

## 1. 램디스크 이미지 제작

커널이미지와 함께 만들어져야 하는 것이 램디스크 이미지 이다.  
 여기서는 램디스크 이미지를 만드는 것에 대하여 설명하고자 한다.

### 1.1. 램디스크란?

램디스크는 별다른 물리적 장치를 지칭하는 것이 아니라, 메모리의 일부분을 디스크 드라이브로 인식시켜 이를 하드디스크처럼 사용하기 위한 것이다.  
 타겟보드에는 하드디스크가 없으므로 이 램디스크에 의해 부팅이 된다.  
 램디스크에는 커널이 완전히 로드 된 후, 시스템의 구동에 필요한 `init`, 유틸리티 등 환경파일과 바이너리 파일 등이 있어야 부팅이 된다.  
 이런 유틸리티들과 라이브러리들을 모아 압축한 것이 램디스크 이미지이다.  
 램은 전원이 꺼지면 기억된 내용이 지워지므로 램디스크 이미지를 특정한 위치에 저장하여야 한다.  
 타겟보드는 램에 직접 써 넣는 방법과, 플래시를 이용하여 저장하는 두 가지 방법을 제시하고 있다.

### 1.2. 램디스크 설정

램디스크는 Kernel Configuration에서 설정한다.

```
# make menuconfig
```

**Block devices** —>

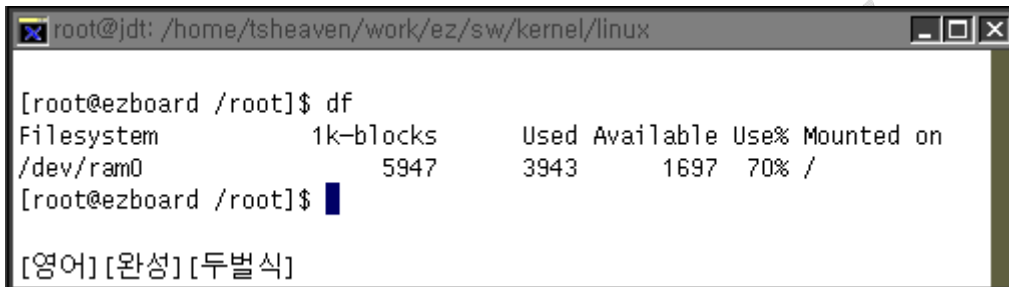


### 1.3. 램디스크 확인

램디스크는 램디스크 디바이스인 /dev/ram0 나 dev/ram1등에 기록된다.

/dev/ram0, /dev/ram1, /dev/ram, /dev/ramdisk등의 디바이스가 있는지를 확인하고. 만약 없다면 mknod 명령으로 /dev/ram0를 만들어야 한다.

타겟보드에서 리눅스 구동 후 login 하여 df 명령으로 현재 파일 시스템이 어디에 저장되어 있는지를 확인할 수 있다.



```
root@jdt: /home/tsheaven/work/ez/sw/kernel/linux
[root@ezboard /root]$ df
Filesystem            1k-blocks      Used Available Use% Mounted on
/dev/ram0              5947          3943      1697    70% /
[root@ezboard /root]$
```

위와 같이 되는 경우와 /dev/roofs 가 /에 동일하게 마운트되어 있어도 상관은 없다.

### 1.4. 기존 램디스크 이미지 구하기

<http://www.falinux.com/win/download.html> 사이트에서 받아온다.

타겟보드 램디스크 이미지는 glibc 2.3.3 버전용이다.

### 1.5. 램디스크 이미지 다루기

유틸리티와 라이브러리들을 추가 수정할 경우 또는 디바이스 드라이브나 어플리케이션을 작성하여 램디스크 이미지에 추가할 경우에 램디스크 이미지 압축을 풀어 작업을 하고 다시 압축을 하여야 한다.

#### ◆ 램디스크 이미지 복사하기

여기서는 제공한 CD에서 램디스크 이미지를 복사하기로 한다.

모든 작업 디렉토리는 /ez/sw/ramdisk 이다. 아래와 같이 작업 디렉토리를 만든다.

```
# mkdir /ez/sw/ramdisk
```

램디스크 이미지를 복사한다.

```
# cp -a /mnt/cdrom/sw/ramdisk/ramdisk.gz /ez/sw/ramdisk
```

작업 디렉토리로 이동한다.

```
# cd /ez/sw/ramdisk
```

```

root@jdt: /
[ root@jdt / ]# mkdir /ez/sw/ramdisk
[ root@jdt / ]# cp -a /mnt/cdrom/sw/ramdisk/ramdisk.gz /ez/sw/ramdisk/
[ root@jdt / ]#
[ root@jdt / ]# cd /ez/sw/ramdisk/
[ root@jdt ramdisk ]# ls -l
total 2080
-rwxr--r--  1 root  root    2124917 Oct 19 11:30 ramdisk.gz
[ root@jdt ramdisk ]#
[영어] [완성] [두벌식]

```

◆ 램디스크 이미지 압축 풀기

램디스크 이미지 압축을 풀어 마운트 할 작업 디렉토리를 만든다.

**# mkdir ramdisk\_dir**

램디스크 이미지 압축을 푼다.

**# gzip -d ramdisk.gz**

압축을 푼 램디스크 이미지를 마운트하고, df 명령으로 마운트 되었는지 확인한다.

**# mount -t ext2 -o loop ramdisk ramdisk\_dir**

**# df**

```

root@jdt: /home/tsheaven/work/ez/sw/ramdisk
[ root@jdt ramdisk ]# mkdir ramdisk_dir
[ root@jdt ramdisk ]#
[ root@jdt ramdisk ]# ls -l
total 2084
-rwxr--r--  1 root  root    2124917 Oct 19 11:30 ramdisk.gz
drwxr--r-x  2 root  root      4096 Oct 22 14:23 ramdisk_dir
[ root@jdt ramdisk ]#
[ root@jdt ramdisk ]# gzip -d ramdisk.gz
[ root@jdt ramdisk ]#
[ root@jdt ramdisk ]# ls -l
total 6160
-rwxr--r--  1 root  root    6291456 Oct 19 11:30 ramdisk
drwxr--r-x  2 root  root      4096 Oct 22 14:23 ramdisk_dir
[ root@jdt ramdisk ]#
[ root@jdt ramdisk ]# mount -t ext2 -o loop ramdisk ramdisk_dir/
[ root@jdt ramdisk ]#
[ root@jdt ramdisk ]# df
Filesystem            1k-blocks      Used Available Use% Mounted on
/dev/hde2              28731484  23603764   3668216   87% /
/eز/sw/ramdisk/ramdisk
                               5947         5184         456   92% /ez/sw/ramdisk/ramdisk_dir
[ root@jdt ramdisk ]#
[영어] [완성] [두벌식]

```

위와 같은 명령어를 입력 하면 램디스크 이미지가 `./ramdisk_dir` 디렉토리에 풀리게 된다. 이렇게 생성된 램디스크 이미지를 바탕으로 하여 자신이 arm용으로 컴파일한 실행파일들이나 라이브러리등을 복사해서 넣는다.

#### ◆ 램디스크 이미지 압축 하기

원하는 대로 램디스크를 만든 후에 다음의 작업을 하여 압축한다.

마운트를 해제한다.

```
# umount ramdisk_dir
```

램디스크 이미지 압축한다.

```
# gzip ramdisk
```

```
root@jdt: /home/tsheaven/work/ez/sw/ramdisk
[root@jdt ramdisk]# ls -l
total 6157
-rwxr--r--  1 root   root    6291456 Oct 19 11:30 ramdisk
drwxr-xr-x 14 root   root      1024 Oct  2 18:53 ramdisk_dir
[root@jdt ramdisk]#
[root@jdt ramdisk]# umount ramdisk_dir/
[root@jdt ramdisk]#
[root@jdt ramdisk]# gzip ramdisk
[root@jdt ramdisk]#
[root@jdt ramdisk]# ls -l
total 2084
-rwxr--r--  1 root   root    2124914 Oct 19 11:30 ramdisk.gz
drwxr-xr-x  2 root   root       4096 Oct 22 14:23 ramdisk_dir
[root@jdt ramdisk]#
[영어] [완성] [두벌식]
```

램디스크 이미지를 바꾼 후에 타겟보드로 다운로드 한다.

다운로드 하는 방법은 [14장.이지부트 매뉴얼]을 참고하기 바란다.

## [ 참고 ]

다음의 내용은 타겟보드의 네트워크 관리를 위해 이더넷의 IP 설정을 예제로 하여 램디스크 이미지를 바꾸어 보겠다.

## ◆ 램디스크 이미지 압축 풀기

램디스크 이미지 압축을 풀어 마운트 할 작업 디렉토리를 만든다.

```
# mkdir ramdisk_dir
```

램디스크 이미지 압축을 푼다.

```
# gzip -d ramdisk.gz
```

압축을 푼 램디스크 이미지를 마운트하고, df 명령으로 마운트 되었는지 확인한다.

```
# mount -t ext2 -o loop ramdisk ramdisk_dir
```

## ◆ 이더넷 설정 수정하기

마운트한 ramdisk\_dir 디렉토리로 이동한다.

```
# cd ramdisk_dir
```

이더넷 설정을 하는 스크립트가 있는 파일로 이동한다.

```
# cd etc/rc.d
```

이더넷 설정을 수정한다.

```
# vi rc.local
```

## [ 수정 전 ]

```
ifconfig lo127.0.0.1 up
if [ -f /root/ez_ip ]; then
    /root/ez_ip
else
    ifconfig eth0 192.168.10.125 netmask 255.255.255.0 up
fi
route add default gw 192.168.10.1 eth0
```

위의 내용을 살펴보면 /root/ez\_ip 라는 파일이 있으면 bootloader 에서 설정한 IP를 사용하고, 만약 /root/ez\_ip 라는 파일이 없으면 ifconfig 로 설정하게 된다.

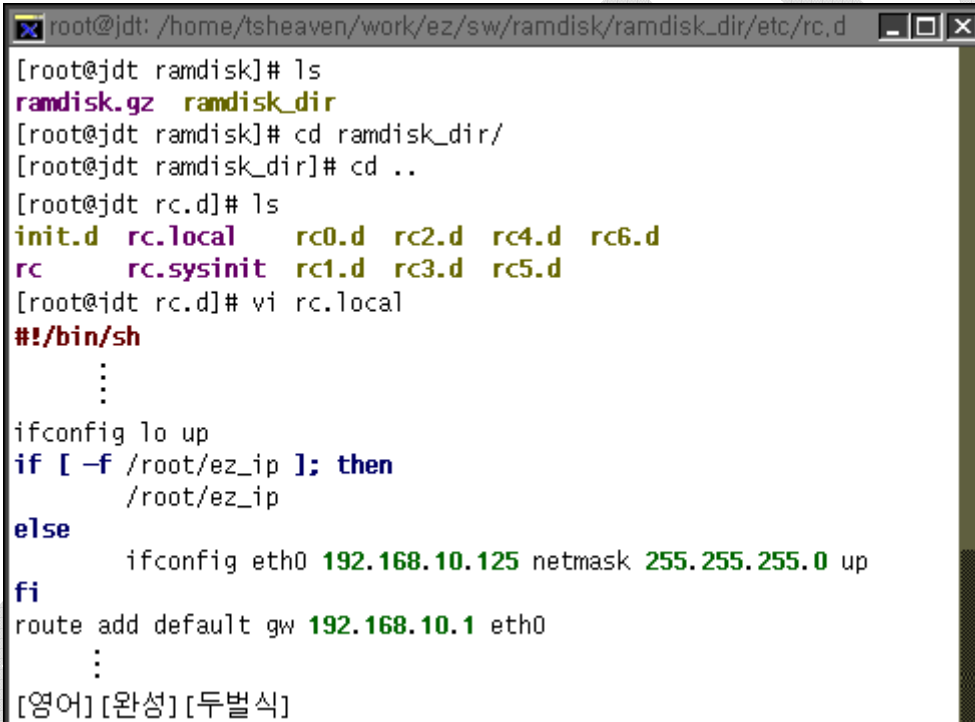
이 예제는 단지 램디스크 이미지를 풀고, 수정하고, 다시 압축하는 것에 대하여 설명하기 위한 것이므로 다음과 같이 수정하도록 하자.

[ 수정 후 ]

```
ifconfig lo 127.0.0.1 up
ifconfig eth0 211.217.10.100 netmask 255.255.255.0 up
route add default gw 211.217.10.1 eth0
```

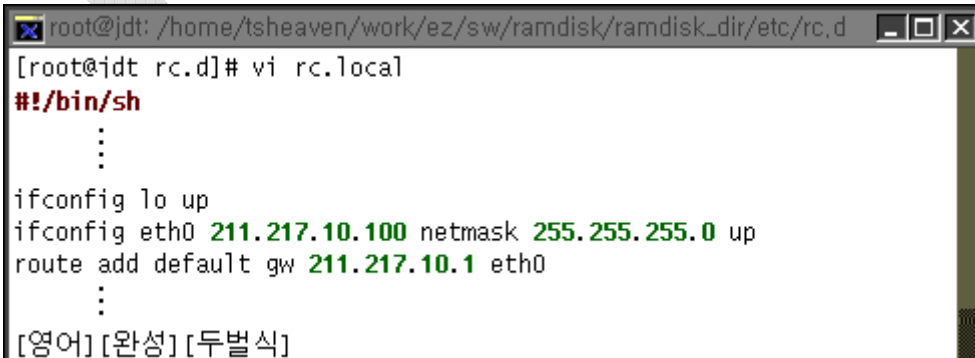
이렇게 수정을 하면 부트로더에서 설정한 IP 주소를 할당 받지 않고, 고정으로 IP를 할당 받게 된다.

[ rc.local 수정 전 ]



```
root@jdt: /home/tsheaven/work/ez/sw/ramdisk/ramdisk_dir/etc/rc.d
[ root@jdt ramdisk ]# ls
ramdisk.gz ramdisk_dir
[ root@jdt ramdisk ]# cd ramdisk_dir/
[ root@jdt ramdisk_dir ]# cd ..
[ root@jdt rc.d ]# ls
init.d rc.local rc0.d rc2.d rc4.d rc6.d
rc rc.sysinit rc1.d rc3.d rc5.d
[ root@jdt rc.d ]# vi rc.local
#!/bin/sh
:
:
ifconfig lo up
if [ -f /root/ez_ip ]; then
    /root/ez_ip
else
    ifconfig eth0 192.168.10.125 netmask 255.255.255.0 up
fi
route add default gw 192.168.10.1 eth0
:
[영어] [완성] [두벌식]
```

[ rc.local 수정 후 ]



```
root@jdt: /home/tsheaven/work/ez/sw/ramdisk/ramdisk_dir/etc/rc.d
[ root@jdt rc.d ]# vi rc.local
#!/bin/sh
:
:
ifconfig lo up
ifconfig eth0 211.217.10.100 netmask 255.255.255.0 up
route add default gw 211.217.10.1 eth0
:
[영어] [완성] [두벌식]
```

## ◆ 램디스크 이미지 압축 하기

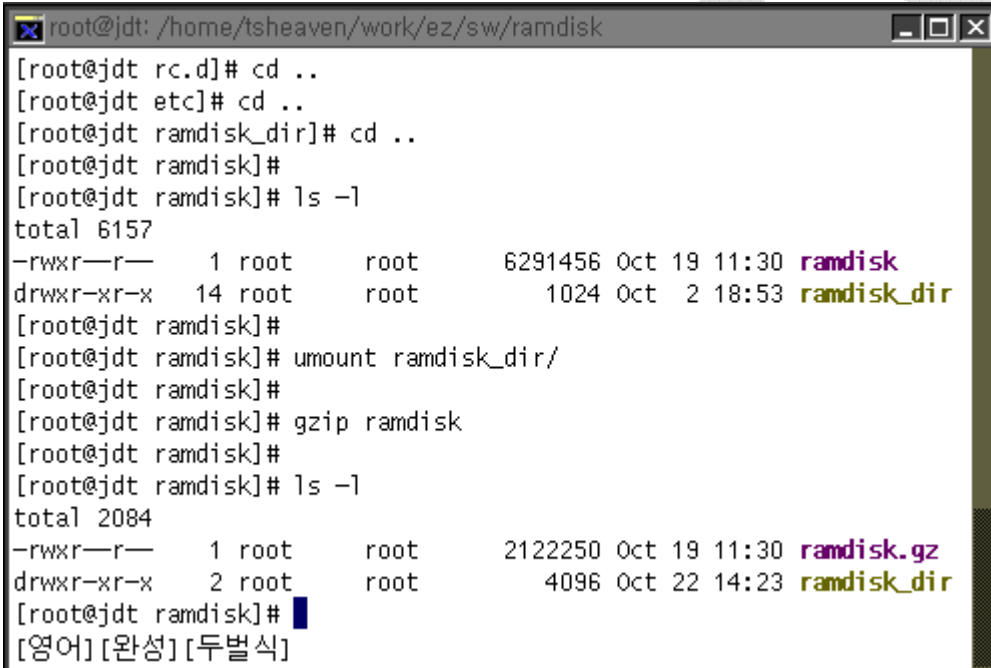
원하는 대로 램디스크를 만든 후에 다음의 작업을 하여 압축한다.

마운트를 해제한다.

```
# umount ramdisk_dir
```

램디스크 이미지를 압축한다.

```
# gzip ramdisk
```



```
root@jdt: /home/tsheaven/work/ez/sw/ramdisk
[root@jdt rc.d]# cd ..
[root@jdt etc]# cd ..
[root@jdt ramdisk_dir]# cd ..
[root@jdt ramdisk]#
[root@jdt ramdisk]# ls -l
total 6157
-rwxr--r--  1 root    root      6291456 Oct 19 11:30 ramdisk
drwxr-xr-x 14 root    root        1024 Oct  2 18:53 ramdisk_dir
[root@jdt ramdisk]#
[root@jdt ramdisk]# umount ramdisk_dir/
[root@jdt ramdisk]#
[root@jdt ramdisk]# gzip ramdisk
[root@jdt ramdisk]#
[root@jdt ramdisk]# ls -l
total 2084
-rwxr--r--  1 root    root      2122250 Oct 19 11:30 ramdisk.gz
drwxr-xr-x  2 root    root         4096 Oct 22 14:23 ramdisk_dir
[root@jdt ramdisk]#
[영어] [완성] [두벌식]
```

램디스크 이미지를 바꾼 후에 타겟보드로 다운로드 한다.

다운로드 하는 방법은 [14장.이지부트 매뉴얼]을 참고하기 바란다.

## 2. 램디스크 디렉토리 구성도

이진보드에 제공되는 램디스크의 내용은 다음과 같다.

```

/
home dev etc var usr lib sbin bin root tmp mnt proc
home dev etc var usr lib sbin bin root tmp mnt proc sys app
    
```

home : 사용자 영역 디렉토리  
 dev : 디바이스 디렉토리  
 etc : 환경 디렉토리  
 var : 환경 및 메세지 및 스펠 디렉토리  
 usr : 일반 어플리케이션 디렉토리  
 lib : 공용 라이브러리 디렉토리(중요)  
 sbin : 관리자 실행파일 디렉토리  
 bin : 실행 디렉토리  
 root : 루트영역 디렉토리  
 tmp : 임시 영역 디렉토리  
 mnt : 디바이스 마운트 영역  
 proc : 실행 프로세서 디렉토리(자동생성)  
 sys : 시스템 상태 디렉토리(자동생성)  
 app : 사용자 영역 디렉토리 - mtblock2

### */home/*

### */dev/*

console	hdb	hdd8	mtd8	mtdr15	sda	sdc8	ttyCL2	ttySS10
cusa0	hdb1	initctl	mtd9	mtdr2	sda1	sdd	ttyP0	ttySS11
cusa1	hdb2	kmem	mtblock0	mtdr3	sda2	sdd1	ttyP1	ttySS12
dsp	hdb3	mem	mtblock1	mtdr4	sda3	sdd2	ttyP2	ttySS13
fb	hdb4	mixer	mtblock10	mtdr5	sda4	sdd3	ttyP3	ttySS14
fb0	hdb5	mk712	mtblock11	mtdr6	sda5	sdd4	ttyS0	ttySS15
fb1	hdb6	mmca	mtblock12	mtdr7	sda6	sdd5	ttyS00	ttySS2
fb2	hdb7	mmca1	mtblock13	mtdr8	sda7	sdd6	ttyS01	ttySS3
fb3	hdb8	mmca2	mtblock14	mtdr9	sda8	sdd7	ttyS02	ttySS4
fb4	hdc	mmca3	mtblock15	null	sdb	sdd8	ttyS03	ttySS5
fb5	hdc1	mmca4	mtblock16	phub	sdb1	sound_dsp	ttyS04	ttySS6
fb6	hdc2	mouse	mtblock2	psaux	sdb2	tty	ttyS05	ttySS7



fb7	hdc3	mtd0	mtdblock3	psystemmap	sdb3	tty0	ttyS06	ttySS8
flh0	hdc4	mtd1	mtdblock4	ptmx	sdb4	tty1	ttyS1	ttySS9
flh1	hdc5	mtd10	mtdblock5	pts	sdb5	tty2	ttyS2	urandom
flh2	hdc6	mtd11	mtdblock6	ram	sdb6	tty3	ttySA0	watchdog
flh3	hdc7	mtd12	mtdblock7	ram0	sdb7	tty4	ttySA1	wdt
hda	hdc8	mtd13	mtdblock8	ram1	sdb8	tty5	ttySA2	zero
hda1	hdd	mtd14	mtdblock9	ram2	sdC	tty6	ttySAC0	
hda2	hdd1	mtd15	mtdr0	ram3	sdC1	tty7	ttySAC1	
hda3	hdd2	mtd2	mtdr1	random	sdC2	tty8	ttySAC2	
hda4	hdd3	mtd3	mtdr10	remocon	sdC3	ttyAM0	ttySP0	
hda5	hdd4	mtd4	mtdr11	root	sdC4	ttyAM1	ttySP1	
hda6	hdd5	mtd5	mtdr12	rs232sa1	sdC5	ttyAM2	ttySP2	
hda7	hdd6	mtd6	mtdr13	rs232sa2	sdC6	ttyCL0	ttySS0	
hda8	hdd7	mtd7	mtdr14	rtc	sdC7	ttyCL1	ttySS1	

***/sbin/***

bootlogd	halt	initlog	losetup	modprobe	route	swapon
e2fsck	hdparm	insmod	lsmod	pivot_root	runlevel	sysctl
eraseall	hwclock	killall5	mke2fs	poweroff	shutdown	syslogd
fdisk	ifconfig	klogd	mklost+found	reboot	sulogin	telinit
getty	init	logread	mkswap	rmmod	swapoff	udhcpc

- modprobe ; 모듈 적재
- insmod ; 모듈 적재
- rmmod ; 모듈 지우기
- lsmod ; 모듈 정보 출력하기
- ifconfig ; 네트워크 인터페이스 환경 설정
- route ; 네트워크 경로 설정 명령
- telinit ; 원격 접속하기
- rarp ; rarp 테이블 관리하기
- mkswap ; 리눅스 스왑 영역을 지정한다
- swapon ; 페이징, 스와핑용 장치/화일 사용 설정
- swapoff -> swapon ; 스왑 해제하기
- hdparm ; 하드디스크 파라미터 설정
- shutdown ; 시스템 종료
- reboot -> halt ; 재부팅 하기
- halt ; 시스템 종료

init ; 초기화 프로세서 제어  
 pidof ; 실행중인 프로그램의 프로세스 ID 찾기  
 killall5 ; 자신을 제외한 모든 프로세스를 죽인다.  
 syslogd ; 로그인 과정 설정하기

**/bin/**

addgroup chgrp deluser fgrep login mount ps stty umount  
 adduser chmod df grep ls mountpoint pwd su uname  
 bash chown dmesg gunzip lsattr mv rm sync usleep  
 bashbug cp echo gzip mkdir netstat rmdir tar uuiddgen  
 busybox date egrep hostname mknod nice sed tinylogin vi  
 cat dd false kill mktemp pidof sh touch zcat  
 chattr delgroup fdflush ln more ping sleep true

stty ; 터미널 설정 변경  
 sh -> bash ; 셸  
 ps ; 프로세서 상태 보기  
 bash ; 셸  
 busybox ;  
 vi ; 편집기  
 netstat ; 네트워크 상태 보기  
 nice ; 프로세스 우선 순위 변경하기  
 mount ; 마운트 명령  
 umount ; 마운트 해제  
 addgroup ; 그룹 계정 만들기  
 adduser ; 사용자 계정 만들기  
 delgroup ; 그룹 계정 삭제  
 deluser ; 사용자 계정 삭제  
 login ; 시스템 로그인  
 su ; 루트권한 변경 또는 사용자 전환  
 cat ; 문서 보여주기 명령  
 chgrp ; 파일, 디렉토리가 속했던 그룹 바꾸기  
 chmod ; 파일 속성 변경  
 chown ; 파일 소유자 변경  
 cp ; 파일 카피 명령  
 dd ; 파일 생성 및 변환 명령

df ; 디스크 사용상태 보기 명령  
 dmesg ; 부팅 메시지 로그 기록 파일  
 du ; 디스크 사용량  
 gunzip ; 압축 명령  
 gzip ; 압축 명령  
 hostname ; 호스트 네임 관련 사항 보기  
 kill ; 프로세서 죽이기  
 ln ; 링크하기  
 ls ; 디렉토리 목록 보기  
 mkdir ; 디렉토리 생성 명령  
 mknod ; 특수 파일 만들기  
 more ; 화면 단위로 출력하기  
 mv ; 파일 이동 명령  
 ping ; ping  
 pwd ; 현재 디렉토리 보기  
 rm ; 파일 제거  
 rmdir ; 디렉토리 지우기  
 sleep ; 잠시 쉬기  
 sync ; 버퍼 재설정 하기  
 tar ; 파일 합치기  
 tee ; 표준 입력으로부터 표준 출력 만들기  
 true ; 종료 코드 리턴하기  
 uname ; 시스템 정보 보기

*/etc/*

HOSTNAME	host.conf	inittab	localtime	profile	services
ae.rc	hosts	inputrc	modules	protocols	shadow
ae2vi.rc	hosts.allow	issue	motd	rc.d	termcap
filesystems	hosts.deny	issue.net	networks	resolv.conf	timezone
fstab	inetd.conf	ld.so.cache	nsswitch.conf	rpc	
group	initscript.sample	ld.so.conf	passwd	securetty	

fstab ; 파일 마운트 환경 파일  
 mtab ; 마운트 된 정보 저장 파일 (자동생성)  
 group ; 그룹 환경 파일  
 host.conf ; 호스트명의 IP를 얻을때 검색 순서 설정 파일

hosts ; 호스트 IP 맵 환경 파일  
 inittab ; 시스템 실행 레벨 환경 파일  
 issue ; login시에 나타나는 메세지 파일  
 issue.net ; 네트워크로 연결시 나타나는 메세지 파일  
 motd ; login직후에 나타나는 메세지 파일  
 passwd ; 암호를 포함한 유저 환경 파일  
 profile ; 셸이 사용하는 기본 속성  
 protocols ; 포트 사용 리스트  
 resolv.conf ; DNS 찾기 환경 파일  
 security ; login시 사용될 Console 출력을 위한 포트 설정 환경 파일  
 services ; 포트 사용 리스트  
 HOSTNAME ; 호스트명  
 ld.so.cache ; ldconfig에서 검색할 라이브러리디렉토리를 지정한 곳

***/etc/rc.d/***

init.d mkip rc rc.sysinit rc1.d rc3.d rc5.d  
 mkclink mkissue rc.local rc0.d rc2.d rc4.d rc6.d

mkip ; 부트로드에서 넘어온 IP로 설정하게 하는 스크립터 파일  
 mkclink ; 부트로드에서 넘어온 Console 포트를 link 하는 스크립터 파일  
 mkissue ; login시에 나타나는 메시지를 표출하는 스크립터 파일

***/etc/rc.d/init.d/***

functions halt inet killall modutils single syslog

***/etc/rc.d/rc0.d/***

K50inet K95syslog S00halt

***/etc/rc.d/rc1.d/***

K50inet K95syslog

***/etc/rc.d/rc2.d/***

K50inet K95syslog

***/etc/rc.d/rc3.d/***

S05syslog S50inet

***/etc/rc.d/rc4.d/***

S05syslog S50inet

***/etc/rc.d/rc5.d/***

S05syslog S50inet

***/etc/rc.d/rc6.d/***

K50inet K95syslog S00reboot

***/var/***

lib adm log spool run lock tmp

adm -> log

tmp -> /tmp

***/var/lib/***

***/var/log/***

dmesg lastlog messages wtmp

***/var/spool/***

cron

***/var/spool/cron/***

crontabs

***/var/spool/cron/crontabs/***

***/var/run/***

utmp runlevel.dir inetd.pid

***/var/lock/***

subsys

***/var/tmp/***

***/usr/***

arm-linux bin include local sbin share

***/usr/arm-linux/***

adm lib tmp

adm -> ../var/log

lib -> /lib

tmp -> /tmp

***/usr/bin/***

arping	cut	ftp	lastb	passwd	sz	tr	vlock	xargs
basename	deallocvt	head	logger	printf	tail	traceroute	wall	yes
bunzip2	dirname	hexdump	md5sum	readlink	tee	tty	wc	
bzcat	du	id	mesg	reset	telnet	uniq	wget	
chvt	env	install	nc	rz	tftp	unzip	which	
clear	expr	killall	nslookup	sort	time	uptime	who	
cmp	free	last	openvt	strings	top	utmpdump	whoami	

***/usr/include***

initreq.h

***/usr/local/***

***/usr/sbin/***

arp chroot in.telnetd inetd rdate tcpd telnetd

***/usr/share/***

terminfo

***/usr/share/terminfo/***

a d l r s v x

***/usr/share/terminfo/a***

ansi

***/usr/share/terminfo/d***

dumb

***/usr/share/terminfo/l/***

linux

***/usr/share/terminfo/r/***

rxvt rxvt-m

***/usr/share/terminfo/s/***

sun

***/usr/share/terminfo/v/***

vt100 vt102 vt220 vt52

***/usr/share/terminfo/x/***

xterm

***/lib/***

Mcrtl.o	libe2p.so	libnss_nisplus.so
Scr1.o	libe2p.so.2	libnss_nisplus.so.2
crt1.o	libe2p.so.2.3	libpcprofile.so
crti.o	libext2fs.so	libproc.so
crtn.o	libext2fs.so.2	libproc.so.3.0.4
e2initrd_helper	libext2fs.so.2.4	libpthread-0.10.so
gcr1.o	libgcc_s.so	libpthread.so
ld-2.3.3.so	libgcc_s.so.1	libpthread.so.0
ld-linux.so.2	libm-2.3.3.so	libresolv-2.3.3.so
ldscripts	libm.so	libresolv.so
libBrokenLocale-2.3.3.so	libm.so.6	libresolv.so.2
libBrokenLocale.so	libmemusage.so	librt-2.3.3.so
libBrokenLocale.so.1	libncurses.so	librt.so
libSegFault.so	libncurses.so.4	librt.so.1
libanl-2.3.3.so	libncurses.so.5	libss.so
libanl.so	libncurses.so.5.3	libss.so.2
libanl.so.1	libnsl-2.3.3.so	libss.so.2.0
libblkid.so	libnsl.so	libstdc++.so
libblkid.so.1	libnsl.so.1	libstdc++.so.6
libblkid.so.1.0	libnss_compat-2.3.3.so	libstdc++.so.6.0.3
libc-2.3.3.so	libnss_compat.so	libtermcap.so
libc.so	libnss_compat.so.2	libtermcap.so.2.0.8
libc.so.6	libnss_dns-2.3.3.so	libthread_db-1.0.so
libcom_err.so	libnss_dns.so	libthread_db.so
libcom_err.so.2	libnss_dns.so.2	libthread_db.so.1
libcom_err.so.2.1	libnss_files-2.3.3.so	libutil-2.3.3.so

libcrypt-2.3.3.so	libnss_files.so	libutil.so
libcrypt.so	libnss_files.so.2	libutil.so.1
libcrypt.so.1	libnss_hesiod-2.3.3.so	libuuid.so
libdb.so	libnss_hesiod.so	libuuid.so.1
libdb.so.1	libnss_hesiod.so.2	libuuid.so.1.2
libdb.so.1.85.4	libnss_nis-2.3.3.so	libz.so
libdl-2.3.3.so	libnss_nis.so	libz.so.1
libdl.so	libnss_nis.so.2	libz.so.1.2.2
libdl.so.2	libnss_nisplus-2.3.3.so	modules

### ***/lib/modules/***

### ***/lib/ldscripts/***

armelf.x	armelf.xr	armelf.xu	armelf_linux.xc	armelf_linux.xsc
armelf.xbn	armelf.xs	armelf.xw	armelf_linux.xn	armelf_linux.xsw
armelf.xc	armelf.xsc	armelf_linux.x	armelf_linux.xr	armelf_linux.xu
armelf.xn	armelf.xsw	armelf_linux.xbn	armelf_linux.xs	armelf_linux.xw

### ***/root/***

mkflashroot nfsmnt

mkflashroot ; NAND 플래시의 mtblock2 영역을 루트 파일시스템으로 바꾼다  
nfsmnt ; NFS 마운트 스크립트 파일 ( ./nfsmnt 접속할NFSD디렉토리 )

### ***/tmp/***

### ***/mnt/***

nfs usb1 usb2 mmc hdd temp

### ***/proc/***

이 디렉토리는 부팅 후 자동으로 생성된다.

### ***/sys/***

이 디렉토리는 부팅 후 자동으로 생성된다.

### ***/app/***

mtblock2 마운트 디렉토리. 플래시 영역을 마운트한다.