Linux+FreeBSD 미니-하우투(mini-HOWTO)

Niels Kristian Bech Jensen, nkbj@image.dk v1.9, December 1998

김 덕환, sasung@pimp.snu.ac.kr

이 문서는 한 시스템에서 Linux와 FreeBSD를 함께 사용할 수 있는 방법에 대해 다루고 있다. 먼저 FreeBSD를 소개한 다음, 스왑(swap) 공간을 공유한다던가 해서 두 운영체제가 어떻게 공생공존할 수 있는지에 대해서 얘기한다. 이 글을 읽기 전에 Linux나 FreeBSD 그리고 하드 드라이브 파티션 나누기(fdisk)에 대한 어느 정도의 경험을 가지고 있어야 한다. 여기서 제공하는 팁들은 FreeBSD 2.2.2를 사용하여 테스트되었지만, 최신 버젼에서도 잘 작동할 것이다. 이 문서에 대한 어떤 코멘트나 질문, 제안들이 있다면 주저하지 말고 필자에게 메일을 보내기 바란다. 그리고 Linux를 NetBSD나 OpenBSD와 함께 사용한 경험을 가진 사람들은 반드시 필자에게 연락해주기를 바란다.

목 차

v1.0 번역: 이호선(ahlhs@cair.kaist.ac.kr) 1997년 9월 25일
v1.9 번역: 김덕환(sasung@atropos.snu.sa.ckr) 1999년 1월 18일

제 1 절 FreeBSD란?

FreeBSD `<http://www.freebsd.org/>`는 Linux와 같이 자유롭게 사용할 수 있는 Unix형태의 운영체제이다. 가장 중요한 차이점은 Linux 커널이 바닥부터 새로 쓰여진데 반해, FreeBSD는 4.4BSD-lite라 알려진 4.4BSD (Berkeley Software Distribution)의 자유롭게 재배포될 수 있는 부분에 기반을 두고 있다는 점이다. 이 사실로부터 어떤 사람들은 FreeBSD가 Linux보다 좀 더 "원조(real)" UNIX에 가깝다고 얘기하기도 한다. FreeBSD는 Intel PC 플랫폼 (i386 이상)에서만 동작하고 있다. 지금 현재 DEC Alpha와 Sun Sparc 플랫폼에 대한 포팅이 진행 중이다. *NetBSD* `<http://www.netbsd.org/>`와 *OpenBSD* `<http://www.openbsd.org/>`는 FreeBSD와 유사한데, 둘 다 좀 더 많은 플랫폼에서 동작한다. 이 세 *BSD 시스템의 하드웨어 요구사양은 Linux의 요구사양과 비슷한 수준이다.

FreeBSD의 개발은 Linux의 개발과는 다른 방식으로 진행된다. 핵심(core) 개발자 팀이 중재인 역할을 하며 프로젝트에 대한 개괄적인 통제를 한다. 커다란 변화들은 메일링 리스트에서 사전에 논의가 이루어진다. FreeBSD 프로젝트는 (Linux처럼) 두 개의 개발 트리를 가지고 있다: **"-CURRENT"**와 **"-STABLE"**. "-CURRENT" 개발 트리는 새로운 기능의 개발이 진행되는 곳이다. "-STABLE" 트리에서의 개발은 버그 수정과 몇몇 완벽하게 테스트되어진 새로운 기능으로 국한된다.

FreeBSD도 Linux처럼 자유로운 사용과 (재)배포가 가능하다. 시스템의 대부분은 BSD copyright로 릴리즈한다; 나머지는 GNU GPL이나 GNU LGPL을 따른다.

제 2 절 FreeBSD 방식의 하드 드라이브 이름붙이기(labelling)

Linux와 FreeBSD는 서로 다른 두 가지 방식으로 하드 드라이브와 파티션에 이름을 붙인다(label). 이 장에서는 그 두 가지 방식의 주요한 차이점에 대해서 다룰 것이다. 실제로 FreeBSD의 이름 붙이는 방식은 전통적인 BSD 이름 붙이는 스타일을 PC의 fdisk 파티션에 맞게 포팅해서 변환한 것이다. 그래서 그 방식은 NetBSD, OpenBSD, Ultrix, Digital Unix, SunOS, 그리고 Solaris와 같은 다른 BSD 기반 유닉스 시스템과 매우 유사하다.

2.1 FreeBSD "슬라이스(slice)"와 "파티션(partition)"

FreeBSD는 PC 하드 드라이브에 있는 파티션 테이블 4개 엔트리(entry) 중에 하나를 필요로 한다. 이 주(primary) 파티션을 FreeBSD 용어로는 **"슬라이스(slice)"**라고 부른다. 그 주 파티션에는 여덟 개까지의 파티션을 만들 수 있는데, 그러기 위해서는 `disklabel` 프로그램을 사용한다. 이 논리(logical) 파티션을 FreeBSD 용어로는 **"파티션(partion)"**이라고 부른다. 이러한 관념은 Linux(DOS도 마찬가지)가 확장(extended) 파티션에서 논리(logical) 파티션을 다루는 방법과 비슷하다. Linux(혹은 DOS)로 만든 확장 파티션에는 FreeBSD를 설치할 수는 없다. Linux `fdisk` 프로그램은 메인 메뉴에서는 FreeBSD 슬라이스로 된 BSD 파티션을 보여주지 않지만, 'b' 명령을 사용하면 BSD 디스크라벨(disklabel) 정보를 볼 수 있다. 출력은 다음과 같은 형태로 이루어진다(`/dev/hda4`가 FreeBSD 슬라이스이다):

```
bash# fdisk /dev/hda

Command (m for help): p

Disk /dev/hda: 64 heads, 63 sectors, 621 cylinders
Units = cylinders of 4032 * 512 bytes

   Device Boot   Begin    Start      End    Blocks   Id  System
/dev/hda1   *        1        1       27    54400+   83  Linux native
/dev/hda2           28       28       55     56448   83  Linux native
/dev/hda3           56       56      403    701568   83  Linux native
```

```
/dev/hda4          404      404      621   439488   a5  BSD/386

Command (m for help): b
Reading disklabel of /dev/hda4 at sector 1624897.

BSD disklabel command (m for help): p

8 partitions:
#        size   offset    fstype   [fsize bsize   cpg]
  a:    64512  1624896    4.2BSD        0     0     0   # (Cyl.  404 - 419)
  b:   104832  1689408      swap                        # (Cyl.  420 - 445)
  c:   878976  1624896    unused        0     0         # (Cyl.  404 - 621)
  e:    64512  1794240    4.2BSD        0     0     0   # (Cyl.  446 - 461)
  f:   645120  1858752    4.2BSD        0     0     0   # (Cyl.  462 - 621)

BSD disklabel command (m for help): q
bash#
```

첫 번째 열에 있는 'a'에서 'f'까지의 문자는 아래에 있는 FreeBSD 슬라이스의 예에서 보여지는 라벨(label)과 완전히 일치한다. BSD 어법(parlace)에 따라 세 개의 특별한 파티션들이 있다. 문자 'a'는 루트 파티션을 가리키고, 'b'는 스왑(swap) 파티션을 가리키며, 'c'는 전체 슬라이스를 가리킨다. 다른 파티션 형태에 이러한 문자를 배정하는 "표준" 방법에 대해서 더 많은 정보가 필요하다면 FreeBSD 문서를 참고하자.

2.2 Linux와 FreeBSD에서 드리이브와 파티션 이름붙이기(labelling)

하드 드라이브는 Linux와 FreeBSD에서 다음과 같은 방식으로 이름이 붙여진다:

	Linux	FreeBSD
First IDE drive	/dev/hda	/dev/wd0
Second IDE drive	/dev/hdb	/dev/wd1
First SCSI drive	/dev/sda	/dev/sd0
Second SCSI drive	/dev/sdb	/dev/sd1

하나의 IDE 드라이브에 있는 파티션(FreeBSD 슬라이스)는 다음과 같은 방식으로 이름이 붙여진다(예에서는 `/dev/hda`를 사용하였다):

	Linux	FreeBSD
First primary partition	/dev/hda1	/dev/wd0s1
Second primary partition	/dev/hda2	/dev/wd0s2
Third primary partition	/dev/hda3	/dev/wd0s3
Fourth primary partition	/dev/hda4	/dev/wd0s4

필자의 FreeBSD 슬라이스에 있는 파티션들은 다음과 같은 방식으로 이름이 붙어있다. 그 방식은 내정치(default)로 붙여지는 이름들이다. FreeBSD의 맞춤(custom) 설치를 한다면 그 이름들을 바꾸는 것도 가능하다 (이번 예에서는 `/dev/hda4`가 FreeBSD 슬라이스이다):

Linux label	FreeBSD label	FreeBSD mount point
/dev/hda5	/dev/wd0s4a	/
/dev/hda6	/dev/wd0s4b	swap
/dev/hda7	/dev/wd0s4e	/var
/dev/hda8	/dev/wd0s4f	/usr

Linux에서 dmesg를 실행하면, 다음과 같은 메세지로 이 사실을 확인할 수 있을 것이다(이것이 제대로 동작하려면 **UFS filesystem support**를 선택한 다음 커널을 새로 컴파일해야한다. 3.1(Linux 설치 및 준비) 절을 참조하자):

```
Partition check:
hda: hda1 hda2 hda3 hda4 < hda5 hda6 hda7 hda8 >
```

FreeBSD를 /dev/sd1s3 슬라이스(Linux 어법으로는 /dev/sdb3)에 설치하고, /dev/sdb2가 두 개의 논리 파티션(/dev/sdb5와 /dev/sdb6)을 가진 Linux 확장 파티션이라면, 앞의 예는 다음처럼 보일 것이다:

```
Linux label     FreeBSD label   FreeBSD mount point
/dev/sdb7       /dev/sd1s3a     /
/dev/sdb8       /dev/sd1s3b     swap
/dev/sdb9       /dev/sd1s3e     /var
/dev/sdb10      /dev/sd1s3f     /usr
```

이것들은 dmesg 출력에서 다음처럼 나타날 것이다.

```
Partition check:
 sdb: sdb1 sdb2 < sdb5 sdb6 > sdb3 < sdb7 sdb8 sdb9 sdb10 >
```

FreeBSD 슬라이스 *뒤에* 리눅스 확장 파티션을 가지고 있다면 곤란한 상황에 처하게 된다. 대부분의 Linux 커널 설치 플로피는 UFS 지원없이 컴파일되어 있기 때문에, 슬라이스 안에 있는 FreeBSD 파티션을 인식하지 못할 것이다. 다음과 같이 나타나야 할 것이(/dev/hda3가 FreeBSD 슬라이스이고 /dev/hda4가 Linux 확장 파티션이다)

```
Partition check:
hda: hda1 hda2 hda3 < hda5 hda6 hda7 hda8 > hda4 < hda9 hda10 >
```

다음처럼 나타난다:

```
Partition check:
hda: hda1 hda2 hda3 hda4 < hda5 hda6 >
```

이것은 잘못된 장치 배정으로 데이타를 잃어 버린다. *항상 모든 Linux 확장 파티션 뒤에 FreeBSD 슬라이스를 설치하고, FreeBSD를 설치하고 난 이후에 Linux 확장 파티션에 있는 어떠한 논리 파티션에도 변화를 가하지 마라!*는 것을 당부하고 싶다.

제 3 절 Linux와 FreeBSD간의 스왑(swap) 공간 공유

이 섹션은 필자가 Linux와 FreeBSD가 스왑(swap) 공간을 어떻게 공유시켰는가를 설명한다. 이와 같은 결과를 만들어내는데 다른 방법이 있을 수도 있다. 그렇게 하기를 원한다면 Linux보다 FreeBSD를 먼저 설치할 수 있는데, FreeBSD 슬라이스에 들어가는 파티션들의 순서에 주의를 해야한다.

3.1 Linux 설치 및 준비

첫번째 단계는 평상시처럼 Linux를 설치하는 것이다. 하드디스크에 FreeBSD 슬라이스를 위한 공간을 남겨두도록 한다. Linux 스왑(swap) 파티션을 만들 필요는 없지만, 원한다면 FreeBSD에 할당한 공간에 만든다. 그렇게 하면 나중에 Linux 스왑 파티션을 지우고, FreeBSD를 위한 공간으로 사용할 수 있다.

Linux를 인스톨한 후에는 새로운 커널을 만들어야 한다. 커널 컴파일하는 일에 익숙하지 않다면, **Linux Kernel HOWTO**를 읽어보도록 하자. 커널을 만들 때 **UFS filesystem support (read only)**와 **BSD disklabel (FreeBSD partition tables) support**를 *반드시* 포함해야 한다:

```
UFS filesystem support (read only) (CONFIG_UFS_FS) [N/y/m/?] y
BSD disklabel (FreeBSD partition tables) support (CONFIG_BSD_DISKLABEL) [N/y/?] (NEW) y
```

새 커널을 설치하고 리부팅한다. 만약 Linux 스왑 파티션을 만들었다면 /etc/fstab 파일에서 *swap*이라는 단어를 포함하는 모든 줄을 삭제하자. *새로 만든 커널을 가진 Linux 부트 플로피가 제대로 동작하는지 확인해야 한다.* 이제 FreeBSD를 설치할 차례이다.

3.2 FreeBSD 설치

FreeBSD 문서에 나와 있는 대로 FreeBSD를 설치한다. Linux 스왑 파티션을 만들었다면 지우도록 하자(FreeBSD fdisk 프로그램을 사용하면 된다). FreeBSD 슬라이스 안에 들어있는 파티션들의 순서에 주의하도록 한다. 내정된(default) 이름들을 사용한다면 두 번째 파티션이 스왑 파티션이 될 것이다. FreeBSD 설치를 마치면 *새로 만든 Linux 부트 플로피를 사용하여* Linux로 리부팅해 들어간다.

3.3 Linux에서 FreeBSD 스왑(swap) 공간 셋업

Linux로 부팅해 들어갔으면 dmesg를 실행하자. 출력에서 다음과 같은 메세지를 볼 수 있어야 한다:

```
Partition check:
 hda: hda1 hda2 hda3 hda4 < hda5 hda6 hda7 hda8>
```

이것은 /dev/hda4는 FreeBSD 슬라이스이며 /dev/hda5, /dev/hda6, /dev/hda7 그리고 /dev/hda8은 FreeBSD 파티션임을 의미한다. 만약 스왑 파티션이 슬라이스 안의 두 번째 파티션이라면, 그것은 /dev/hda6가 될 것이다.

스왑 파티션을 사용하려면 Linux의 /etc/fstab 파일에 다음 줄을 추가해야 한다.

```
/dev/hda6        none             swap             sw               0       0
```

FreeBSD는 어떤 종류의 파티션이든지 스왑 공간으로 사용할 수 있지만, Linux는 스왑 파티션에서 특별한 표시(signature)를 필요로 한다. 그 표시는 mkswap을 사용하여 만든다. 그런데, FreeBSD가 그 공유되어지는 스왑 파티션을 사용할 때 이 표시를 없애 버리기 때문에, Linux로 부팅할 때마다 mkswap을 실행시켜야 할 것이다. 이것을 자동으로 하기 위해서는 부트할 때 swapon을 실행시키는 스크립트를 찾아야 한다. Red Hat Linux에서는 그 스크립트는 /etc/rc.d/rc.sysinit이다. 그 파일에서 swapon -a하는 부분 바로 앞에 다음 줄을 넣어주도록 하자:

```
awk -- '/swap/ && ($1 !~ /#/) { system("mkswap "$1"") }' /etc/fstab
```

이것은 매번 부팅할 때마다 /etc/fstab에 있는 주석처리(그 줄 첫 번째 칸에 "#"를 넣어주면 된다)되지 않은 스왑 파티션에 대하여 mkswap을 실행시킬 것이다.

다시 Linux로 리부팅했을 때 스왑 공간의 크기를 확인하기 위해서 free를 실행하도록 한다. 또한 모든 것이 원하는 대로 작동하는지 확인하기 위해서 FreeBSD로도 리부팅해 들어가야 한다. 만약 제대로 작동하지 않는다면, 아마도 스왑 공간으로서 잘못된 파티션을 사용했을 것이다. 그 문제에 대한 유일한 해결책은 FreeBSD를 재설치하고 다시 시도해 보는 것이다. 경험은 위대한 스승이다. :-)

제 4 절 LILO를 사용한 FreeBSD 부팅

LILO를 사용해서 FreeBSD를 쉽게 부팅할 수 있다. LILO를 사용하길 원한다면 FreeBSD 부트 선택기(boot selector)(`Booteasy`)를 설치하지 않도록 한다. 사용 중인 `/etc/lilo.conf` 파일에 다음 줄을 추가하고 `lilo`를 실행시킨다(FreeBSD 슬라이스가 `/dev/hda4`라고 가정한다):

```
other=/dev/hda4
        table=/dev/hda
        label=FreeBSD
```

만약 FreeBSD를 두 번째 SCSI 드라이브에 인스톨하였다면, 다음과 같은 내용을 사용한다(FreeBSD 슬라이스는 `/dev/sdb2`라고 가정한다):

```
other=/dev/sdb2
        table=/dev/sdb
        loader=/boot/chain.b
        label=FreeBSD
```

제 5 절 파일시스템 마운트

5.1 Linux에서의 UFS 파일시스템 마운트

아쉽게도 Linux 2.0.xx 커널에 들어있는 UFS 드라이버는 FreeBSD에 대한 지원을 포함하고 있지 않다. FreeBSD 파일시스템을 마운트하려고 시도하면, 어떤 에러 메세지만 나오게 된다(사실 파일시스템은 마운트되지만 마운트된 파일시스템을 가지고 할 수 있는 게 아무 것도 없다). 이 문제는 최신 Linux 커널(버전 2.1.87 이상)에서는 이미 해결이 된 상태다.

SunSite `<ftp://sunsite.unc.edu/pub/Linux/ALPHA/ufs/>`에는 Linux 2.0.xx 커널(xx <= 30)을 위한 UFS 드라이버의 다른 버전이 올라와 있다. 그것은 **U2FS**라고 불리는데 현재 버전은 `u2fs-0.4.3.tar.gz`이다. Linux 2.0.31 이상(2.0.xx; 2.1.xx는 지원하지 않는다) 버전을 위한 U2FS 버전(`ufs-0.4.4.tar.gz`)은 *이 사이트* `<http://www.mathi.uni-heidelberg.de/~flight/projects/u2fs/>`에서 U2FS(UFS도 포함)에 대한 더 많은 정보와 함께 얻을 수 있다.

이제 U2FS 파일시스템과 BSD 디스크 이름 붙이기(disklabel)를 지원할 수 있는 새로운 커널을 만들어야 한다. 이것에 관한 더 많은 정보을 얻기 위해서는 3.1(Linux 설치 및 준비) 절을 참고하도록 하자. U2FS를 사용할 때는 커널에서 **UFS filesystem support**를 삭제할 수 있다.

새로운 커널을 설치했을 때는, 다음 명령을 가지고 UFS 파일시스템들(스왑 파티션을 제외한 FreeBSD 슬라이스 내에 있는 모든 파티션들)을 마운트할 수 있다:

```
mount -t u2fs /dev/hda8 /mnt
```

Linux 커널 버전 2.1.87 이상을 사용하고 있다면 다음과 같은 명령을 사용해야 한다:

```
mount -t ufs /dev/hda8 /mnt
```

UFS (U2FS도 마찬가지) 드라이버는 읽기 전용이다. 즉; UFS 파일 시스템으로부터 읽어들일 수는 있지만 그곳에다 쓸 수는 없다. 실험적인 읽고 쓸 수 있는 UFS 드라이버가 Linux 커널 버전 2.1.112 이상에서는 읽기 전용 드라이버를 대체한 상태이다; FreeBSD 파티션에 쓰는 것은 버전 2.1.127부터 지원된다.

5.2 FreeBSD에서의 ext2fs 파일시스템 마운트

FreeBSD에서 ext2fs 파일시스템을 마운트하기 위해서는 먼저 ext2fs 지원 기능을 가지 새로운 커널을 만들어야 한다. 커널 컴파일하는 방법을 배우기 위해서는 *FreeBSD 핸드북* `<http://www.freebsd.org/handbook/>`을 참조하도록 한다. 새로운 커널 설정 파일에 다음 줄을 넣어야 한다.

```
options         "EXT2FS"
```

새로운 커널로 부팅을 했을 때는, 다음과 같은 명령으로 ext2fs 파일시스템을 마운트할 수 있다.

```
mount -t ext2fs /dev/wd0s3 /mnt
```

FreeBSD로부터는 확장파티션에 있는 ext2fs 파일시스템은 마운트할 수 없다는 사실에 주의하도록 한다.

FreeBSD 2.2.5 이하에 있던 버그로 인하여, 이 버젼대를 사용하고 있는 경우에는 FreeBSD를 셧다운(shutdown)하기 *전에* 모든 ext2fs 파일시스템을 언마운트해야 한다. 만약 ext2fs 파일시스템이 마운트된 상태에서 FreeBSD를 셧다운하게 되면, FreeBSD는 UFS 파일시스템을 동기화(sync)시키지 못한다. 결과적으로 다음 FreeBSD 부팅 때에 `fsck`가 돌아가게 된다.

제 6 절 외부 바이너리 실행

6.1 Linux에서의 FreeBSD 바이너리 실행

iBCS 패키지는 Linux 하에서의 FreeBSD 바이너리 실행에 대한 지원하고 있다; 그러나 그것은 오래되었으며 관리되어지지않고 있다. 필자 자신을 그것을 제대로 작동시킬 수 없었다. 이와 관련해서 필자보다 운이 좋았던 사람이 있다면 반드시 알려주기 바란다.

6.2 FreeBSD에서의 Linux 바이너리 실행

FreeBSD는 a.out과 ELF 형식으로 된 Linux 바이너리를 실행할 수 있다. 이것을 하기 위해서는 다음 세 단계를 거쳐야 한다:

1. Linux와의 호환이 가능하도록 해야 한다. (FreeBSD 2.2.2에서 — 다른 버젼에서는 세부적인 것들이 달라질 수 있다) 이것을 하기 위해서는 `/etc/rc.conf` 파일을 수정해서

   ```
   linux_enable="NO"
   ```

   을 다음과 같이

   ```
   linux_enable="YES"
   ```

   로 바꾸고 리부팅해야 한다. Linux 바이너리 지원을 로드하는 또다른 방법은 명령어 `/usr/bin/linux`을 실행하는 것이다. 이 방법으로 하면 리부팅할 필요가 없으며, 항상 Linux 바이너리 지원을 로드하는 것도 아니다(즉, 메모리를 절약할 수 있다). 새로운 FreeBSD 커널을 만든다면 FreeBSD 커널 설정(config) 파일에 다음 줄을 추가하는 것을 잊지 않도록 하자.

   ```
   options         COMPAT_LINUX
   ```

2. Linux 바이너리가 동적으로 링크되어있을(dynamically linked) 경우에는 Linux 공유 라이브러리를 설치해야 한다. 그 라이브러리들은 FreeBSD 2.2.{2,5,6}에서는 `linux_lib-2.4.tgz`라는 패키지로 포함되어 있다. 패키지를 설치하기 위해서는 다음 명령어를 실행한다:

```
pkg_add <path_to_package>/linux_lib-2.4.tgz
```

<path_to_package>는 패키지가 저장되어있는 디렉토리이다.

```
pkg_add ftp://ftp.freebsd.org/pub/FreeBSD/packages-stable/All/linux_lib-2.4.tgz
```

나 /stand/sysinstall를 재실행함으로써 네트웍으로 패키지를 설치할 수도 있다. /stand/sysinstall를 사용할 경우에는 "Configure", "Packages"를 선택한 다음 메뉴를 사용하면 된다. 정적으로 링크된(statically linked) Linux 바이너리를 실행하려고 할 때는 다음 명령러를 실행시켜야 한다:

```
brandelf -t Linux <name_of_statically_linked_linux_binary>
```

3. 실행하고자 하는 Linux 프로그램을 설치한다. 프로그램은 UFS나 ext2fs 파일시스템 어디에나 설치할 수 있다. FreeBSD에서 ext2fs 파일시스템을 사용하는 것에 대한 더 자세한 정보는 5.2(FreeBSD에서의 ext2fs 파일시스템 마운트) 절을 참고하기 바란다.

필자는 이 방법을 사용해서 FreeBSD 2.2.2에서 Applixware 4.3과 Netscape 3.01의 Linux버젼(둘 다 ELF 형식)을 성공적으로 실행시켰다(물론, 필자도 Netscape 4의 FreeBSD 전용 버젼이 있다는 것을 알고 있다). acroread와 StarOffice도 FreeBSD 하에서 잘 작동한다. 이 주제에 대한 더 많은 정보를 얻기 위해서는 FreeBSD 문서를 읽도록 하자.

제 7 절 참고 자료

이 미니하우투(mini-HOWTO)의 최신버젼은 (SGML과 포스트스크립트를 포함한) 여러 가지 형태로 *이 사이트* `<http://www.image.dk/~nkbj/>`에서 다운로드할 수 있다. *JF 프로젝트* `<ftp://jf.linux.or.jp/pub/JF/other-formats/>`의 일환으로 Teruyoshi Fujiwara 씨가 일본어로 이 문서를 번역하였다.

Gregor Hoffleit은 U2FS와 UFS의 개발에 대한 정보를 가지고 *사이트* `<http://www.mathi.uni-heidelberg.de/~flight/projects/u2fs/>`를 운영하고 있다.

Annelise Anderson은 이 미니-하우투(mini-HOWTO)와 같은 (그리고 더 많은 것들을 포함한) 주제를 다루는 *Linux+FreeBSD FAQ* `<http://freebsd.stanford.edu/FreeBSD/linux.html>` 를 작업하고 있다. 그 FAQ는 freebsd-doc 프로젝트의 일부가 될 예정이다.

이 사이트 `<http://www.freebsd.org/>`에서 FreeBSD에 대한 더 많은 정보를 찾을 수 있으며 전체 시스템을 다운로드할 수도 있다. 또한 *Walnut Creek CDROM* `<http://www.cdrom.com>`에서 CDROM의 형태로 구입할 수도 있다(그 곳의 서버도 FreeBSD를 사용하고 있다).

Linux 커널 하우투(HOWTO)는 (이 미니-하우투도 함께) *The Linux Documentation Project* `<http://sunsite.unc.edu/LDP/>`의 일환으로서 배포되어진다.

제 8 절 감사의 말과 저작권

FreeBSD 초보자의 질문에 대한 답변을 해준 *덴마크의 *BSD 사용자 그룹* `<http://hotel.prosa.dk/bsd-dk/>`, U2FS의 존재를 알려준 Takeshi Okazaki 씨, 중요한 조언을 해준 David O'Brien 씨에게 감사를 드린다.

8.1 저작권