ZIP-drive mini-HOWTO

Grant Guenther, grant@torque.net

번역: 정재익 advances@nownuri.net

다음 문서는 ZIP-Drive mini HOWTO를 번역한 것으로 약간의 사견이 같이 가미되어 있다. (물론 기술적인 면은 전혀 사견을 첨삭하지 않았다.) 그리고 이 문서는 ZIP drive를 LINUX에 설치하는 과정에 관한 전반적인 사항을 기술한 것으로 기술적인 면은 전혀 언급되어 있지 않다. 그리고 반드시 SCSI HOWTO와 같이 참고할 것을 권한다. 작으나마 도움이 되길 바란다. 만일 틀린 부분이 있다면 본인에게 메일을 주기 바란다. 이 번역문서의 사용은 GPL에 준한다. IOMEGA ZIP drive는 널리 사용되고 있는 removable media disk drive이다. 이것은 SCSI와 parallel` ort의 두 가지의 주된 version이 있다. 이 문서는 LINUX에서 ZIP Drive를 어떻게 사용하는지를 기술하고 있다. 이것은 SCSI HOWTO와 같이 읽어야만 한다.

목 차

제 1 절 정보

Iomega ZIP drive는 싼가격, low-performance를 특징으로하는 싼 가격의 휴대하기 간단한 removal media disk drive이다. 이것은 96MB의 용량을 가지는 disk이고 드라이브는 SCSI와 parallel의 두가지 버젼을 이용할 수 있다. parallel port version은 실재로는 SCSI device와 같으며, IOMEGA사의 PPA-3 parallel-to-SCSI protocol을 사용하여 설정을 한다.

이 문서는 리눅스에서 ZIP drive를 어떻게 사용하는가를 기술한다. 이것은 SCSI device이기 때문에 여러분은 SCSI HOTO를 같이 읽는 것이 아주 중요하다. SCSI에 관한 문서는 LINUX SCSI system에 관한 전반적인 정보를 준다.

이 문서는 다른 사람들에 의해 수집되고 발행된 정보들을 포함하고 있는데, 특히:

- Scot Wilcoxon, `sewilco@fielday.mn.org`
- Joe Mack, `mack@ncifcrf.gov`
- Byron Jeff, `byron@cc.gatech.edu`

나는 그들의 아주 중요한 기여에 감사를 표하며, 내가 인용한 것의 어떤 error에도 책임을 지겠다.

제 2 절 ZIP 드라이브

3가지 version의 Iomega ZIP 100 drive가 있다. 그들은 모두 *100 Mbytes* 를 저장할수 있는 3.5" FD를 닮은 특수한 cartridges를 사용한다. 이 디스크는 실재로는 96cyl-2048sec-512byte 를 저장한다. 이것은 정상적으로는 96 M bytes라고 해야한다.

첫번째 version 은 SCSI interface를 갖는 절반높이의 5.25" 내장형 drive이다. 다른 2가지는 작은 파란색의 가벼운 plastic case를 갖는 외장형 drive로서 외부 전원을 받는다. 이 외장형은 SCSI version과 parallel port version이 있다.

모든 drive는 전면에 큰 push button을 가지고 있다. 이것은 disk를 eject시키는데 사용된다.

LINUX는 drive를 사용하는 동안에는 drive의 문을 잠궈 놓는다. 하지만 door가 잠궈져 있는 동안에 button이 눌러지면 ZIP drive는 그것을 기억하고서 software unlock를 시키면서 곧바로 disk를 eject 시킨다.

2.1 스카시용

외장형 SCSI version은 drive의 후면에 2개의 DB25F 연결 단자와 2개의 configuration switch를 가진다. 그중 하나는 drive의 target address를 선택하는 것이다 : 실재로 선택은 5 와 6으로 한정이 되어 있다. (스카시 ID를

결정하는 것이다.) 다른 하나는 drive가 연결된 chain에서 마지막일 경우에 internal terminator를 활성화 시키는 것이다. 25 pin SCSI connector는 친숙한 매킨토시 형의 배선을 사용한다.

만약 당신이 Centronix 또는 high-density connector를 가진 host adapter를 사용할 경우에 (대표적으로 AHA1542CF 가 이 형의 connector를 필요로 합니다.) 그리고 만약 드라이브에 Mcintosh type cable이 들어 있다면 standard cable과 converter를 쉽게 구할수 있다. (이 이야기는 만일 여러분이 가진 SCSI host adapter가 centronix high-density connector를 사용하는 것이라면, ZIP drive는 25pin connector를 사용하므로 이 둘 사이를 변환시켜줄 cable이 필요로 한다는 것이다. 이것은 용산상가에서 약 2만원정도에 구할 수 있다.)

나는 내장형 SCSI drive를 보지는 못했다. 하지만 나는 이것은 표준 50pin DIP header connector와 똑 같은 두개의 스위치를 가질것이라고 생각한다.

당신이 선택한 target address(SCSI id)가 다른 SCSI 장치와 충돌하지 않는 다는 것을 확인하도록 하십시오. 그리고 SCSI에 연결된 마지막 장치의 terminator가 enable되어 있는지 확인하도록 하십시오. 또는 external terminator가 설정되어 있는지 확인하도록 하십시오.

만약 당신이 internal SCSI disk나 CD-ROM 을 가진다면 그리고 ZIP drive를 이미 존재하는 adapter에 연결을 한다면 당신은 카드에 있는 어떻한 terminator도 이미 제거 되어 있다는 것을 check하여야만 합니다. SCSI bus의 양쪽 말단에서만 terminator가 확성화 되어 있어야만 합니다. 만약 당신의 bus가 일부는 내장형, 일부는 외장형이라면 terminator는 외장형의 말단 장치와 내장형의 마지막 장치에 설정되어야 한다. 그러나 adapter card 자체에는 terminator가 없다.

모든 cable이 견고히 부착되어 있는지 확인을 하라.

2.2 ZIP Zoom 호스트 어뎁터

IOMEGA는 *ZIP Zoom* 이라는 이름의 SCSI host adapter를 팔고 있다. 이것은 근본적으로 *Adaptec 1520* 계열의 adapter이다. 이것은 Macintosh type DB25F connector를 가지고 ZIP drive와 같이 오는 cable 과 호환된다.

LINUX 는 이 host adapter를 `aha152x` driver 로서 지원한다.

2.3 패러럴용

Parallel port ZIP drive는 역시 두개의 DB25 connector를 가진다 이것은 숫놈은(DB25M)은 제공되는 parallel cable로서 여러분의 컴퓨터의 parallel port로 연결시킨다. 다른 것은 (female, DB25F) printer를 지원하기 위한 것이다. LINUX는 현재 ZIP drive와 연결된 printer를 동시에 지원하지는 않는다. 대략적인 작업은 modules을 이용해서 같이 사용할 수는 있지만 둘을 동시에 사용할 수는 없다. 이에 대한 configuration switch는 없다.

Parallel port ZIP drive는 몇가지형태의 병렬포트와 호환이 된다, 그러나 현재의 Linux driver 는 표준형과 양방향성 병렬포트만을 지원한다. 만약 당신이 병렬포트가 hardware에서나 CMOS setup screen에서 configuration이 가능하다면 이 둘중 하나의 모드로 setup을 해야만 한다.

모든 cable이 견고하게 부착되었는지 다시한번 확인을 하라...

제 3 절 ZIP 드라이브를 위한 커널 설정

ZIP drive를 LINUX와 같이 사용하기 위해서는 당신은 SCSI system과 SCSI drive를 지원하도록 kernel configuration을 잡아야 하며, 당신이 사용하고 있는 host adapter를 지원하도록 해야 한다. 만약 당신이 kernel을 새로 만드는데 익숙하지 않다면 `/usr/src/linux`로 가서 거기에 있는 `README` file을 연구하여야 한다. 최신 버전의 `documentation` subdirectory에는 유용한 정보가 있다.

당신은 configuration 단계부터 kernel을 만드는 과정을 시작해야한다. 여기서 여러분은 당신이 필요한 특수한 kernel의 구성요소를 발견할 수 있다. `make config`는 고전적이면서, 순차적이고, 질답형식의 kernel configuration 방법이다.

최근의 커널들은 몇가지 새로운 변형을 제공한다 : `make menuconfig`는 menu-oriented interface를 가지며, `make xconfig`는 X 사용하에서 *tk* toolkit을 이용하는 version이다.

일단 여러분들이 당신의 kernel configuration 이 끝나고 나면 `make dep` 그리고 `make zlilo` (만일 *LILO*를 사용하지 않는다면 `make zImage` 를 사용하라)를 사용하여 새로운 kernel을 만들고 이것을 install 시킨다. 물론 shutdown & reboot하는 과정을 잊지 마라.

SCSI system의 일부 또는 전부를 modules로서 생성할 수 있다. 만일 여러분들이 이와 같이 했다면 여러분이 ZIP drive를 access하기 전에 `scsi.o`를 올리고 다음 `sd.o` 그리고 마지막으로 여러분의 host adapter를 올리는 것을 확인해야한다.

3.1 스카시용

만약 여러분들이 이미 system에 SCSI disk가 있다면, 그리고 ZIP drive가 같은 controller에 연결되어 있다면 더이상의 kernel configuration은 필요없다. 그렇지 않다면 여러분들은 필히 kernel을 재설정해서 compile해야만 한다.

만약 당신이 SCSI version의 ZIP drive를 support하도록 kernel을 만들었다면, SCSI와 SCSI disk를 지원하도록 만들어야만 한다. 그리고 여러분들은 여러분의 SCSI host adapter를 지원하는 driver 역시 꼭 선택해 주어야만한다. 만약 ZIP Zoom을 사용한다면 `aha152x` driver를 선택해야만 한다.

Linux source tree의 `drivers/scsi` subdirectory에 있는 `README` file과 SCSI HOWTO를 반드시 읽도록 하라. 당신의 adapter를 초기화하는 command parameter에 주의를 기울여야 한다.

예를 들면, 만약 당신이 ZIP Zoom card를 사용한다면, boot command에 여러분은 다음과 같이 parameter를 추가해 주어야 한다.(또는 `/etc/lilo.conf` file에 appen절로서 추가를 해 준다.)

```
aha152x=0x340,11,7,1
```

이것은 driver에게 ZIP Zoom card의 port address, IRQ를 말해준 것이다. - 당신의 카드에 jumper가 제대로 잡혀 있는지 확인하도록 하여라.

여러분들은 *LILO*나 *LOADLIN*에서 kernel configuration을 위하여 Paul Gortmaker's *BOOTPROMPT HOWTO* 역시 읽어야만 한다.

3.2 커널 1.2.13 의 PPA 드라이버

만약 여러분들이 안정 커널버젼 1.2.13에서 parallel port ZIP drive를 사용하길 원한다면 driver의 version 0.18을 가져와야만 한다. 이것은 `<ftp://gear.torque.net/pub/ppa.c>`에서 anonymous ftp를 통하여 가져 올수 있으며, 어떻게 loadable module로서 만드는지 설치 방법은 `ppa.c` source 내에 잘 설명되어 있다.

여러분들은 거의 항상 새로운 커널을 만들어야 한다는 것에 유념을 하도록 하십시오. 특히 *Slackware*의 pre-built kernel에서는 `ppa`와 같이 작동하지 않는다. 당신의 kernel이 SCSI support, SCSI disk support를 하도록 하고, `lp` printer driver는 support하도록 include 시켜서는 절대 안된다. 1.2.13에서는 이 두 driver가 같은 kernel 내에 같이 존재할 수가 없다.

여러분들은 `ppa` driver를 올릴때 (load 시에) port number와 약간의 timing parameter를 맞추어주어야 할 필요성이 있다. 이렇한 조정은 `ppa.c` file 내에 명시되어 있다. default로 driver는 ZIP drive가 parallel port 0x378에 물려 있다고 생각을 한다.

1.2.13 kernel을 위한 이 driver는 더 이상의 성능향상은 없을 것이다. 여러분들은 수개월후 2.0 kernel이 나올때 이것이 사라지길 기대해야 한다. (아마도 이문서의 작성시기가 옛날이군요.)

3.3 현재 배포되는 커널을 위한 PPA 드라이버

version 1.3.74 이후 ppa driver는 커널의 표준의 일부가 되었다. 1.3.78 근처로 해서 kernel의 다른 부분에 일부의 변화가 있었다. 그러나 1.3.85부터는 version 0.26 driver는 상당히 안정되어 졌다. LINUX 2.0 을 위한 `code-freeze`가 유용해 졌기 때문에 0.26이 다음 커널속에 들어갈것으로 기대한다. (이미 들어가 있지요.)

`ppa`를 지원하도록 kernel을 만들기 위해서는 low-level SCSI adapter의 list로 부터 SCSI support, SCSI disk support, IOMEGA ZIP/PPA-3 support를 가능하도록 include 시켜야만 한다. 여러분들은 driver를 loadable module로서 만들어야 한다.

여러분들은 driver를 조정하기 위하여 `/etc/lilo.conf`나 `insmod` 에서 command parameter를 사용하여야만 한다. 이것들은 모두 kernel source tree의 `drivers/scsi/README.ppa`에 기술되어 있다. 다음section에서 이에 관한 간략한 요약을 기술하도록 하겠다.

만약 여러분들이 `lp`와 `ppa` driver를 같은 parallel port에서 모두 사용하려 한다면 loadable module로서 커널 컴파일을 하여 필요시 각각 뛰워서 사용해야 한다. 동시에 둘을 모두 사용할 수는 없다.

loadable module은 더욱더 LINUX를 강력하게 만드는 사양이 되었다. 그리고 나날이 더욱 복잡해져가고 있다. 여러분들은 kernel source tree의 `docs/modules.txt`를 읽어야만 한다.

3.4 PPA 명령 라인에서의 변수

만약 여러분들이 당신의 kernel에서 `ppa`를 생성하였다면 여러분들은 parameter를 LILO 나 LOADLIN으로 부터 다음과 같은 문법으로 parameter를 조절해 주어야 한다.

```
ppa=base[,speed_high[,speed_low[,nybble]]]
```

여기서 `base`는 여러분들 parallel port의 i/o address를 말하며, `speed_high`는 driver에서 fast loop를 위한 timing constant이다. `speed_low`는 slower loop를 위한 timing constant이며, 그리고 `nybble`은 driver가 4-bit, 또는 nybble mode를 사용할수 있도록 해주는 flag이다.

예를 들면, defaults는 다음과 같이 기술될 수 있다 :

```
ppa=0x378,1,6,0
```

제 4 절 ZIP 드라이브 사용

만약 여러분들이 모든 필요한 부분을 설정했다면, kernel은 여러분들의 adapter와 drive를 booting 도중에 인식할 것이다. 만약 loadable module로서 여러분의 driver를 설정했다면 모듈을 올릴때 한번더 다음의 토의를 따르야 한다.

4.1 부팅시에 드라이버 찾기

여러분들의 system이 boot 될때 adapter, drive 그리고 drive내의 disk에 관한 몇가지 정보들을 보여준다. 만약 여러분들이 drive내에 disk가 없다면 이들 정보중 일부는 보이지 않을 수 있으며, disk를 넣을 때 또는 drive를 access할려고 touch할 경우에 보여지게 된다. 초보자들은 system boot시에 drive내에 disk를 넣고 booting 해 주길 충고한다. - 덜 햇깔리는 방법이다.

이렇한 message가 어디에 정확히 나타나나 하는 것은 여러분들이 system을 configuration하는 방법에 달려 있다. 그러나 이들은 `/var/adm/message`와 같은 log file로 저장되어 있다. 만일 어디에서도 이렇한 message를 발견할 수 없다면, 일반적으로 여러분들은 `dmesg` command로 이렇한 message를 볼수 있습니다.

만일 누군가에서 도움을 청할려면 이들 log information message가 아주 필요하다. bug report시에도 이 message를 반드시 카피해 주어야 한다.

여기에 kernel's log message의 예가 있다. 다음과 같이 보일 것이다 :

```
scsi0 : PPA driver version 0.26 using 8-bit mode on port 0x378.
scsi : 1 host.
  Vendor: IOMEGA    Model: ZIP 100           Rev: N*32
  Type:   Direct-Access                      ANSI SCSI revision: 02
Detected scsi disk sda at scsi0, channel 0, id 6, lun 0
scsi : detected 1 SCSI disk total.
SCSI device sda: hdwr sector= 512 bytes. Sectors= 196608 [96 MB] [0.1 GB]
  sda: Write Protect is off
Partition check:
  sda: sda1
```

이렇한 output은 parallel ZIP drive를 가진 1.3.87 system의 output이다. 정확한 output은 여러분들의 host adapter과 여러분들이 다른 SCSI device를 가지고 있는가에 따라 다양할 것이다.

첫째줄은 host adapter를 위한 low-level driver로 부터의 initialization message를 표시한 것이다. 이 경우에는 PPA이다. 다음줄은 drive의 식별정보가 SCSI bus에서 발견되는 각각의 device에의 의해서 보고되어 진다. Line 5는 drive에 인가된 device 이름이 보고된다. 이경우는 `/dev/sda1`이다. 이들보고는 만일 여러분들이 하나 이상의 SCSI adapter, 몇개의 drive를 가졌다면 하나이상의 서로 상이한 모듈들로부터 보고되어 지며 각각의 section은 더욱더 상세한 정보를 포함한다.

만약에 여러분들이 새로운 disk를 사용한다면 partition check에서 `/dev/sda4`라는 하나의 파티션을 보여 줄것이다. 만일 여러분들이 다른 SCSI device를 가지고 있다면 여러분들은 ZIP drive가 `/dev/sdb` 등과 같은 다른 device로소 나타난다는 것을 이해해야만 한다. - 그래서 message를 check해야 한다.

만약 이런 줄들이 보이지 않는다면, 무엇인가 당신의 hardware또는 kernel에서 잘못 설정된 것이다. mail을 보내기 전에 조심스럽게 모든 과정을 check 해보도록 하라.

만일 `scsi0:` line이 나타나지 않는 다면 그러면 당신은 host adapter와 그것의 driver를 적절하게 configuration을 해 주지 않은 것이다. 다른 driver는 당신에게 무엇이 틀렸는지 hint를 줄 것이다. 만약 당신의 driver가 detection되지 않는다면 cable이 잘못된 것이다. 만일 driver 이름이 할당되지 않는다면 당신은 아마도 SCSI disk support를 kernel에 include 시키는 것을 잊어 먹었을 것이다.

다른 debugging 정보를 위해서 kernel source tree의 `driver/scsi` 내의 `README` file을 참고하고 SCSI HOWTO를 참고하라.

4.2 Fdisk, mke2fs, mount, etc.

일단 여러분들이 ZIP drive의 drive 이름을 알고 이를 맞추어야 한다. 여러분들은 이 drive를 정상적인 LINUX disk management command를 가지고 다룰수 있다. `fdisk` (또는 `cfdisk`)는 disk에서의 partition disk를 다루기 위해 사용된다. `mke2fs`는 partition을 `ext2` file system으로 format하기 위해서 사용된다. - LINUX에서 가장 널리 사용되는 file system이다. `mount`는 formatted partition을 여러분의 계층적인 디렉토리 구조로 연결하기위해 사용된다.

여러분들은 이렇한 tools에 관한 man page를 필히 공부를 하여야하며 이들에 익숙해 져야 한다. 이제 몇가지의 아주 상이한 version의 `fdisk` program이 있으므로 주의해야만 한다.

나는 두가지의 흔한 시나리오에 대해 설명을 하겠다.

4.2.1 도스 포맷의 디스크

먼약 여러분들이 원래의 IOMEGA의 tools에 의해 생성된 DOS file structure를 가진 ZIP disk를 가지고 있다면 이들은 partition scan상에서 disk가 `/dev/sda4` 의 하나의 파티션만을 가진다고 보고를 할 것이다.

여러분들은 이들 disk를 `/zip`라는 directory에 mount를 할려면 이런식으로 해야만 한다.

```
mkdir /zip
mount -t msdos /dev/sda4 /zip
```

이제 여러분들의 disk는 `/zip`에 마운트되어 보일 것이다. disk가 mount 되어 있는 동안에 여러분들은 그것을 제거하려 해서는 안된다. 여러분들이 disk를 가지고 작업이 끝났을 경우 그것을 `umount` 시키고 release 시켜 계층적인 directory로 부터 분리시키면 된다.

```
umount /zip
```

일단 여러분들이 mount되는 directory `/zip`를 만들고 나면 여러분들은 다시 이것을 할 필요는 없으며 후에 다시 어디선가 mount를 시키면 된다.

4.2.2 리눅스로 다시 포맷하기

만약 여러분들이 ZIP disk를 지우고 그 위에 Linux native file system을 만들기를 원하면, 전 disk에 대하여 `fdisk`를 해 주어야 한다.

```
fdisk /dev/sda
```

그리고 모든 기존의 파티션들을 지워야 한다. (`d` 명령을 사용한다) 그리고 나서 `n` 명령 이용하여 새로운 partition을 만들어 주고 `w` 명령으로 변화사항을 기입을 하고 `q` 명령으로 마친다.

파티션을 format한다.

```
mke2fs /dev/sda1
```

(여기서 **1**이란 수는 여러분들이 `fdisk`에서 partition에 부여한 번호이다. 이제 여러분들은 disk를 mount하여 사용하면 된다.

```
mount -t ext2 /dev/sda1 /zip
```

(위에서 사용한 mount point를 다시 사용한다.)

4.2.3 ZIP Tools 디스크

여러분들은 ZIP drive와 같이 오는 disk를 이용하여 무언가 다른 작업을 할 수 있다. 예를 들면 software controlled write protection이 가능하다. 대부분의 사람들은 이것을 Linux에서 사용하기 위해여 DOS에서 unlock을 시킨다. LINUX는 locked disk를 access할 수 없다. 그리고 이것은 Iomega의 tools을 이용하여 unlock 시켜야만 한다.

다른 무엇보다도 write protection 기능이 Linux native program에서 곧 실현되어야 할 것이다.

4.3 디스크 꺼내기

여러분들은 disk가 mount되어 있지 않을 때 eject 시킬수 있다. 현재는 disk 를 eject 시킬수 있는 명령어가 없다. disk를 빼내기 위해서는 drive의 앞에 있는 button을 눌러 주어야만 한다.

Linux는 disk가 unlock 되기전에 쓰여진 모든 자료를 보증한다.

새로운 Linux를 위한 ZIP tools program은 SCSI and parallel version drive 둘다에서 IOMEGA의 특별한 기능을 지원할 것이다. (software-controlled ejection & write protection)

제 5 절 질문과 답

여기에 우리가 자주 접하는 의문들과 답을 적어 놓았다. 이들을 다시 묻기전에 답을 먼저 읽어 주기 바란다.

5.1 ZIP drive로 부터 시스템을 기동할수 (booting 할수) 있는가?

이것은 여러분들이 사용하는 host adapter에 달려 있다. 만약 여러분들이 5,6번 SCSI로 부터 booting을 할수 있는 ROM BIOS를 가진 host adapter를 가지고 있다면 SCSI ZIP drive로 부터 booting 할수도 있다. parallel port drive로 부터는 booting 할 수는 없다.

만약 여러분들이 main disk에 도스를 탑재하고 있고 ZIP disk로 부터 Linux를 부팅하고 싶다면 이것은 LOADLIN을 이용하여 가능하다. 여러분들의 kernel image는 반드시 DOS disk의 어딘가에 존재하고 있어야만 한다. 여러분들이 사용하는 kernel이 올바른 내장 driver를 가지고 있는지 확인하라.

내가 아는 한 parallel ZIP drive에 포함된 어떻한 일반적인 배포본에도 installation boot floppy는 없다. 그러나 이것은 Linux 2.0이 나오면 바뀔 것이다.

5.2 왜 IOMEGA는 파티션 번호를 4번을 사용하는가?

이것은 가장 흔하게 질문하는 것 중의 하나이다. 하지만 아무도 이에 대한 명확한 답을 들고 있지는 않다. 아마도 별다른 이유는 없을 것이다.

5.3 boot time시에 어떻게 disk를 mount할 수 있는가?

여러분들이 해야 할 것은 /etc/fstab file을 고쳐주는 것이다. 예를 들면 여러분들이 booting 시에 항상 DOS disk만을 사용한다고 가정을 하면 다음 과 같이 /etc/fstab 파일에 기입을 해 준다.

```
/dev/sda4   /zip  msdos  defaults  0 0
```

배포본에 따라서는 initialization scripts가 fstab 에 기술된 파티션을 대상으로 `fsck`를 실행하기도 한다. 이렇한 사실은 여러분들이 만일 boot시에 drive에 disk를 넣어 놓지 않든지 또는 다른 file system의 disk를 넣어 놓을 경우 문제를 야기할 수 있다는 점을 명심하기 바란다.

5.4 내가 만일 boot시 disk를 넣어 놓지 않을 경우 어떤일이 발생하는가?

kernel은 partition table을 읽을려고 시도를 할 것이다. 하지만 이렇한 작동은 시간만 낭비하게 될 것이다. 그러나 걱정할 필요는 없다. 일단 여러분들이 disk를 넣어주고 나면 그리고 나서 kernel에게 그기에 disk가 있으니 rescan을 하라고 알려 주기만 하면 된다.

HINT : 여러분들이 disk를 바꿀려고 할때 새로운 disk의 partition을 인식하도록 하기 위하여 fdisk 를 실행 시켜 주는 것이 아주 훌륭한 꽁수이다.

어떤 SCSI host adapter의 BIOS는 boot하는 동안에 파티션 table을 읽을려고 시도하는 경우도 있다. 만일 이렇한 checking 과정을 disable 시켜 놓지 않으면 여러분들은 boot 과정동안 항상 drive에 disk를 넣어 놓아야 할 것이다.

5.5 어떤 SCSI adapters가 ZIP drive와 호환이 되는 가...

Iomega는 화환이 되는 host adapter를 다음의 site에 기술해 놓았다.

`<http://www.iomega.com/techs/zip/what063.html>`

5.6 parallel drive를 real SCSI disk 처럼 사용할 수 있는가?

PPA-3 parallel-to-SCSI adapter는 single ASIC chip으로서 구현되어 있음 Iomega는 *VPI0*를 call한다. 이것은 ZIP drive의 controller card에 내장이 되어 있다. 실재로 SCSI bus가 존재하고 있다. 그러나 유용한 form은 아니다.

내가 두 cards 를 비교할려고 시도한 바에 의하면 SCSI bus를 외부로 확장할 때 요구되는 conventional electrical buffering circuits를 VIP0로 대체하는 것이 논리적으로 보인다.

5.7 parallel drive에 print를 끼워 사용할 수 있는가?

물론 할수 있다. 그러나 현재까지 여러분들은 `lp`와 `ppa` driver를 동시에 같은 parallel port에서 활성화 시켜서 사용할 수는 없다. resource sharing protocol이 현재 개발중에 있으며 초기 버젼이 나와있다. 그러나 Linux 2.0이 release 되기 전까는 불가능 할 것이다. (지금 버젼에서도 아마도 불가능하지요.)

결론적으로 말해서 printer와 ZIP drive를 연결할수는 있다. 그리고 이것을 여러분은 `lp`와 `ppa` driver를 loadable module로서 rebooting하지 않고 사용할 수 있다. 이렇게 하고 나서 여러분들이 만약 print를 하길 원한다면 `lp` module을 insmod로 load 해야만 한다. 그리고 만일 ZIP drive를 access할려면 insmod로 `ppa` driver를 load해야만 한다.

물론 만약 여러분들이 두개의 parallel port를 가지고 있고 하는 printer, 나머지 하나는 ZIP drive용으로 사용을 한다면 걱정할 필요가 없다. 단지 kernel을 `lp`와 `ppa` driver를 둘다 포함시켜서 컴파일해 주면 된다. 그리고 나서 적절한 configuration commands를 kernel command line에 포함시켜 주면 된다. 만일 여러분들이 LILO를 사용하고 있고, 그리고 printer를 0x378에 그리고 ZIP을 0x3bc에 연결해서 사용한다면 다음 line 을 `/etc/lilo.conf` 파일에 추가해 준다.

```
append = "lp=0x378 ppa=0x3bc"
```

5.8 PPA에서 EPP/ECP를 지원해줄 계획이 있는가?

`ppa`는 처음에 parallel bus에 대한 어떻한 command protocol과 technical specification도 가지지 않고서 개발되었다. 대신에 DOS emulator는 protocol의 reverse engineer로 정보를 모을수 있도록 되어 있다. emulator하에서 enhaced mode를 지원해주는 driver를 얻는 것은 불가능하다.

IOMEGA의 technical reference manual이 이제 이용이 가능하다. 그리고 enhanced parallel port chip에 관한 data sheet를 찾았다. 그래서 이제는 driver의 upgrade가 가능하다.

5.9 PPA가 IOMEGA의 parallel port tape drive와 같이 사용이 가능한가?

이들 drive는 floppy-tape drive이다. 그리고 SCSI가 관여되어 있지 않다.

몇몇 사람들은 이들 tape driver를 개발하는데 나의 방법을 적용 시키는 것이 홍미있다고 지적했었다. 첫번째 단계는 DOSemu 하에서 작동하는 dos driver를 얻는 것이다. 나는 누구에게서도 이렇한 방법이 성공했다는 얘기를 들어 보지 못했다.

5.10 PPA가 parallel SyQuest EZ135와 같이 작동을 하는가?

안된다. EZ135는 *ShuttlePort* parallel-to-IDE converter를 내장하고 있는 IDE drive 이다. 나는 작동을 하는 기본적인 driver를 가지고 있다. 그러나 언제 driver가 완성될 것인지에 대해서는 아직 예정이 없다.

나의 *web page* `<http://www.torque.net/ez135.html>` 에 관심을 가져 주기 바란다. 나는 결국에는 약간의 베타 테스터를 필요로하게 될것이다.

제 6 절 보다 최신의 정보를 얻을려면...

나는 나의 home page에 항상 최근의 update된 mini-HOWTO를 유지 할 것이다.

<http://www.torque.net/zip.html>.

보다 일반적인 ZIP FAQ web oage가 유지되고 있다. (좀더 Macintosh적인 견해에서)

<http://earth.cnct.com/home/steveg/zip.html>.

Iomega의 web page도 참고하기 바란다.

<http://www.iomega.com/>.

그리고 만일 parallel port programming에 대하여 좀더 일반적인 정보를 원한다면 다음을 방문해 보기 바란다.

<http://www.lvr.com/parport.htm>