로 수정하여 제작
 - Ex> 카메라 사운드 함수 삭제
- 쿠킹롬
 - 기존의 파일들을 추출한 후 수정하여 제작
 - Ex> 카메라 사운드 파일 삭제

Custom Rom 만들기

- 장점

- 스마트폰을 느리게하는 불필요한 프로그램들 삭제
 - Ex> SK 마켓 ...
- 바탕화면, 시작화면 테마 변경
- 성능 향상
 - Ex> 스케줄링 개선
- 여유를 가지고 천천히 개발하다보니 오히려 순정롬보다 좋은 품질이~
 - 반면 개발자들을 due에 쫓김

Flashing

- KIES
 - 삼성 개발
 - 공식 펌웨어 업그레이드 소프트웨어

- CWM
 - by clockwork mod
 - <http://caleb1783.tistory.com/236>

ODIN
- by 삼성 (leaked)

DNW

- 삼성 개발
- Usb 기반 펌웨어 업로드 툴

HEIMDALL
- by AdamOutler

FASTBOOT

- 안드로이드 제작
 - fastboot flash bootloader <location/name_of_bootloader_file.img>
 - <http://rootzwiki.com/topic/28544-guide-nexus-7-bootloadersrecoveriesrootback-to-stock/#entry764328>
 - 리눅스커맨드 기반
 - 예제
- http://techshek4u.blogspot.co.uk/2012/01/applying-card-emulation-patch-to_03.html

Flashing

- KIES
 - 삼성 개발
 - 공식 펌웨어 업그레이드 소프트웨어
- DNW
 - 삼성 개발
 - 개발자들 사이에 사용되는 펌웨어 업그레이드 소프트웨어
- ODIN
 - 삼성 개발 (leaked)

Flashing

- FastBoot
 - 안드로이드 개발
 - 펌웨어 업그레이드 툴
 - 리눅스 커맨드 기반
- CWM
 - 유명한 커스텀롬 개발 그룹에서 제작
- Heimdall
 - 개인(AdamOutler)이 개발하여 배포

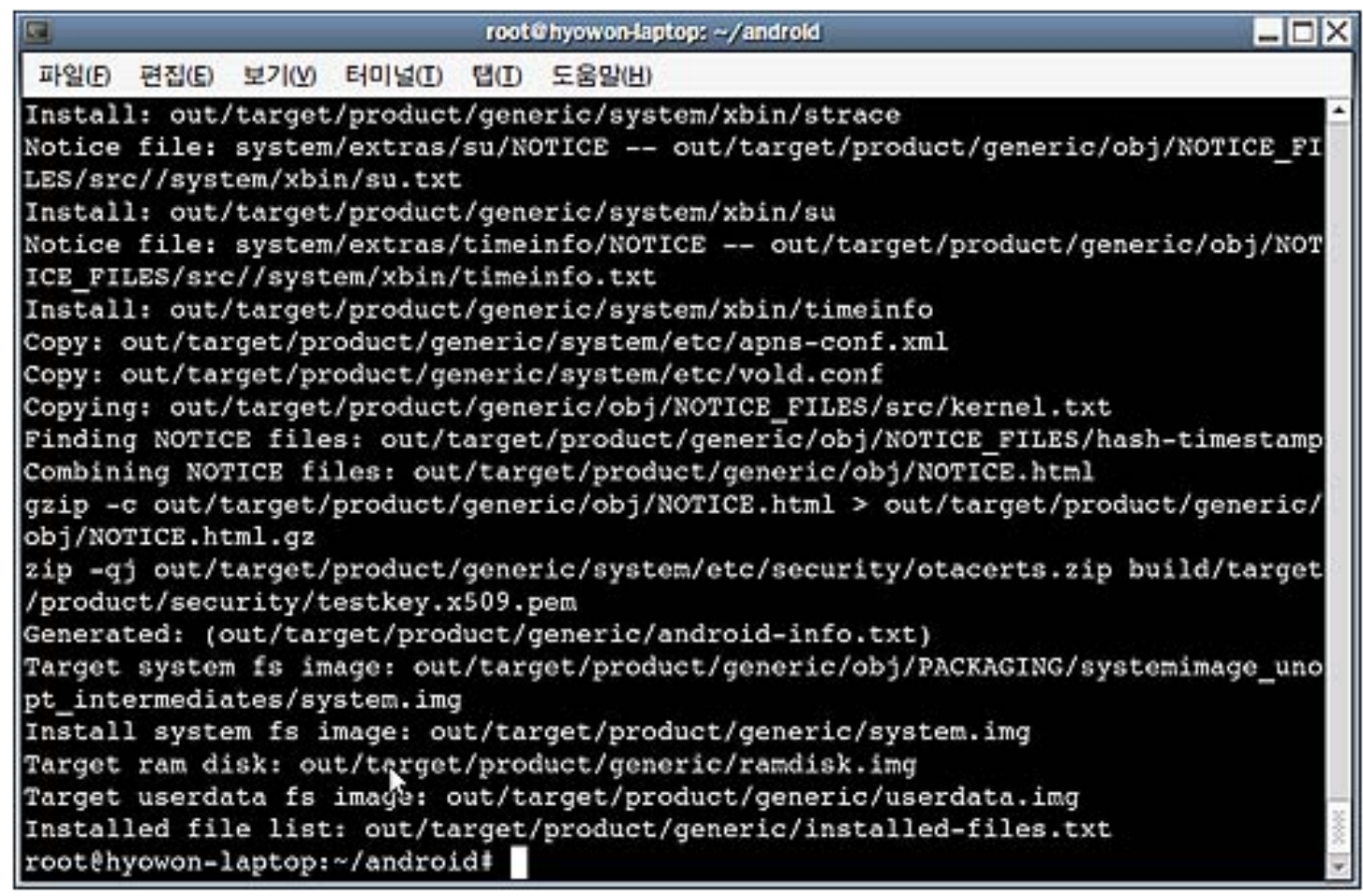
안드로이드 전용 커널 소스

- 스마트폰 및 태블릿에 특화된 커널
 - 불필요한 기능 제거
 - 필요한 기능 추가
- 리눅스 커널 소스를 기반으로 만들어짐
- <http://source.android.com/source/downloading.html>

안드로이드 전용 Root File System

```
# mkdir ~/bin
# curl http://android.git.kernel.org/repo > ~/bin/repo
# chmod 755 ~/bin/repo
# cp ~/bin/repo /bin
# mkdir ~/mydroid
# cd ~/mydroid
# repo init -u
    git://android.git.kernel.org/platform/manifest.git
# cd ~/mydroid
# repo sync
```

안드로이드 전용 Root File System



```
root@hyowon-laptop: ~/android
파일(F) 편집(E) 보기(V) 터미널(T) 탭(I) 도움말(H)
Install: out/target/product/generic/system/xbin/strace
Notice file: system/extras/su/NOTICE -- out/target/product/generic/obj/NOTICE_FILES/src//system/xbin/su.txt
Install: out/target/product/generic/system/xbin/su
Notice file: system/extras/timeinfo/NOTICE -- out/target/product/generic/obj/NOTICE_FILES/src//system/xbin/timeinfo.txt
Install: out/target/product/generic/system/xbin/timeinfo
Copy: out/target/product/generic/system/etc/apns-conf.xml
Copy: out/target/product/generic/system/etc/vold.conf
Copying: out/target/product/generic/obj/NOTICE_FILES/src/kernel.txt
Finding NOTICE files: out/target/product/generic/obj/NOTICE_FILES/hash-timestamp
Combining NOTICE files: out/target/product/generic/obj/NOTICE.html
gzip -c out/target/product/generic/obj/NOTICE.html > out/target/product/generic/obj/NOTICE.html.gz
zip -qj out/target/product/generic/system/etc/security/otacerts.zip build/target/product/security/testkey.x509.pem
Generated: (out/target/product/generic/android-info.txt)
Target system fs image: out/target/product/generic/obj/PACKAGING/systemimage_unopt_intermediates/system.img
Install system fs image: out/target/product/generic/system.img
Target ram disk: out/target/product/generic/ramdisk.img
Target userdata fs image: out/target/product/generic/userdata.img
Installed file list: out/target/product/generic/installed-files.txt
root@hyowon-laptop:~/android#
```

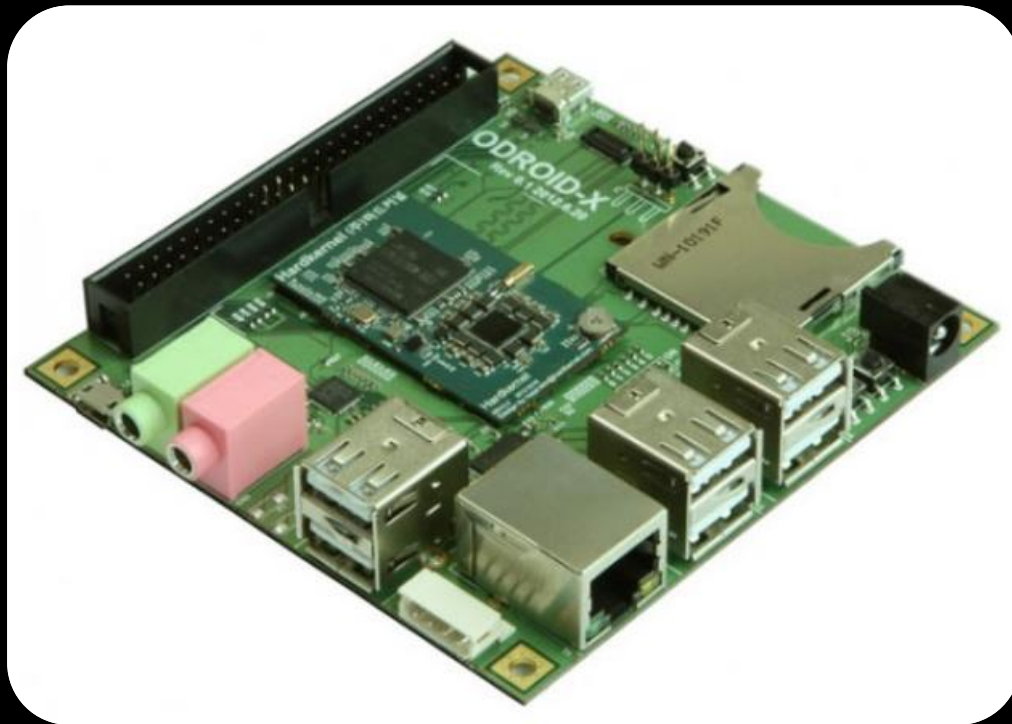

QEMU에 android 올리기

- <http://www.kandroid.org/board/board.php?board=androidsource&command=body&no=24>



실제 장비에 android 올리기

- ODROID-X
- http://hardkernel.com/renewal_2011/shop/good_list.php?lang=en



참고 자료

- QEMU+ARM
 - <http://mmmyddd.freeshell.net/wiki/embed/linuxonarmqemu.html>
- QEMU+ANDROID
 - <http://www.kandroid.org/board/board.php?board=androidsource&command=body&no=24>
- ODROID-X
 - <http://dev.odroid.com/projects/ics/>

Q/A

감사합니다.