Partition Rescue mini HOWTO

Jean-Daniel Dodin

july 27, 2000 - do005722_partition_03

역자 : 이호연 <mailto:guru2k@kldp.org>, 김정중 <mailto:justi@kldp.org>

헉 ! 내 디스크가 비어 있다 ! 내 리눅스가 없어졌다 ! 이와 같은 문제가 생겼다면, 이 문서를 읽어라...

목 차

제 1 절 안에 있는 것

이 미니 하우투는 "잃어버린 파티션 테이블" 문제에 관해서만 설명하며 다음의 경우에 사용할 수 있다 :

- "운영체제가 없다"는 메시지를 보이며, 컴퓨터에 접근할 수 없을 때,
- 당신이 새로운 시스템을 설치했고 리눅스로 더 이상 들어가지 못하며, 윈도우가 디스크의 모든 용량을 흡수 했을 때.

여기서는 옳은 방법을 알고 있으며 그렇게 한다면, 리눅스는 이러한 것들로부터 보통 안전하다는 것을 알게될 것이다. 윈도우도 운 좋게 가능하다.

우리는 문제를 쉽게 해결하기 위해서 먼저 문제 무엇을 해야할지와 복구 후 무엇을 해야 하는지 알게 될 것이다. 디스크가 지워지는 것을 막기 위해 할 일이 조금 있다. 보통 프로그램 혹은 사용자의 실수의 행동으로 자동적인 윈도우 혹은 리눅스 설치에 의해 된다 - 조심성을 제외하고는 이 것을 막는 것은 아무 것도 없다. 그러나 당신은 이미 조심스러울 것이다.

이 것은 MSDOS/Windows fdisk의 사용으로 할 수 있다. 당신이 할 수 있는 최대한 이 것을 피하라. 그러나 당신은 아마 할 수 없을 것이다.

나는 여러번 이 것을 했고, 나의 컴퓨터와 다른 사람 컴퓨터의 리눅스와 윈도우를 가끔 복귀를 시켰다. 행운을 빈다 !

제 2 절 지금 당장 할 것은 무엇인가 ?

이 문서를 호기심 혹은 정보를 찾기 위해서 읽으며 리눅스 시스템을 운영하고 있다면, 즉시 아래의 사항대로 하라 :

- 루트 터미널 혹은 xterm을 열어라,
- ”/sbin/fdisk -l”

컴퓨터의 모든 디스크에 있는 현재의 모든 파티션의 리스트에 만족할 것이다.

- 종이에 이 것을 쓰라 (혹은 이 것을 프린트 하기 위해 ”/sbin/fdisk -l | lpr” 을 하라) 그리고 미래에 쓰기 위해서 안전한 장소에 보관해둬라.

시스템 관리자가 아니라면, 그 문제에 관하여 걱정할 필요가 없고 이 문서를 그만 읽어도 된다.

제 3 절 Legal stuff

I am not responsible of any damage on any computer as a result of anyone reading this HOWTO. If you do any damage, _it is YOUR fault, NOT MINE !_ Be careful when partitioning disks, and don’t make any mistakes, because it can be fatal ! Backup all your important data and check that everything you do is correct ! What is described here worked on my computer, but it may or may not work on your computer. Although it should work for everyone, I can’t guarantee anything. This is the last warning you get : _BACKUP IMPORTANT DATA !_ Or, to put it short: Use at your own risk !

제 4 절 지금 당장 알아야 할 것은 무엇인가 ?

당신의 하드 디스크의 어떠한 문제가 생기면, 당신은 쓰기 모드로 사용하는 것을 중단 해야만 한다. 적어도 일어난 일을 이해하기 위해서는 시간이 필요하다.

어느날 아침, 일어났을때, 당신의 컴퓨터가 ”적재할 수 없습니다, 시스템이 설치되지 않았습니다” 라는 메시지를 보이더라도, 당신은 모든 프로그램을 다시 설치하면 안된다.

당신이 윈도우를 설치했으면, 당신의 데이터를 복구 할수 있다고 약속하지는 못하겠지만 당신의 리눅스 프로그램의 전부를 복구할 수 있을 것 같다. 이 것은 디스크 구조의 너무 낮은 곳에 위치해 있지 않기 때문이다. 몇몇 윈도우 바이러스는 디스크 실린더를 먼저 지우기 때문이다. 그렇지만 나는 이러한 바이러스 실험을 하지 않았고 확실하다고는 말하지 않겠다. 어쨌든 복구해보려고 노력해보라.

당신은 또한 내가 주는 모든 정보를 알아야만 한다. 나도 다른 사람도 아니지만 당신은 이 정보를 쓸 수 있는 데이터에 관한 어떠한 문제에 책임이 따른다. 세계에는 매우 많은 다른 종류의 시스템이 있다. 나는 오직 당신에에 행운을 있기를 바라며, 데이터를 복구시키면 행복해질 것이다.

제 5 절 디스크

하드 디스크는 0에서부터 최대 값까지의 섹터 넘버로 만들어진다.

dmesg는 알려준다, 예를 들면 :

hdb: ST34321A, 4103MB w/128kB Cache, CHS=523/255/63

CHS는 Cylinders, Heads, Sectors를 말한다.

523*255*63=512 바이트의 8401995 섹터, 이와 같이 4103Mbytes이다. 이 것은 오직 논리적인 기록이다. 디스크 커버에 쓰여 있는 것은 필요하지 않다 (토탈 사이즈 제외).

섹터의 실제 크기에 대해서 우리는 흥미가 없다. 우리는 아무 것도 수정하지 않기를 원하며 이 전의 상태로 되돌리기를 원한다. 우리는 이미 fdisk에 의해 크기를 알 것이다.

시스템에 의한 이 크기는 BIOS(Basic Input/Output System - the PC's Rom)의 작업에 직접적으로 의존 한다. BIOS에서 가리키는 하드 디스크의 방법은 매우 중요하다. 새로운 디스크 에서는, BIOS의 하드 디스크 자동 인식을 사용하기 위해 "yes"를 선택하는 것이 좋다. 어쨌든, 이 레벨의 어떠한 수정은 아마 디스크의 모든 데이터를 파괴할 것이다. 그래서 매우 중요한 이유 없이 이 작업을 하지 말기를 바란다.

이 것은 아마도 당신의 디스크 사용량일 것이다, 그래서 걱정하지 마라.

제 6 절 파티션

지금 디스크는 매우 커서, 13 Gb가 보기 드문것이 아니다. 그래서 자료들 을 한곳에만 보관하여 관리 하는 것은 쉽지 않다. 윈도우즈만 사용하거나 리눅스를 사용한다면 아마도 이것이 얼마나 비 능률적인 것인가를 깨달을 것이다.

그래서 하드 디스크는 때때로 "파티션(partitions)"이라는 여러 조각들로 나뉜다(자세한 사항을 알고 싶으면 "파티션 미니 하우투("partition mini HOWTO"를 참조하고 또한 /usr/doc/package/util/README.fdisk를 읽어 봐라)

그럼 나의 fdisk -l의 결과중 일부분을 보자 :

Disk /dev/hdb: 255 heads, 63 sectors, 523 cylinders Units = cylinders of 16065 * 512 bytes

Device Boot Start End Blocks Id System

/dev/hdb1 1 153 1228941 83 Linux

/dev/hdb2 154 166 104422+ 82 Linux swap

/dev/hdb3 * 167 291 1004062+ 83 Linux

/dev/hdb4 295 523 1839442+ 5 Extended

/dev/hdb5 295 422 1028128+ 83 Linux

/dev/hdb6 423 523 811251 6 FAT16

이건 나의 두번째 하드 디스크이다. 이것을 이해하려고 노력해 보아라. (첫번째는 너무 간단해서 흥미로울 정도이다).

/dev/hdb는 나의 두번째 ide 디스크이다(프라이머리의 슬레이브)

/dev/hdb1은 첫번째 파티션이다. 1블록 부터 153블록 까지 사용되고 있다.

첫번째 파티션과 같은 것들이 4개 존재 할수 있다. 만약 4개 이상을 원한 다면 그것은 "extended"이야만 한다(반드시 4번째일 필요는 없다). 그리고 다른 파티션들은 확장된 것(extended one) 속에 "logical"라는 이름으로 있어야 한다. 파티션 번호 5와 파티션 번호 4는 같은 시작이 라는 것을 주 목 해야 한다. 5번은 논리적인 것이고, 4번은 확장된것이다. 만약 4개의 프라이머리 파티션이 없더라도논리적인 것은 항상 5번에서 시작 한다.

제 7 절 문제가 있는 이유는 ?

문제는 설치된 운영체제가 디스크를 공유하고 시작할때 바이오스(bios)는 오직 파티션 테이블(partition table)이라고 불리는 디스크의 매우 앞부분 에 있는 것만 검색 한다. 이 파티션 테이블은 Master Boot Record (MBR) 속에 부트 로더와 함께 위치해 있다.

어떠한 운영체제에 의한 MBR의 잘못된 사용은 문제를 야기 시킨다. 어떤 시스템을 인스톨 시키려 할때 "자동으로 파티션을 조정하시겠습니까?"와 같은 질문에 대해 '예'라고 답한다면 문제가 생길 수 있다. 이것은 특별히 윈도우즈의 관습 적인 설치(custom windows installation)에 의해 생긴다 (만약 그렇지 않는다면 윈도우즈 cd가 포함 되었다). 그러나 그러한 문제 는 대부분의 최신 리눅스 배포판에 있는 "영리한"(전혀 영리하지 않다!) 설치 프로그램에 의해서도 생길수 있다.

제 8 절 문제 해결하기

조심해라 ! 당신이 이전의 시스템으로 돌려주는 것을 이끌 다음의 설명을 따라 해라 ! 당신은 결정을 해야 한다...

8.1 간단한 경우

이제 할 작업은 모두 간단하다 :

- fdisk를 쓸 수 있는 디스크(플로피 혹은 cd)로 리눅스를 시작하라 - 대부분의 배포판의 복구 디스크로 할 수 있다.
- fdisk -l 의 내용을 적은 종이에 적는다.

이 정도면 충분하다

1. 리눅스를 시작하라.
2. fdisk /dev/hda (복구할 디스크)를 실행하라.
3. 피해를 입은 디스크에 존재하는 모든 파티션들을 지우기 위해 (d 명령) fdisk를 사용하라.
4. 종이에 업급된 모든 프라이머리 (1-4) 파티션을 만들기 위해 fdisk를 사용하라.
5. 적합한 태크 (t 명령)을 주라 : 82 는 리눅스 스왑, 83은 리눅스 메인 (L은 리스트를 보여준다), 5는 확장 그리고 논리 파티션을 만들기전에 해야만 한다.
6. 논리 파티션을 만든다.

fdisk는 작지만 훌륭한 프로그램이다. fdisk로 만든 다른 프로그램이 많이 있다. 그러나 드러난 것을 좋아한다 (나는 리눅스의 것을 말하며, 물론, 다른 운영체제는 아니다...).

fdisk는 w를 치기 전에는 디스크에 어떠한 것도 기록하지 않는 다는 것을 알아둬라. 당신이 실수를 두려워하는 경우라면, q (quit) 혹은 Ctrl C (\^{ }C) 를 쳐서 안전하게 종료하라.

당신의 새 파티션 테이블이 쓰여졌을때, 리눅스를 다시 시작하라. 평상시 대로 하면 안된다 : lilo는 피해를 입었을 수 있고 당신은 부트 플로피 혹은 부팅 시디가 필요할 것이다 ("설피된 파티션으로 부팅"을 골라라).

당신이 lilo를 사용하여 부팅을 한다면, 루트로 로그인 하자마자, 당신이 제일 좋아하는 부트 로더를 재설치하기 위해 "lilo"를 쳐라.

당신의 리눅스는 모두 있을 것이다, 테스트 해보라. 윈도우가 있다면 윈도우로 시작해보라. 만약 할 수 없다면, 당신의 데이터를 리눅스로부터 읽을 수 있다면, (매우 조그만) 문제가 있다. 아마도 섹터 리드에 의한 raw 섹터일 것이다. 만약 당신이 당신의 데이터가 디스크 섹터에 있다는 것을 확인할 수 있다면, 그 것을 파일로 복사하기 위해 dd를 사용하라. 이 것은 형명한 것이다. 기 것은 이 복구 mini-HOWTO의 범위에 벗어난다.

8.2 간단하지 않은 경우

8.2.1 By hand

이 것은 이 전의 방법으로 못 할때, fdisk 문서의 부족 혹은 데이터 밖의 사용을 사용하지 못 할때이다.

먼저, (fdisk를 제외하고) 디스크를 쓰는 작업을 중단해야 하는 것을 알아둬라. 당신은 데이터를 지울 수 없고, 그래서 블럭 바이 블럭을 시도해 볼 수 있다. 이 것으로 파티션이 시작한다는 것을 알아야할 필요가 있다. 만약, 153이 맞지 않다고 말하면, 154를 시도하고, 등등.

이 것은 지겨운 작업이지만 리눅스 파티션의 사이즈는 대체로 기억하고 있다면, 할 수 있는 기회가 있다.

8.2.2 gpart

그러나 넷에 접속할 수 있거나 "gpart"를 가지고 있다면 더 좋은 방법이 있다.

"gpart - guess PC-type hard disk partitions" 은 gpart (man gpart)의 맨 페이지의 첫째줄에 있다.

"gpart 하드 디스크에 있는 파티션을 추측한다. 만약 프라이머리 파티션 테이블이 없어졌고, 덮어썼거나 파괴된 디스크가 존재하지만 운영체제는 그들에 접근을 못 한다.". 이 것은 정확히 우리가 필요하다.

gpart는 아직 베타임에도 불구하고 아주 좋은 툴이다.

문제는 아래와 같다 : 어떤 파니션의 첫번째 블록이 마크되었다. 그러나 덮어쓰여지지 않았다면 절대 "unmarked"가 아니다. 그래서 많은 "첫번째 파티션 블록"은 오래된 디스크에 존재하고 있으며 gpart는 작업을 하려고 시도 하는 프로그램 중 좋은 프로그램이다. 사실은 시도 하는 것이 아주 어렵지 않다. gpart에 의해서 쓰는 것은 아무 것도 없다.

hdb의 이전의 gpart의 결과이다 :

root@charles:/home/jdd > gpart /dev/hdb

Begin scan...

Possible partition(Linux ext2), size(1200Mb), offset(0Mb)

Possible partition(Windows NTFS), size(1200Mb), offset(1200Mb)

Possible partition(Linux ext2), size(1004Mb), offset(2402Mb)

Possible partition(Windows NTFS), size(1600Mb), offset(4102Mb)

End scan.

Checking partitions...

* Warning: partition(OS/2 HPFS, NTFS, QNX or Advanced UNIX) ends beyond disk end .

Partition(Linux ext2 filesystem): primary

Partition(OS/2 HPFS, NTFS, QNX or Advanced UNIX): primary

Partition(Linux ext2 filesystem): primary

Partition(OS/2 HPFS, NTFS, QNX or Advanced UNIX): invalid primary

Ok.

Guessed primary partition table:

Primary partition(1)

type: 131(0x83)(Linux ext2 filesystem)

size: 1200mb #s(2457880) s(63-2457942)

chs: (0/1/1)-(152/254/61)d (0/1/1)-(152/254/61)r

Primary partition(2)

type: 007(0x07)(OS/2 HPFS, NTFS, QNX or Advanced UNIX)

size: 1200mb #s(2457880) s(2457944-4915823)

chs: (152/254/63)-(305/253/60)d (152/254/63)-(305/253/60)r

Primary partition(3)

type: 131(0x83)(Linux ext2 filesystem)

size: 1004mb #s(2056256) s(4919781-6976036)

chs: (306/61/49)-(434/60/47)d (306/61/49)-(434/60/47)r

Primary partition(4)

type: 000(0x00)(unused) size: 0mb #s(0) s(0-0) chs: (0/0/0)-(0/0/0)d (0/0/0)-(0/0/0)r

당신이 보는대로, 프라이머리 파티션은 덮어 쓰여졌을 수 있다. 그러나 확장은 그대로다.

도스 파티션은 "windows NTFS"하고 라벨이 붙어있는데 Windows 2000 (매우 무서운 경험 !)를 설치하려고 시도하는 동안 만들어졌다. "invalid"는, 사실은 확장 파티션.

이 것과 함께, fdisk를 사용하여 파티션 테이블을 다시 만들 수 있다.

8.3 부자의 경우

Partition Magic는 상업 제품이지만, 싸지는 않지만 매우 좋은 평가를 받고 있는 프로그램이다 (프랑스돈으로 백 달러에 거의 가깝다). 그렇지만 나는 절대 이 것을 사용하지 않을 것이며 평가도 하지 않을 것이다. 이 것은 파티션과 어떠한 것도 할 수 있다고 하고 복구시킬 수도 있다.

최초의 Ralf partition-rescue mini HOWTO는 본래 Partiton Magic의 사용에 기반을 두었다. 그래서 나는 당신의 리눅스 파티션에 가치가 있는 데이터가 있고 능력이 있다면 이 것이 가장 좋은 해결책이라고 생각한다. 그렇지만 매우 최근의 Partition Magic보다 더 좋은 프로그램이 있다고 생각하고 더 좋은 메뉴얼이 있다고 생각한다.

제 9 절 저자

이 하우투의 저자는 Jean-Daniel Dodin이다. 나는 jdanield@doodin.net 혹은 jdanield@linux-france.org에서 참여할 수 있다.

나의 웹 사이트는 http://www.dodin.net이다. 나의 최대 관심사인 리눅스 페이지는 www.linux-france.fr/article/jdanield 에 있다.

나는 이전의 partition rescue mini HOWTO를 쓴 Rolf Klausen (E-mail: rolfk@romsdal.vgs.no http://www2.romsdal.vgs.no/ rolfk/) 에게 감사한다. 심지어 나는 거의 다시 썼다. 그는 좋은 아이디어를 처음으로 가지고 있었다.

리눅스 커뮤니티의 다른 멤버 들과 리눅스를 지원하고 리눅스 문서와 리눅스 프로그램을 쓰는 사람과 모든 LDP의 저자들과 리눅스와 실질적으로 어떠한 혼란이 있는 사람들. 특히 Linus B. Torvalds - 그는 _왕_이다 !!!

나는 또한 gpart를 쓴 Michail Brzitwa <mb@ichabod.han.de> 에게 감사한다. I want also to thank Michail Brzitwa <mb@ichabod.han.de> for writing gpart !

제 10 절 가장 최근의 버전

이 미니 하우투의 가장 최근의 버전은 나의 리눅스 웹 사이트인 www.linux-france/article/jdanield/ 에서 찾을 수 있다.