HARD DISK UPGRADE mini HOWTO

Yves Bellefeuille, *yan@ottawa.com* v0.1, 7 September 1998

번역: 고광선 *rilla91@klnet.co.kr*, 편집: 엄기성 *jurist@netkorea.co.kr*

리눅스 시스템을 하드디스크에서 다른 하드디스크로 복사하는 방법에 대해서 설명했다.

목 차

제 1 절 소개

최근에 249M짜리 하드디스크를 더 큰 용량의 하드디스크로 교체한 후, **LILO**를 포함한 리눅스 전체 시스템을 새로운 디스크로 옮기길 원했다. 이후에 기술된 내용은 그 방법에 관한 내용이다.

다음 설명에서,***/dev/hda***는 예전 하드디스크, ***/dev/hda1***은 예전 리눅스 파티션을 나타내고, ***/dev/hdb***는 새로운 하드디스크, ***/dev/hdb1***은 새로운 하드디스크의 리눅스 파티션을 의미한다.

따라서, 당신이 취향에 따라서 바뀔 수도 있겠지만, 하드디스크의 첫번째 파티션을 리눅스 파티션이라 생각하자.

이 문서에 나와있는 내용들은 **Red Hat 4.2**로 운영되는 본인의 시스템을 기준으로 테스트를 하였고, 또한 **Debian 1.3.1**과 **Slackware 3.3**에서도 테스트를 하였다. 만일 각각의 배포본간에 차이가 있을 경우에는 차이점에 대해서 명시를 하겠다.

만일 기술되어 있는 명령어들이 당신의 시스템에 정확하게 동작하지 않는다면, 당신이 사용하고 있는 리눅스의 버전을 본인에게 알려주길 바란다.

제 2 절 시스템에 두 개의 하드디스크 설치하기

현재의 컴퓨터 시스템들은 하드디스크 콘트롤러에 4개의 EIDE 방식의 장비를 연결할 수 있기 때문에 여러분의 시스템에 두 개의 하드디스크를 연결하는 것에는 커다란 문제가 없을 것이다. 일반적으로 하드디스크와 CD-ROM은 EIDE 장비에 해당하며, 플로피 드라이브와 테이프 드라이브는 하드디스크 콘트롤러가 아닌 플로피 콘트롤러에 연결된다.

SCSI 어댑터는 보다 유연성이 있어서 7개의 장비를 연결할 수 있다. 이미 여러분들중에 SCSI 어댑터를 가지고 있다면, 어떠한 장비가 SCSI 어댑터에 연결되어 있는지에 대한 사실을 이미 알고 있었을 것이다.(보다 자세한 내용은 SCSI HOWTO를 참조)

당신이 만일 구형 시스템을 가지고 있다하더라도 구형 시스템 하드디스크 콘트롤러에 2개의 하드디스크를 연결할 수 있으므로, 하드디스크를 2개 동시에 연결해서 사용하는 데에는 문제가 없다. 만일, 하드디스크 하나에 다른 장비를 이미 연결해서 사용하고 있는 상태라면 (예를 들면, 이미 다른 하드디스크를 사용중이거나, CD-ROM을 연결해서 사용중일 경우) 당신은 추가로 하드디스크를 연결할 수 없기 때문에 기존에 연결되어 있는 장비를 제거한 후 연결해야 한다.

연결시에는 하드디스크의 "master"와 "slave"를 구분해서 연결을 한다. 만일 구분을 할 수 없을 경우에는 manual이나 디스크 제조업체에 연락을 해야 할 것이다.

제 3 절 리눅스 이외의 OS가 사용하는 파티션 Unmount 시키기

많은 사람들이 DOS, Windows, OS/2와 같은 OS를 리눅스 시스템에서 같이 사용하고 있다. 이 와 같이 리눅스 이외의 OS 파티션은 해당 OS상에서 만들어주어야 한다. 그리고 당신의 리눅 스 파티션을 복사하기 전에 반드시 Unmount시켜야 한다는 사실을 명심하기 바란다. 예를 들 면 여러분의 DOS파티션이 /dos에 마운트 되어 있다면 다음과 같은 명령어로 Unmount해야 한 다.

```
# umount /dos
```

주의 : umount에서 두번째 글자인 n이 빠져있다는 사실에 주의하기 바란다.

제 4 절 새로운 디스크에 파티션 만들기

새로운 디스크에 파티션을 만들기 위해서 다음과 같은 명령어를 사용한다

```
fdisk /dev/fdb
```

파티션을 만드는 방법에 대해서 더 많은 정보를 원한다면 Installation HOWTO나 Partitioning Mini-HOWTO를 보기 바란다

만일 새로운 디스크가 1024이상의 실린더를 가지고 있다면 Large Disk Mini-HOWTO를 보기 바란다. 정리하면, 리눅스를 부팅하기 위해서는 모든 파일들을 첫번째 1024실린더 안에 모 든 파일을 설치해야 한다. 이를 위한 한가지 방법으로 1M에서 2M의 용량에 해당하는 조그마 한 파티션을 만들고 ***/boot***디렉토리를 설치한다(RedHat 기준, Slackware에서는 커널이 ***/boot/vmlinuz***보다는 ***/vmlinuz***에 위치하기 때문에 이 조그마한 파티션에 / 디렉토리 를 넣어야 한다.)

리눅스를 제외한 다른 시스템에서는 해당 OS에서 제공하는 **fdisk** 명령을 사용해서 파티션을 만든다.

제 5 절 새로운 디스크 포맷하기

새로운 디스크를 포맷하기 위해서는 다음과 같은 명령어를 사용한다

```
# mkfs.ext2 /dev/hdb1
```

디스크의 물리적 bad sector를 검사하기 위해서 ***/dev/hdb1***앞에 **-c** 옵션을 주면된다.

(주의 : man 페이지에 설명되어 있는 것과는 다르게, **mkfs -t ext2 -c /dev/hdb1**이라는 명령은 RedHat, Debian 그리고 Slackware상에서 Bad block을 검사하지 않는다)

제 6 절 새로운 디스크 마운트하기

새로운 디스크를 마운트하고자 원하는 곳의 디렉토리를 만든다. ***/new-disk***에 마운트한다 예를 들면

```
# mkdir /new-disk
# mount -t ext2 /dev/hdb1 /new-disk
```

제 7 절 예전 디스크에서 새로운 디스크로 파일 복사하기

아마도 여러분은 링크파일을 포함한 디스크의 구조전체를 복사하기 원할 것이다.

그러나, ***/new-disk*** 디렉토리는 새로운 디스크 자체이므로 복사를 해서는 안된다.

게다가, 새로운 디스크에 ***/proc*** 디렉토리를 만들기를 원하지만, 그 내용을 복사할 수 없을 것 이다. ***/proc***에 들어 있는 파일들은 실질적인 파일들이 아니라 시스템상에서 돌아가는 프로세 스들에 대한 정보를 가지고 있는 **virtual**시스템 파일이기 때문이다.

여기에는 새로운 디스크로 파일들을 복사하기 위한 4가지의 다른 방법들이 있다. 만일 당신이 용량이 큰 디스크와 적은 메모리를 사용하고 있는 상태라면 시간이 꽤 걸릴 것이며, 약 초당 10M에 해당하는 용량의 파일들이 복사된다고 생각하면 될 것이다.

다른 가상 터미널에서 `df`라는 명령어를 통해서 복사 진행상황을 확인해 볼 수 있다. 또 다른 방법으로는 `watch df`나 `watch ls -l /new-disk`라는 명령으로 2초마다 업데이트되는 정보를 볼 수 있을 것이다. 단 `watch`라는 명령이 시스템에 부하를 주게 되어 복사속도를 떨어뜨릴 수 있다는 사실에 대해서 명심하기 바란다.

7.1 첫번째 방법

```
cp -ax / /new-disk
```

이 방법은 매우 쉽다. 그러나 이러한 방법은 복사하고자 하는 리눅스 시스템이 하나의 디스크 파티션으로 되어 있는 경우에 한해서 제대로 작동할 것이다. -a 옵션은 preserves the original system as much as possible. -x 옵션은 cp명령어를 단일 파일시스템으로 제한한다. 이러한 옵션의 선택은 ***/new-disk***와 ***/proc***디렉토리를 복사하지 않도록 한다.

7.2 두번째 방법

```
cd / && cp -a `/bin/ls -1A | egrep -v "^new-disk$|^proc$"` /new-disk
```

루트 디렉토리로 이동한 후 ***/new-disk***와 ***/proc***를 제외한 모든 파일과 디렉토리를 /new-disk로 복사를 한다. 여기서 ls다음에 오는 첫번째 옵션은 영문자 'L'이 아닌 아 라비아 숫자 1을 뜻한다.

7.3 세번째 방법

```
(cd / && tar cpf - . --exclude new-disk --exclude proc) | (cd /new-disk && tar xpf -)
```

먼저 루트 디렉토리로 이동한 후 ***/new-disk***와 ***/proc***를 제외한 모든 것들을 tar시킨 후 /new-disk로 이동하여 untar를 실시한다. 여기서 –exclude 뒤에 오는 디렉토리 이름들 앞에서 뒤에는 /가 들어가서는 안된다.

주의 : tar의 -l 옵션이 사용되지 않았는데, 이것은 ***/new-disk***와 ***/proc***의 내용을 복 사하지는 않지만, 디렉토리는 다시 만들기 때문이다, 따라서, tar의 -l 옵션은 cp 의 -x 옵션과 다르게 작동한다.

7.4 네번째 방법

```
# cp -a /bin /boot /dev /etc /home /lib /lost+found /mnt /root /sbin /tmp /usr /var /new-disk
```

이 명령에서 마지막에 있는 ***/new-disk***디렉토리가 대상 디렉토리가 된다. 따라서 루트 에 있는 모든 디렉토리를 ***/new-disk***로 복사를 하겠다는 뜻이다.

이와 같은 방법은 당신이 가지고 있는 리눅스 시스템 디렉토리들을 ***/new-disk***와 ***/proc*** 를 제외한 나머지 디렉토리를 루트를 기준으로 나열을 하기만 하면 된다. 만일 다른 때문에 다른 명령을 사용할 수 없다면 복사하기 원하는 디렉토리들을 임의적으로 나열해서 이 명령 을 사용할 수 있다.

만일 이 방법을 사용한다면, 루트 디렉토리 밑에 있는 다른 파일들을 다른 명령을 이용해서 복사해야 한다. 특히나, Debian과 Slackware를 사용하고 있는 시스템에서는 루트디렉토리 밑에 파일들이 존재하므로 다음과 같은 명령이 반드시 필요하다

```
# cp -dp /* /.* /new-disk
```

위에서 나열한 4가지 방법들을 사용한 후, 새로운 디스크에 ***/proc디***렉토리를 반드시 만들어 야 한다.

```
# umount /new-disk
# fsck.ext2 -f /dev/hdb1
# mount -t ext2 /dev/hdb1 /new-disk
```

만일 파일들이 제대로 복사가 되었는지 확인하고자 한다면 다음에 있는 스크립트를 사용해도 된다.

```
#!/bin/sh
cd /
for file in `/bin/ls -1A | egrep -v '^new-disk$|^proc$'`
do
    find $file -xtype f -exec cmp \{\} /new-disk/\{\} \;
done
```

(단 Slackware : 기본적인 Slackware 파일들만을 설치하였을 경우에는 `cmp`혹은 `diff`명령어 는 설치되지 않는다.)

제 8 절 */etc/fstab*을 적절하게 변경하기

만일 당신의 새로운 디스크가 예전 디스크와는 다르게 파티션구조를 가지고 있다면, ***/etc/fstab***파일을 새로운 디스크에 맞도록 적절하게 변경해야 한다. 기억해둘 것은 해당파일 이 ***/new-disk/etc/fstab***에 있다는 사실이다.(***/etc/fstab/***은 예전 디스크에 해당함)

제 9 절 새로운 디스크로 부팅하기 위해 LILO 준비하기

이 단계는 가장 복잡한 과정이다. 가장 일반적인 상황이라고 생각하는 LILO가 **Master Boot Record(MBR)** 에 설치되어 있다는 가정하에서 시작하도록 하겠다.

당신은 LILO를 2번째의 새로운 디스크에 설치하고자 한다. 확실한 사실 하나는 LILO는 두번 째 하드디스크에서는 실행할 수 없다는 것이다. 그러나, LILO 관련 문서를 살펴보게 되면 첫 번째 하드디스크를 제거했을 경우와 같은 상황에서 두번째 하드디스크에 LILO를 설치할 수 있 는 방법에 대해 나와 있다.

> LILO는 다음에 나와있는 어떠한 곳에 store 할 수 없다. 두번째 하드디스크(백업을 위해서가 아니거나, 현재의 첫번째 디스크를 제거하거나 사용할 수 없을 때거나 혹은 다른 boot loader이 사용중일 경우 다른 장치로부터 boot sector을 loading 할 수 있는 가능성이 있다

그러나 이 문서에서는 첫번째 하드디스크를 제거하였을 경우, 두번째 하드디스크에 LILO를 설치하는 적당한 방법에 대해서는 설명을 해주지 못하고 있다. 결국, 여러번의 시도와 함께 두 번째 하드디스크의 MBR에 LILO를 설치하고 작동하는데 있어서 가능하지 않다라는 결론에 도달 하게 되었다.

대신에, 처음으로 새로운 하드디스크로 부팅하는 순간에 부팅디스켓을 이용하는 방법을 제안한 다.

공디스켓을 집어넣고, 포맷을 실시한 후 파일시스템을 만들어 마운트를 시킨다.

```
# fdformat /dev/fd0H1440
# mkfs.ext2 /dev/fd0
# mount -t ext2 /dev/fd0 /mnt
```

(Debian의 경우 : `fdformat`이라는 명령어가 기본설치에 포함되지 않았다. 포맷이 이미 되어 있다면, `mkfs.ext2`라는 명령에 -c옵션을 주어 검사한 후 마운트를 해서 사용해도 상관이 없다)

(Debian과 Slackware의 경우 : fd0H1440 가운데에 있는 'H' 대신에 'h'를 사용해야 한다)

/boot에 있는 모든 파일을 디스켓으로 복사한다

```
# cp -dp /boot/* /mnt
```

다음과 같이 /mnt/lilo.conf파일을 새로 만든다.

```
boot=/dev/fd0           #Install LILO on floppy disk.
map=/mnt/map            #Location of "map file"
install=/mnt/boot.b     #File to copy to floppy's boot sector
prompt                  #Have LILO show "LILO boot:" prompt
timeout-50              #Boot default system after 5 seconds
                        #(Value is in tenths of seconds.)
```

```
image=/mnt/vmlinuz      #Location of Linux kernel on floppy
     label=linux        #Label for Linux system
     root=/dev/hda1     #Location of root partition on new hard
                        #disk. Modify this as appropriate for your system
                        #Note that you must use the name of the future location
                        #Once the old disk has been removed
```

(Debian의 경우 : "image"줄에서, 리눅스 커널의 실제적인 이름을 사용해야 한다, 예를 들어 Debian 1.3.1을 사용하고 있다면, '/mnt/vmlinuz-2.0.29'라고 사용한다)

부팅디스켓에 LILO를 설치하기 위해서

```
/sbin/lilo -C /mnt/lilo.conf
```

여기에서 '-C'옵션은 `/sbin/lilo`에게 어떠한 configuration 파일을 사용하는지 알려주는 옵션 이다.

디스켓을 언마운트 하기 위해서

```
# umount /mnt
```

를 실시한후 시스템을 끈다.

제 10 절 예전의 디스크 제거하기

예전의 디스크를 제거하고 새로운 디스크의 점퍼를 고쳐준 후 **BIOS**상에서 수정된 사항을 고친 다.

제 11 절 새로운 시스템을 재부팅한 후 새로운 디스크에 LILO 설치하기

당신이 이미 만들어 놓은 부팅디스켓을 이용하여 시스템을 재실행시키기 전에 BIOS상에서 부팅 순서를 'a: c:'의 순서로 수정하였는지 확인한다.

/etc/lilo.conf 파일에 필요한 수정작업을 실시하고 새로운 디스크에 LILO를 설치하기 위해서 ***/sbin/lilo***를 다시 실행시킨다. Debian에서는 당신이 LILO를 MBR에 설치하였다면 ***/etc/lilo.conf*** 상의 boot 라인상에서 ***/dev/hda1*** 보다는 ***/dev/hda***을 사용해야 한다는 사실을 명 심하기 바란다.

마지막으로 모든 것이 제대로 작동하는지 확인하기 위해서 새로운 하드디스크로 시스템을 재시 동한다. 만일 새로운 하드디스크로 작동하는데 문제가 있다면, 당신이 가지고 있는 부팅디스켓 을 이용할 수 있다는 사실을 염두해 두길 바란다.

제 12 절 Acknowlegdements

Special thanks to _Dr Konrad Hinsen_ of the Institute de biologie structurale, Grenoble, France, who has been kindly acting as my personal Linux guru. Thanks also to _Frank Damgaard_, _Paul Koning_ and _josh Rabinowitz_, and to _Scott Christensen_ for alerting me to some particularities of the Slackware distribution.