LILO mini-HOWTO

Cameron Spitzer, *cls@truffula.sj.ca.us* `<mailto:cls@truffula.sj.ca.us>` 1996년 12월 12일

한국어 번역 : 이 철 우 <*wowlee@nownuri.nowcom.co.kr* `<mailto:wowlee@nownuri.nowcom.co.kr>`>

이 파일은 몇가지의 전형적인 리로 설치법을 설명한다. 이 파일은 리로 사용자 가이드를 보충해주는 것으로 사용하도록 의도되었다. 비록 당신의 설정 상황이 이것과 같지는 않더라도 나는 이 예제들이 유익할것이라 생각한다. 이 파일이 당신의 어려움을 덜어주기를 희망한다.

목 차

제 1 절 예제 1

나는 2개의 IDE 드라이브와 하나의 SCSI 드라이브를 가지고 있다. SCSI 드라이브는 바이오스에서 인식할 수 없다. 리눅스 로더, 리로(LILO), 는 바이오스 호출(BIOS call)을 사용하고 오직 바이오스에서 인식하는 드라이브만 인식할 수 있다. 나의 멍청한 AMI 바이오스는 단지 A 드라이브 또는 C 드라이브 에서만 부팅이 된다. 나의 루트 파일 시스템은 SCSI 드라이브의 파티션에 존재한다.

해답: 커널, 맵 파일, 그리고 리눅스 파티션에서의 체인로더를 첫 번째 IDE 드라이브에 저장한다. 당신의 루트 파티션위에 커널이 반드시 있어야할 필요는 없다는걸 주목하라. 첫 번째 IDE 드라이브(/dev/hda2)의 두 번째 파티션은 /u2 디렉토리에 마운트 되어 있다. 여기 내가 사용하는 /etc/lilo.conf 파일 이다.

```
#  첫 번째 IDE 드라이브의 마스터 부트 레코드(MBR)에
#  리로 부트 레코드를 설치한다.
#
boot = /dev/hda
#  /sbin/lilo 가 다음의 파일로부터 리로 부트 레코드를
#  마스터 부트 레코드(MBR)의 위치에복사한다.
install = /u2/etc/lilo/boot.b
#
#  나는 verbose 부트 메뉴를 썼다. 리로가 이곳에서 이것을 찾는다.
message = /u2/etc/lilo/message
#  설치 프로그램이 다음 파일을 만들것이다.
#  이것은 부트-로더에 어디에 커널의 블럭이 있는가를 알려준다.
map = /u2/etc/lilo/map
compact
prompt
#  10초를 기다리면 기본값으로 1.2.1 커널을 부팅하다.
timeout = 100
#  다음과 같이 함으로써 바이오스가 이것을 인식할 수 있는 곳에 커널이 저장된다.
```

```
#      cp -p /usr/src/linux/arch/i386/boot/zImage /u2/z1.2.1
image = /u2/z1.2.1
        label = 1.2.1
#  리로가 커널에게 첫 번째 SCSI 파티션을 루트로서 마운트하는 것을 알려준다
#  바이오스는 이것을 인식할 수 없다.
        root = /dev/sda1
#  이 파티션이 검사되고 /etc/rc.d/rc.S에 의해 다시 마운트 될것이다.
        read-only
#  같은 커널이 루트 파티션 대신에 다른 파티션을 마운트하도록 할 수 있다.
#  나는 비상용 루트 셋업을 /u2에 가지고 있다.
image = /u2/z1.2.1
        label = 1.2.1-hda2
        root = /dev/hda2
        read-only
#  나는 새로운 커널이 작동하지 않을경우를 대비하여 구버젼의
#  슬랙웨어 커널을 보관해 놓았다. 나는 실제로 한 번 이것이 필요했다.
image = /u2/z1.0.9
        label = 1.0.9
        root = /dev/sda1
        read-only
#  나의 DR-DOS 6 파티션.
other = /dev/hda1
        loader=/u2/etc/lilo/chain.b
        label = dos
        alias = m
```

제 2 절 예제 2

내 사무실에 있는 시스템은 1GB IDE 드라이브를 사용한다. 바이오스는 단지 IDE 드라이브의 504 MB 만을 인식한다. (MB는 10**6 아니라, 2**10 바이트를 의미한다.) 그래서 나는 350메가는 도스용 파티션(/dev/hda1) 그리고 120메가는 나의 리눅스 루트 파티션(/dev/hda2)으로 사용한다. MS-DOS는 드라이브가 새것일 때 정확하게 도스 자신을 설치하지 못한다. 노벨 도스 7도 같은 문제점을 가지고 있다. 운이 좋게도, "Options by IBM" 은 드라이브와 함께 박스안에 "OnTrack" 디스?을 넣는 것을 잊었다. 그 드라이브는 "OnTrack Disk Manager." 라 불리우는 제품과 함께 딸려 올것이다. 만약 당신이 단지 도스만을 가지고 있다면, 당신은 이걸 사용해야 할거라고 생각한다. 그래서 나는 리눅스의 fdisk로 파티션을 만들었다. 도스 6.2는 다음과 같은 메시지를 남기면서 /dev/hda1에 설치되지 않았다.

'this release of MS-DOS is for new installations. Your computer already has MS-DOS [위의 메시지는 틀렸다, 그 드라이브는 새것 이었다] so you need to get an upgrade release from your dealer.'

그래서 나는 리눅스 fdisk를 다시 실행시키고 테이블에서 1번 파티션을 지웠다. 이것이 내가 조금전에 지우고 도스를 설치했던 1번 파티션과 동일한 파티션을 생성시켰던 도스 6.2를 만족시켰다. 도스 6.2는 그 드라이브에 마스터 부트 레코드를 썼다. 하지만 이것은 부팅할수 없었다. 운이 좋게도 나는 플로피에 슬랙웨어 커널(슬랙웨어 설치 프로그램인 "setup"에 의해 만들어진)이 있다. 그래서 나는 리눅스를 부팅하고 도스의 손상된 마스터 부트 레코드(MBR)에 리로(LILO)를 썼다. 이것은 동작을 한다. 여기 내가 사용하는 /etc/lilo.conf 파일 이다.

```
boot = /dev/hda
map = /lilo-map
delay = 100
```

```
ramdisk = 0           # 슬랙웨어 커널에서 램디스크 기능을 중지시킨다.
timeout = 100
prompt
disk = /dev/hda         # 바이오스(BIOS)는 오직 첫 번째 500메가만을 인식한다.
   bios=0x80            # 첫번째 IDE 드라이브를 명시한다.
   sectors = 63         # 이 숫자들은 당신의 드라이브 설명서에서 얻는다.
   heads = 16
   cylinders = 2100
image = /vmlinuz
  append = "hd=2100,16,63"  # 이건 아마 중복될 것이다.
  root = /dev/hda2
  label = linux
  read-only
  vga = extended
other = /dev/hda1
  label = msdos
  table = /dev/hda
  loader = /boot/chain.b
## end
```

내가 이 시스템을 설치한 후에, "스태커" 또는 "더블스페이스"를 사용하지 않은한 도스의 파일 시스템을 사용할 수 있는 zImage, boot.b, map, chain.b, 그리고 message files가 있는 파티션을 조사했다. 그래서 나는 /dev/hda1 의 500 MB에 도스 파티션을 만들 수 있었다. 나는 또한 "OnTrack"이 파티션 테이블의 처음 몇십 바이트를 드라이브에 기록했을 것이라는 것과 리눅스 IDE 드라이버를 이 문제를 해결하도록 해킹하는 것이 가능하다는 것을 배울 수 있었다. 하지만 미리 컴파일되어있는 슬랙웨어 커널을 가지고도 설치는 불가능 했을것이다. 결국, IBM에서 나에게 "OnTrack" 디스ㅋ을 보냈다. 나는 OnTrack 의 기술 지원부서에 전화를 했다. 그들은 리눅스가 깨졌다고 나에게 말했다. 왜냐하면 리눅스는 바이오스를 사용하지 않기 때문이다. 나는 그 디스크을 버렸다.

제 3 절 예제 3

나는 사무실의 시스템에 윈도우-95를 설치하였다. 그런데 윈도우-95가 리로 마스터 부트 레코드(LILO MBR)를 날려버렸다. 그런데 윈도우-95는 나의 리눅스 파티션만을 혼자 남겨두었다. 플로피로부터 커널을 적재하는 것은 오랜 시간이 걸린다. 그래서 나는 IDE 드라이브로부터 커널을 부팅할 수 있는, 플로피에 리로 셋업을하는 플로피 디스크를 만들었다. 나는 리로 플로피 디스크를 다음과 같이 만들었다:

```
  fdformat /dev/fd0H1440   #  빈 디스?에 트랙을 만든다.
  mkfs /dev/fd0 1440       # 미닉스 파일 시스템을 만든다.
  mkdir /3                 # 미닉스 파일 시스템을 마운트할 디렉토리를 만든다.
  mount /dev/fd0 /3
  cp -p /boot/chain.b /3    #  채인로더를 복사한다.
  lilo -C /etc/lilo.flop       #  디스?에 리로와 맵을 설치한다.
  umount /3
```

디스ㅋ은 당신이 설치 프로그램을 실행시킬 때 반드시 마운트되어야 한다는걸 주목해라. 그래야 리로가 맵을 적절히 쓸 수 있다. 이 파일은 /etc/lilo.flop 이다. 이것은 마지막것과 거의 똑같다:

```
#  하드 디스크로부터 커널을 부팅할수 있는 플로피를 만든다.
boot = /dev/fd0
```

```
map = /3/lilo-map
delay = 100
ramdisk = 0      # 슬랙웨어커널에서 램 디스크 기능을 중지시킨다.
timeout = 100
prompt
disk = /dev/hda  # 1기가 IDE 드라이브, 바이오스는 단지 처음 500메가만을 인식한다.
  bios=0x80
  sectors = 63
  heads = 16
  cylinders = 2100
image = /vmlinuz
  append = "hd=2100,16,63"
  root = /dev/hda2
  label = linux
  read-only
  vga = extended
other = /dev/hda1
  label = msdos
  table = /dev/hda
  loader = /3/chain.b
## 끝
```

마지막으로, 나는 사무실의 컴퓨터에 도스 6.2를 필요로한다. 하지만 첫 번째 드라이브는 건드리고 싶지 않았다. 나는 SCSI 콘트롤러와 드라이브를 추가하고, 리눅스의 mkdosfs 명령어로 그 드라이브에 도스 파일 시스템을 만들었는데, 윈도우 95는 이것을 D 드라이브로 인식한다. 그러나 당연히 도스는 D 드라이브에서 부팅되지 않을것이다. 이것은 당신이 리로를 가지고 있다면 문제가 되지 않는다. 나는 예제 2.에서의 lilo.conf에 다음의 내용을 추가했다.

```
other = /dev/sda1
  label = d6.2
  table = /dev/sda
  loader = /boot/any_d.b
```

도스 6.2가 실행되고, 도스는 자신이 C 드라이브에 있고 윈도우 95는 D 드라이브에 있다고 생각한다.