Linux Complete Backup and Recovery HOWTO

CurleyCharles

ccurley at trib dot com

도운이

송호철

fanatics (at) dreamwiz.com

당신의 하드 디스크 드라이브가 막 아주 비싼 하키 퍽이 되버렸다고 상상해보라. 화재가 나서, 당신의 컴퓨터 케이스는 이제 살바도르 달리가 즐겨 그렸던 것처럼 보인다고 상상해보라. 이제 무엇을 해야 할까?

완전 복구, 혹은 Bare Metal Recovery(BMR)라고 불리는 과정은 재난에 준하는 오류 상황에서 컴퓨터를 다시 재설치해 가는 과정을 말한다. 완전 복구를 위해서는, 당신은 파일 시스템 뿐만 아니라, 분할 정보와 기타 데이터의 완벽한 백업을 가지고 있어야 한다. 이 HOWTO는 Bare Metal Recovery를 하기 위한 리눅스 컴퓨터 백업을 만드는 방법과 그 백업을 이용해서 Bare Metal Recovery를 하는 방법을 단계별로 설명하고 있는 문서이며 관련된 스크립트를 포함하고 있다.

고친 과정

고침 1.1 2002-09-10고친이 c^2

`make.fdisk`에 ext3 분할영역을 다루는 새로운 코드와 `initrd`에 대한 주의 사항 추가

고침 1.0 2002-07-24고친이 c^2

1단계에서 bz2 압축이 사용 가능하고, 손상된 블록에 대한 점검을 실행할 수 있으며, 1단계 전체를 실행할 수 있는

고침 0.052002-07-04고친이 c^2

조그마한 수정을 가함.

고침 0.042002-05-09고친이 c^2

테마 추가, 절(section)에 소폭 변화, 구성에서 몇가지 소폭 변화.

고침 0.032002-04-09고친이 c^2

자료 출처에 Beeblebrox Project추가.

고침 0.022002-01-27고친이 c^2

자료출처에 Mondo 추가

고침 0.012001-10-25고친이 c^2
LDP 배포를 위한 초기판

1. 서문(Introduction)

보통의 Bare Metal Restoration의 순서는 다음과 같다. 1) 제품디스크로부터 운영체제를 설치한다. 2) 데이터를 복구할 수 있는 백업 소프트웨어를 설치한다. 3) 데이터를 복구한다. 4) 설정과 허가권 등을 점검해서 컴퓨터의 기능성을 복구한다.

이 HOWTO에 설명되어 있는 순서와 스크립트는 여러분이 운영체제를 재설치하는 시간을 줄여 줄 것이다. 여기서 설명되어 있는 내용에서는 컴퓨터로부터 백업된 파일만을 복구할 것이다. 여러분이 복구할 때 설정파일은 그대로 있을 것이며, 이것이 설정파일과 데이터를 점검하는 것에 소요되는 시간을 줄여줄 것이다.

1.1. 저작권 정보(Copyright Information)

저작권은 Charles Curley에 있고 아래에 서술한 GNU Free Document License(GFDL) 하에 배포된다. GNU Free Document License 1.1 버전 혹은 Free Software Foundation에서 발간한 최신의 버전의 조항에 의거하여 이 문서를 복사, 배포 수정할 수 있는 행위가 보장되어 있다. 이 라이센스의 복사본은 “GNU Free Documentation License”.라고 이름 붙은 절(節)에 포함되어 있다.(번역자 주: GNU Free Documentation License는 번역하지 않았습니다. 따로이 번역된 문서를 찾아 보시기 바랍니다.)

만약 질문이 있다면 `<linux-howto at metalab.unc.edu>`로 연락해 보길 바란다.

1.2. 책임의 한계(Disclaimer)

저자 혹은 Linux Documentation Project (http://www.tldp.org/)나 그 외의 누구도 이 문서의 내용에 책임을 질 의무가 없다. 개념, 예제와 그 외의 내용을 당신 자신이 책임지고 사용하라. 당신의 시스템을 망가뜨릴 수 있는 오류나 부정확함이 있을 수 있다. 오류가 발생할 것 같지는 않더라도 주의 깊게 진행하라. 저자는 그것에 어떤 책임도지지 않는다.

특별히 다르게 명시되어 있지 않다면, 모든 저작권은 각각의 소유자에게 있다. 이 문서에 있는 용어의 사용함으로써 어떤 상표나 서비스 마크의 유효성에 영향을 미치는 것으로 생각되어서는 안 된다.

특정 제품이나 상표의 이름을 거론하는 것이 추천으로 오해되지 않았으면 한다.

주기적인 간격으로 백업을 할 것이며 주요한 설치를 하기 전에 꼭 시스템을 백업하는 것을 강력히 추천하는 바이다.

덧붙여 이 HOWTO에 있는 어떤 내용을(특히 스트립트) 막 시험해 볼 수 있는 희생용(!) 컴퓨터를 사용할 것을 강력히 추천한다.

1.3. 새 버전(New Versions)

여러분은 이 문서를 홈페이지 (http://w3.trib.com/~ccurley/Linux-Complete-Backup-and-Recovery-HOWTO.html))나 Linux Documentation Project (http://www.tldp.org/) 홈페이지에서 여러 가지 형태로 발견할 수 있을 것이다. 이 문서에 대한 조언은 `<ccurley at trib dot com>`로 보내 주기 바란다.

- bzip2 compressed chunky (많은 작은 페이지로 되어 있고, 빨리 읽을 수 있음.) HTML (http://w3.trib.com/~ccurley/Linux-Complete-Backup-and-Recovery-HOWTO/Linux-Complete-Backup-and-Recovery-HOWTO.chunky.html.tar.bz2).
- bzip2 compressed smooth (하나의 거대한 페이지 --검색하기 용이함.) HTML (http://w3.trib.com/~ccurley/Linux-Complete-Backup-and-Recovery-HOWTO/Linux-Complete-Backup-and-Recovery-HOWTO.smooth.html.tar.bz2).
- bzip2 compressed postscript (US letter format) (http://w3.trib.com/~ccurley/Linux-Complete-Backup-and-Recovery-HOWTO/Linux-Complete-Backup-and-Recovery-HOWTO.ps.bz2).
- bzip2 compressed PDF (US letter format) (http://w3.trib.com/~ccurley/Linux-Complete-Backup-and-Recovery-HOWTO/Linux-Complete-Backup-and-Recovery-HOWTO.pdf.bz2).
- Source (http://w3.trib.com/~ccurley/Linux-Complete-Backup-and-Recovery-HOWTO/Linux-Complete-Backup-and-Recovery-HOWTO.tar.bz2)를 사용해,
  Luke.

1.4. 감사의 글(Credits)

이 문서는 *Linux Journal* (http://www.linuxjournal.com/)에서 발간된 2개의 기사로부터 유래하여 만들어 졌다. 이 HOWTO를 만드는데 도움을 준 *Linux Journal*에 감사를 표한다.

훌륭하게 HOWTO 편집을 해준 Joy Y Goodreau에게 감사를 표한다.

1.5. 피드백(Feedback)

이 문서에 대한 feedback을 환영한다. 당신의 교정, 제안 등이 없이 이 문서는 존재하지 못할 것이다. 당신의 내용추가, 논평과 비판 등을 나(`<ccurley at trib.com>`)에게 보내주십시오.

1.6. 번역(Translations)

모두가 영어를 사용하지는 않는다. 자원자를 환영한다.

2. 개요(Overview)

아래에 설명된 과정은 쉽지 않으며 당신의 데이터에 해를 입힐 수 있다. 필요가 있어 사용 하기 전에 많은 연습을 해야 한다. 내가 한대로 할 것이며, 연습용 컴퓨터에서 꼭 연습하길 바란다!

이 HOWTO의 대상이 된 컴퓨터는 한개의 IDE 하드 드라이브에 Red Hat (http://www.redhat.com) 7.1 리눅스 서버 혹은 워크스테이션이 설치된 펜티엄 컴퓨터이다. 이 컴퓨터는 "희생용"으로 시험될 목적으로 사용될 것이므로, 많은 양의 데이터를 가지고 있지는 않다. 즉, 나는 이 내용을 주 컴퓨터와 주 데이터에 시험해 보기를 원하지 않는다는 말이다. 또한, 시험을 시작하기 전에 막 설치를 새로이 하였고 이는 곧 내가 원래의 설정으로 되돌될 필요가 있다면 항상 다시 설치를 할 수 있다는 말이 된다.(시험용 컴퓨터를 꼭 사용해야 한다는 얘기다.)

> **주의:** 예제 명령어는 대부분의 경우에 내가 대상 시스템을 복구하기 위해서 타이핑해야 만 했던 것을 그대로 보여줄 것이다. 여러분은 비슷하지만 다른 매개 변수를 가진 명령어를 사용해야 할지도 모른다. 여러분 자신의 설정을 반복했는지, 시험용 컴퓨터의 설정을 복제했는지 확인하는 것은 여러분에게 달려있다.

기본 절차는 내가 *Linux Journal* (http://www2.linuxjournal.com/lj-issues/issue78/3839.html)에서 호의적으로 서평을 쓴, W. Curtis Preston의 *Unix Backup & Recovery* (http://www.oreilly.com/catalog/unixbr/), (O'Reilly & Associates, 1999)에 설명되어 있다. 그러나, 그 책은 세부적이고 당면한 문제에 대해서는 내용이 조금 적다. 예를 들어, 정확히 어떤 파일을 백업해야 하는가? 보존해야할 필요가 있는 메타데이터는 무엇이며 어떻게 보존해야 하는가?

이 HOWTO에 설명된 순서대로 진행하기 전에 전형적인 백업 도구인 Amanda, BRU™, tar, Arkeia®나 cpio를 이용해서 여러분의 시스템을 백업할 필요가 있다. 그러면, 이제 질문은 뜨겁게 익혀진 하드웨어로부터 시작해서 데이터를 복구할 소프트웨어를 실행시킬 수 있는 지점까지 어떻게 도달할 것인가이다.

Red Hat Package Manger(RPM)을 기반한 리눅스 배포판의 사용자는 또한 일상 백업의 한 부분으로서 RPM 메타데이터 또한 저장해 두어야 한다.

```
bash# rpm -Va > /etc/rpmVa.txt
```

위의 것과 비슷한 명령어를 여러분의 백업 스크립트에 넣음으로써 bare metal restoration 후에 비교할 만한 기준을 얻을 수 있을 것이다.

이 지점까지 가기 위해서, 여러분은 다음과 같은 것을 가지고 있을 필요가 있다.

- 필요하다면 부품을 대체한 후, 여러분의 하드웨어를 다시 시동시킨다. 시간, 날짜, 하드 드라이브 매개 변수는 BIOS에 정확히 설정되어 있어야 한다. 이 시점에서 또 다른 하드 드라이브를 사용할 것을 미리 대비할 필요는 없다.
- 병렬 포트를 쓰는 Iomega® (http://www.iomega.com/) ZIP® drive (http://www.iomega.com/zip/products/par100_250.html)나 그와 동등한 것. 적어도 30 MB의 여유 공간이 필요할 것이다.

- 백업 매체
- 복구 소프트웨어를 실행시킬 수 있는 최소한의 리눅스 시스템

그 지점까지 가기 위해서, 여러분은 적어도 두 단계 혹은 세단계의 백업을 필요로 한다. 정확히 무엇을 백업하며 어떤 단계에서 백업할 것인지는 여러분의 복구 절차에 따라 결정된다. 예를 들어, 여러분의 tape server를 복구할 때, 복구 중에 네트워킹은 필요가 없을 것이다. 그래서 정규 백업에만 네트워크를 백업하라.

여러분은 여러 단계로 복구를 할 것이다. 1단계에서 여러분은 분할, 파일 시스템, 등을 만들고 ZIP disk로부터 최소 파일시스템을 복구할 것이다. 1단계의 목표는 부팅 가능 후에 네트워크 연결, 테이프 드라이브, 복구 소프트웨어 혹은 2단계를 위해서 필요한 것들을 실행시킬 수 도록 하는 것이다.

2단계는, 만약 필요하다면, 백업소프트웨어와 관련 데이터베이스의 복구로 이루어져 있다. 예를 들어 여러분이 Arkeia를 사용하고 백업 서버로서 Bare metal recovery ZIP 디스크를 사용한다고 가정 해보자.Arkeia는 서버의 하드 드라이브에 거대한 데이터베이스를 가지고 있을 것이다. 만약 원한다면 여러분은 테이프 등에서 데이터베이스를 복구할 수 있을 것이다. 그러는 대신에, 전체 Arkeia 디렉토리(/usr/knox에 위치한)를 tar 나 gzip으로 압축하고 nfs나 ssh를 통해서 그것을 다른 컴퓨터에 저장하는 것은 왜 안 될까? 우리가 아래에 정의하겠지만, 1단계는 X를 포함하고 있지 않고 있다. 그래서, 만약 여러분이 백업 프로그램 뿐만 아니라 X 또한 백업하기를 바란다면 몇 가지 실험을 해봐야 할 것이다. 몇몇 복구 프로그램은 X를 필요로 한다.

물론, 여러분이 다른 백업 프로그램을 사용하고 있다면, 여러분은 거기에 맞춰서 몇 가지 다른 작업을 해야 할 것이다. 여러분은 그 프로그램을 실행시키기 위한 디렉토리와 파일들을 찾아야만 할 것이다. 만약 여러분이 tar, gzip, cpio, mt 혹은 dd를 백업과 복구 도구로서 사용한다면, 그 프로그램들은 아래에 설명한 1단계의 한 부분으로서, 우리의 ZIP 디스크에 저장된 후 역시 ZIP 디스크로부터 복구될 것이다.

마지막 단계는 테이프나 다른 매체로부터 완전한 복구이다. 마지막 단계를 완료한 후에 여러분은 완전히 복구 되고 잘 작동하는 시스템으로 부팅 될 수 있어야만 한다.

2.1. 한계(Limitations)

이 HOWTO는 최소한의 백업을 만드는 것만을 다룬다. 그래서, 새 하드웨어("Bare Metal")에 그 백업을 복구한 후, 완전히 복구하기 위해서는 정규 백업을 이용해야 한다. 이 HOWTO에는 정규 백업은 다루지 않는다.

매우 범위가 좁고 간결한 것임에도, 이 HOWTO가 모든 것을 설명하지는 못한다. 여러분은 여전히 좀더 연구, 스크립트 편집과 시험 등을 해야 할 것이다.

여기에 나온 스크립트는 하드 드라이브에서 있던 대로 분할 영역(partition) 데이터를 정확히 복구할 것이다. 만약 여러분이 동일한 컴퓨터에 복구를 한다든지, 적어도 동일한 하드 디스크에 복구를 한다든지 할 경우에 이 스크립트는 매우 훌륭하다. 하지만 그렇게 자주 발생하는 경우는 아니다. 당분간은 사용할 수 있는 2개의 방법이 있다.(이 HOWTO의 나머지를 읽은 후에 좀더 이해하기 쉬울 것이다.)

- 분할영역표(Partition table) 입력 파일을 편집하라. 나는 몇 분이면 가능하다. 여러분은 이것을 새 파티션을 추가하거나 기존의 파티션을 지우는데 사용할 수 있다.(파티션 테이블 입력 파일을 이용하는 스크립트도 수정하는 것이 좋다.)
- 새로운 파티션 테이블을 만들고 거기서 시작하라 이것이 왜 `restore.metadata`가 하드 드라이브 재구성 스크립트를 호출하지 않는가에 대한 이유이다.(다른 이유는 내가 여러분의 하드 드라이브가 어떤 것인지 모르기 때문이다.) 재작성 스크립트에서 **fdisk**를 호출하는 부분을 삭제하는 것을 잊지 말라.

여기에 나온 스크립트는 단지 ext2fs, FAT12와 FAT16만을 다룬다. 다른 열정적인 자원자가 이 스크립트에서 가능한 코드를 제공할 때까지, 여러분은 우리가 다루지 않는 파일 시스템을 백업하고 복구하는 것에는 다른 도구를 이용해야 할 것이다. Partition Image (http://www.partimage.org/)는 쓸만한 대안이 될 것 같아 보인다.

3. 준비(Preparation)

> 경고!: 규칙적인 계획에 따란 통상적인 백업을 실시하시오. 여러 분이 그렇게 하지 않는 다면, 이 HOWTO는 아무 쓸모가 없다.

응급 디스크를 만들기 바란다. 나는 tomsrtbt (http://www.toms.net/rb)를 사용한다. 문서화가 잘 되어 있고 하나의 플로피 디스켓에 많은 쓸 만한 도구를 담고 있다. 활성화 된 목록이 있으며, 내가 가졌던 몇몇 의문점 들이 빠르고 정확하게 답변되어 있다.

다음으로 앞으로 필요하게 될 운영 체제 백업을 어떻게 하는 지 상세하게 이해하길 바란다. 그러면, 여러 분은 통상 백업으로 복구할 수 있게 될 것이다. 나는 Preston의 조언을 따랐고 Iomega 병렬 포트 ZIP 드라이브를 사용했다. 그 드라이브에는 90MB의 사용 공간이 있다. 나는 내 데스크 탑을 백업하는 데 약 85MB를 필요로 했다. 그래서, 100MB ZIP 드라이브를 사용하는 것이 여러 분을 위해서 좋을 것이다.

3.1. ZIP 드라이브 설치(Installing the ZIP Drive)

ZIP 드라이브의 설치는 the Linux Documentation Project (http://www.tldp.org/)와 홈페이지, http://www.njtcom.com/dansie/zip-drive.html에서 이용 가능한 ZIP Drive HOWTO (http://www.tldp.org/HOWTO/mini/ZIP-Drive.html)에 설명되어 있다.

4. 1단계 백업만들기(Creating the Stage 1 Back Up)

만약 데이터 백업을 이미 가지고 있다면, 분할 영역을 다시 만들 수 있도록 분할 영역 정보를 보존하는 것도 필요하다.

스크립트 `make.fdisk`는 하드 드라이브의 분할 영역의 정보를 조사하고 그것을 두 개의 파일에 저장한다. 하나는 실행 가능한 스크립트이고 `make.dev.x`로 불리운다. 다른 것은 **fdisk**로 분할 영역을 만드는데 필요한 명령어들이 들어 있고 `dev.x`로 불리운다.(여기서 `x`는 장치 파일의 이름이다. 예, hda). 여러분은`make.fdisk`에 연관된 장치 파일을 인수로 전달하게 함으로써, 어느 드라이브에서 작동하는 스크립트를 만들고 있는지를 명확하게 표시해야 한다. 예를 들어, 보통의 IDE 시스템에서는,

```
bash# make.fdisk /dev/hda
```

라고 입력하면, 스크립트 `make.dev.hda`와 **fdisk**를 위한 입력 파일, `dev.hda`가 만들어 진다.

덧붙여, 만약 `make.fdisk`가 FAT32나 혹은 FAT를 만나게 되면 그 분할영역의 부트 섹터를 `dev.xy`라고 이름 붙여진 파일에 보관한다. 여기서 x 는 드라이브의 장치 이름(예, sdc, hda)이고 y는 분할 영역의 번호이다. 부트 섹터는 분할 영역의 첫 번째 섹터, 512 바이트이다. 이 섹터는 `make.dev.hda` 스크립트에 의해서 파티션이 다시 만들어 질 때 동시에 복구 된다.

다행히도, 선거 이후에 정치인에 대한 대중의 신뢰도가 떨어지는 만큼이나 빠른 속도로 하드 드라이브의 가격이 수직 하락하고 있다. 그래서, 출력 파일이 텍스트이고 일일이 편집 가능하다는 것은 좋은 일이라고 할 수 있다. 지금 당장에는, 더 큰 대체 드라이브에 다시 복구를 하는 유일한 방법이다. (해야할 일들 목록을 보라.)

다른 메타데이터들은 `save.metadata` 스크립트에의해 보존된다. 그 스크립트는 ZIP 드라이브의 루트에 있는 fdisk.hda 파일 내에 분할 영역의 정보를 저장한다. 이 파일과 여러분의 `/etc/fstab`을 출력하는 것은 훌륭한 생각이며, 그 출력물을 이용해서 여러분은 분할영역 데이터를 여러분의 손으로 복구할 수 있을 것이다. 여러분은 두 개의 가상 콘솔을 토글하면서 하나는 **fdisk**를 실행시키고, 다른 하나에서는 `/etc/fstab`이나 `/fdisk.hda`를 출력하면서 위치 정보를 저장할 수 있을 것이다. 그러나, 그렇게 하는 것은 오류가 발생하기 쉽다.

여러분은 또한 복구 방법에 관련된 파일을 보존하기를 원할 것이다. 예를 들어, 데이터를 저장하기 위해서 nfs를 사용한다면, 여러분은 hosts.allow, hosts.deny, exports 등등의 파일도 보존할 필요가 있을 것이다. 또한, 여러분이 어떤 네트워크를 이용한 백업 시스템, Amanda나 Quick Restore을 사용한다면, 여러분은 HOSTNAME, hosts같은 네트워크 파일과 소프트웨어 위치 정보에 관련된 파일을 보존할 필요가 있게 될 것이다.

이것들 혹은 이와 비슷한 것들을 다루는 가장 간단한 방법은 etc 디렉토리 전체를 보존하는 것이다.

100MB ZIP 드라이브로는 요즘의 서버 설치용 리눅스 배포판을 담기는 힘들다. 그래서, 단순히 전체를 보존하기 보다는 좀더 선택적으로 골라야만 한다. 우리가 필요한 것은 무슨 파일인가?

- 부트 디렉토리.
- /etc 디렉토리와 하부 디렉토리들.
- 부트시에 필요한 디렉토리들
- /dev에 있는 장치 파일들.

부트시에 필요한 파일을 결정하기 위해서는, 부트 초기화 파일 /etc/rc.sysinit를 보면 된다. 거기에는 경로를 다음과 같이 설정하고 있다.

```
PATH=/bin:/sbin:/usr/bin:/usr/sbin
export PATH
```

시행착오를 거치면 /dev와 같은 다른 디렉토리를 필요로 한다는 것을 알 수 있다. 리눅스에서는 장치 파일 없이 아무 것도 할 수 없다.

`save.metadata`스크립트를 읽을 때, 절대 경로로 파일을 저장할 필요는 없다는 것을 알아두길 바란다.

제대로 작동하는 백업 스크립트를 얻기 위해서, 우리는 백업, bare metal restore시험, CD로부터 재설치 그리고 다시 시도하는 일련의 과정을 반복적으로 해야할 수도 있다. 내가 이 HOWTO에 대해 작업하는 동안, 나는 위와 같은 과정을 다섯 번 반복한 후 성공적으로 복구할 수 있었다. 이것이 비록 사용 가능할 때일지라도 왜 스크립트 사용에 주의해야 하는가에 대한 한가지 이유이다. 충분히 시험해보라.

4.1. 주제와 약간의 변화(Theme and Variations)

4.1.1. ZIP드라이브 없는 백업(No ZIP drive)

이 백업 과정은 여러분이 백업할 때 만 ZIP 디스크를 필요로 한다. 만약 충분히 하드 드라이브가 넉넉하다면, 여러분은 ZIP 디스크의 내용물을 디렉토리에 만들거나, 네트워크로 저장할 수 있다. 그후, 복구를 할 때 저장된 파일로 부터 ZIP 디스크를 만들면 된다.(**cp -rp**를 사용)

백업 과정이 더 빨라질 것이지만, 만든 파일이 ZIP 디스크에 적절히 들어가는지 점검해야 한다(**du -hs $target.zip**를 사용). 여러분은 `save.metadata`에 있는 변수 `zip`의 정의를 수정해야만 할 것이다.

내 랩탑은 네트워크 카드와 ZIP 드라이브를 동시에 실행시킬 수 없었기 때문에, 이 방법을 사용해서 백업을 한다.

또는, 하드 드라이브에 여러개의 ZIP 디스크를 만든 후에 복구 대상의 시스템으로 공급하는 방법이 있다.

4.1.2. 여러개의 ZIP디스크(Multiple ZIP disks)

2개의 1단계 스크립트,`restore.metadata`와`save.metadata`를 분리함으로써, 여러분은 1단계 metadata를 여러개의 ZIP 디스크에 나누어 둘 수 있다.

4.1.3. 1단계 저장물로부터 추출(Excluding From First Stage Saving)

1단계 저장 자료로 부터 몇 메가바이트가 안 되는 자료를 뽑아낼 필요가 있을 때, 특히 ZIP 디스크의 용량 제한에 맞춰야 할 때가 있다. `save.metadata` 스크립트에 있는 **crunch**의 역할은 **tar**에 여러개의 인수를 주는 것이다. **--exclude** 매개변수 또한 줄 수 있다. 그래서, 예를 들면 아래와 같이 해서 **emacs**와 **gs**를 추출할 수 있다.

```
crunch etc etc --exclude etc/samba --exclude X11
```

왜 그들 둘을 뽑아내는가? 왜냐하면, 그들이 하드 드라이브 공간만 많이 차지하고 부트 시에는 필요가 없기 때문이다.

어떻게 추출해 낼 후보를 잘 골라낼 수 있을까? 각 파일에 대해서는 **ls -alSr**를 이용하고, 디렉토리에 대해서는 **du | sort -n**로 나열해서 살핀다.

4.1.4. Initrd

만약 여러분의 시스템이 부트하기 위해 RAM 디스크나 initrd를 사용한다면, `save.metadata`가 `/initrd` 디렉토리를 만드는지 확인해라.

SCSI 드라이브로부터 부트한다면, 여러분의 시스템은 initrd를 사용하거나, 혹은 ext3fs 분할영역에 root를 가질 것이다. 호출하는 것을 알기 위해서는 `/etc/lilo.conf`를 점검하라.

5. 1단계 복구(First Stage Restore)

5.1. tomsrtbt로 부팅하기(Booting tomsrtbt)

복구를 진행하기 전에 첫째로 확인해야 할 것은 하드웨어의 시간이 바르게 설정되어 있는지 확인하는 것이다. BIOS 설정을 확인하기 바란다. 얼마나 시간을 정확히 설정해야 하는 지는 여러분의 응용프로그램(application)에 달려있다. 복구를 위해서는 정확한 시간에서 몇 분정도 차이나는 정도는 맞아야 한다. 이것이 여러분이 최종적으로 복구된 시스템을 시동시켰을 때 시간이 결정적으로 중요한 영향을 미치는 (time-critical) 일들이 제대로 돌아가게 만들 것이다.

tomsrtbt (http://www.toms.net/rb)로 부팅하기 전에, 여러분의 ZIP 드라이브가 병렬 포트, 혹은 /dev/lp0 나 /dev/lp1에 연결되었는지 확인하라. 시작 프로그램은 ZIP 드라이브를 별렬 포트에 작동하게 할 것이다.

다음 단계는 비디오 상태를 설정하는 것이다. 보통 나는 스크린에 가능한 많은 것이 보이는 것을 선호한다. 비디오 상태를 선택하는 옵션이 나올 때, 나는 보통 6번, 즉 60라인과 80열로 된 것을 사용한다. 여러분의 하드웨어는 고해상도를 지원하거나 못할 수 있으니 실험을 해보시라.

5.2. 복구(Restoration)

일단 tomsrtbt (http://www.toms.net/rb)가 부트되고 콘솔이 나타나면, ZIP 드라이브를 마운트하라. 다음과 같이 읽기 전용으로 마운트하는 것도 좋은 아이디어가 될 수 있다.

```
# mount /dev/sda1 /mnt -o ro
```

다음과 같이 제대로 존재하는지 확인하라:

```
# ls -l /mnt
```

이 시점에서, 여러분은 자동 혹은 수동으로 복구를 할 수 있을 것이다. 앞으로 어떤 변화를 줄 필요가 없다면, 자동화된 복구방법을 사용하라.

5.2.1. 자동 복구(Automated)

완전히 자동화된 1단계 복구를 하기위해서는 다음과 같은 명령어를 입력하라.

```
# /mnt/root.bin/first.stage
```

If you want to check for back blocks, add the **-c** option.

5.2.2. 수동 복구(manually)

수동복구를 진행하기 위해서는, ZIP드라이브에 있는 스크립트가 위치한 디렉토리를 바꾼다.

```
# cd /mnt/root.bin
```

이제 분할영역 정보를 복구할 스크립트를 실행하라. 예를 들면:

```
# ./make.dev.hda
```

손상된 블록에 대한 점검을 하기 원한다면, **-c** 옵션을 추가하라.

이 스크립트는 다음과 같은 것을 할 것이다:

- 기존의 분할 영역표(partition table)와 마스터 부트 레코드(Master Boot Record, MBR)을 삭제하면서, 하드 드라이브의 첫 번째 1024바이트를 깨끗이 만든다.
- 여러분이 `make.fdisk`를 실행했을 때 모은 정보로부터 분할 영역을 다시 만든다.
- ext2 파일 시스템 분할 영역과 적당한 리눅스 스왑 분할 영역을 만든다. 만약 **-c** 옵션을 넣는다면 손상된 블록에 대한 점검도 할 것이다.
- 몇몇 형태의 FAT 분할 영역을 만든다.
- 마운트 위치를 만들고 ext2 분할 영역을 마운트한다.

> 주의**:** 만약 여러분이 복구할 운영 체제가 다른 것이라면, 지금이 시도해 볼 좋은 때이다. 첫째로, 리눅스 복구를 끝내기 위해서 tomsrtbt (http://www.toms.net/rb)로 다시 부팅하라. 여러분은 지금 막 만든 분할영역을 다시 마운트 해야 할 것이다. `make.dev.x` 스크립트의 끝 부분에 분할 영역을 마운트하기 위한 새로운, 그리고 기존의 것과 독립적인 스크립트를 만들어라.

여러분이 모든 디렉토리를 만들고 분할 영역을 모두 마운트한 후, 여러분은 `restore.metadata` 스크립트를 실행시킬 수 있다. 이것은 ZIP 드라이브의 내용물을 하드드라이브로 복구할 것이다.

여러분은 ZIP 드라이브 루트 디렉토리 하부의 디렉토리들과 복구된 저장 파일의 목록을 살펴보아야 한다. tomsrtbt (http://www.toms.net/rb)에 대한 tar 파일로 부터 tar의 블록크기가 20이고 그걸로 충분하다는 것을 알 수 있다. lilo는 다음과 같은 결과를 표시해야 한다는 것을 명심하라.

```
Added linux *
```

위의 것은 "**df -m**" 명령어의 결과물에 뒤이어 표시될 것이다.

5.2.3. 마감하기(Finishing Touches)

만약 여러분이 부팅 후 바로 X window로 진행된다면, 몇몇 문제가 발생할 수 있다. 안전하게 하기 위해서, 여러분의 부트 레벨을 일시적으로 바꾸어라. `/target/etc/inittab`에 있는 다음과 같은 줄을 찾아서

```
id:5:initdefault:
```

다음과 같이 바꾸어라.

```
id:3:initdefault:
```

6. 2단계 복구(Second Stage Restoration)

컴퓨터가 다시 부팅하면, BIOS로 들어가서 시간이 어느 정도 정확한지 확인한다.

시간이 정확한지 확인하면, BIOS를 나가서 하드 드라이브로 부팅을 진행시킨다. 보통의 부팅 순서대로 하면 그렇게 된다. 이러면 여러분은 엄청나게 많은 오류 메시지를 만나게 될 것이다. 내용은 "~~~와 같은 것을 찾을 수 없습니다." 같을 것이다. 만약 여러분이 지금까지의 작업을 정확히 했다면, 오류 메시지는 중요하지 않다. 여러분은 할일을 하기 위해서 linuxconf나 apache가 필요하지 않다.

> 주의**:** 대안으로, 단독 사용자 모드로 부팅할 수 있다.(lilo의 프롬프트에서 linux single이라고 입력하라.) 그러나 여러분은 네트워크를 수동으로 설정해야 하고 sshd와 시스템 복구를 위해 필요한 daemon을 수동으로 가동시켜야 할 것이다. 이런 것을 어떻게 하는가 하는 것은 시스템에 따라서 매우 다르다.

여러분은 루트 콘솔로 로그인할 수 있어야 하며(X window 없이도, 다른 사용자 없이도), 네트워크, 예를 들어 여러분 시스템의 백업을 nfs로 마운트 할 수 있어야 한다.

만약 여러분이 내가 제안한 대로 Arkeia를 이용해서 두단계 백업을 했다면, 여러분은 Arkeia의 데이터베이스와 실행가능한 파일들(executables) 을 이용해서 복구를 할 수 있을 것이다. 또한

```
/etc/rc.d/init.d/arkeia start
```

를 실행시키고 서버를 시작할 수 있어야 한다. X를 이용해서 GUI가 설치된 다른 컴퓨터로 접속할 수 있다면, 여러분은 테이프 서버에 있는 Arkeia에 로그인 할 수 있고 복구를 준비할 수 있을 것이다.

> **주의:** 여러분이 복구할 때, 복구 프로그램의 문서를 주의깊게 읽어야 한다. 예를 들어, tar는 어떤 특성(suid bit를 가지는 등의)의 파일은 보통 복구하지 않는다. 파일 허가권은 사용자의 umask에 의해서 설정된다. 여러분이 저장한 그대로 복구하길 원한다면, tar의 p 옵션을 사용하라. 복구 소프트웨어가 여러분이 저장한 대로 모든 것을 정확히 복구했는지 확인하라.

시험 컴퓨터를 복구하기 위해서는 아래와 같이 입력하라.

```
bash# restore.all
```

만약 여러분이 백업과 복구에 tar를 사용했다면, -k 옵션(오래된 파일을 보존하고 덮어쓰지 말 것)을 사용하라. 그러면, 아래와 같은 것들을 많이 보게 될 것이다.

```
tar: usr/sbin/rpcinfo: Could not create file: File exists
tar: usr/sbin/zdump: Could not create file: File exists
tar: usr/sbin/zic: Could not create file: File exists
tar: usr/sbin/ab: Could not create file: File exists
```

tar가 일단계의 복구 동안에 복구한 파일을 덮어 쓸 것을 거부하고 있으므로, 이런 것들은 정상적인 현상이다.

이후, 다시 부팅하라. 종료하는 동안, “그런 pid가 없습니다.”와 같은 많은 오류 메시지를 보게 될 것이다. 이것도 정상적인 과정의 하나이다. 종료 코드는 지난번 부팅 때는 사용하지 않았던, 백업이 만들어 졌을 때 사용된 daemon으로부터 pid를 사용한다. 당연히 그런 pid는 없다.

이제 여러분의 시스템은 이전보다는 작지만 여전히 많은 오류 메시지를 가진 채로 부팅되고 실행되어야 한다. RPM에 기반한 시스템에서 여러분의 복구 작업이 얼마나 잘 되었는지 엄격히 시험하려면 모든 패키지를 확인해야 한다.

```
bash# rpm -Va
```

설정파일이나 로그파일 같은 것들은 정상적인 과정을 거쳐서 바뀌게 될 것이니, 여러분이 그런 것은 적당히 걸러서 이해할 수 있어야 한다. 출력을 파일로 돌려서 만들 수도 있고, 백업 때 만들어진 것과 현재 것을 diff로 비교할 수도 있다 (/etc/rpm.Va.txt). 그래서 이 과정을 상당히 빨리 처리할 수 있다. Emacs 사용자는 emacs에 포함된 diff 기능을 이용하라.

이제 응용 프로그램, 특히 daemon으로서 실행되는 것들을 시험해 볼 때다. 응용프로그램이 복잡하면 할 수록, 더 많은 시험이 필요할 것이다. 만약 원격 사용자가 있다면, 시험하고 있는 중에는 그들이 시스템을 못 사용하게 막거나 “읽기 전용”으로만 사용하게 하라. 이것은 데이터베이스의 경우에는 아주 중요한데, 자료의 상실이나 망가짐을 막기 위해서이다.

만약 보통 X로 부팅하고 위에 한 대로 X를 사용하지 못하게 했다면, 다시 사용 가능한 상태로 하기 전에 X를 먼저 시험해 보라. X를 다시 사용하는 것은 /etc/inittab에 있는 다음과 같은 줄을 찾아서 아래와 같이 하면 된다.

```
id:5:initdefault:
```

이제 신나게 즐길 준비가 되었을 것이다.----아니면 아스피린과 소파가 필요할 지도.(성공하거나 실패하거나에 대한 비유인 것 같네요...^^)

7. 배포판별 주의 사항(Distribution Specific Notes)

아래에 나오는 것은 과거 경험에서 나온 주의 사항입니다. 만약 여러분이 다른 배포판에 대한 추가 사항이 있다면 저에게 알려주십시오.

7.1. Red Hat 7.1

이 배포판은 내가 시험용 컴퓨터에 사용한 유일한 배포판이다. 전혀 문제가 없었다.

7.2. Red Hat 7.0

이 버전은 사용자를 인증하기 위해서 libcrack(/usr/lib에 있음)과 부수적인 파일을 필요로 하는 것 같다. 그래서 `save.metadata`에 /usr/lib를 저장하는 줄에 /usr/lib/*crack*을 추가해서 사용하라.

8. 응용프로그램별 주의사항(Application Specific Notes)

특별한 응용 프로그램을 사용해서 백업을 할 경우의 주의사항을 나열하였다.

8.1. Squid

Squid는 http proxy와 cache이다. 그 자체로 수 많은 일시적 자료를 하드 드라이브에 간직하고 있다. 두 번째 단계의 백업 스크립트 중 적절한 tar 명령어에 다음을 추가한다. “--exclude /var/spool/squid”. 그러면, squid로 하여금 여러분을 위하여 적절히 디렉토리를 재구성하도록 할 것이다. 두 번째 단계의 스크립트 끝 부분에 squid를 초기화하는 명령어를 삽입하라. 여기에 내가 ssh을 이용해서 `restore.tester`에 어떻게 하였는지를 설명했다.

```
ssh $target "mkdir /var/spool/squid ; chown squid:squid /var/spool/squid;\
      /usr/sbin/squid -z;touch /var/spool/squid/.OPB_NOBACKUP"
```

마지막 명령어는 .OPB_NOBACKUP이라는 0 byte의 크기를 갖는 파일을 생성한다. 이것은 Arkeia를 위한 것으로써, Arkeia로 하여금 그 이하 디렉토리는 백업하지 않도록 한다.

8.2. Arkeia

이 주의 사항들은 Arkeia 4.2를 시험한 것에 기반하고 있다.

Arkeia (http://www.arkeia.com/)는 다양한 플랫폼에서 가동되는 백업과 복구 프로그램이다. 여러분은 Arkeia를 Bare Metal Recovery의 일부로 사용할 수 있지만, 여기에는 두 가지 경고할 것이 있다.

첫째는 아마 가장 문제가 될 것인데, 쓸만한 해결책이 없기 때문에, 여러분은 복구시에 탐색기(navigator)에서 복구할 디렉토리를 일일이 손으로 선택해 주어야 할 것이다. 그 이유는 **tar**의 -p 선택사항과 비슷하게, Arkeia는 복구시에 디스크에 이미 존재하는 파일에 대해 복구하지 않는 메카니즘이 존재하지 않는다. 만약 단순하게 전체 복구를 허용한다면, 복구 시에 사용하고 있는 라이브러리(예를 들면, lib/libc-2.1.1.so)를 덮어 써 버릴 것이고 복구를 망쳐버릴 것이다.손으로 디렉토리를 선택하는 것이 가장 좋다.

두 번째 경고 사항은 Arkeia의 자료 사전(data dictionary)와 프로그램도 백업을 해야 한다는 것이다. 그러기 위해서는 `save.metatdata` 스크립트를 변경해서 Arkeia를 저장하도록 한다.

```
# arkeia specific:
tar cf - usr/knox | gzip -c > $zip/arkeia.tar.gz
```

Arkeia는 자료사전(data dictionary)을 백업하지 않기 때문에, 여러분은 반드시 이 방법으로 자료 사전(data dictionary)을 백업해야 한다. 이것이 내가 Arkeia에 가지는 불만 사항 중 하나이며 나는 이것을 The TOLIS Group's BRU (http://www.estinc.com/)를 이용하여 tape에 저장하는 것으로 해결하였다.

자료 사전(data dictionary)은 `restore.metadata` 스크립트로 자동으로 복구될 것이다.

9. 재해 복구에 대한 몇가지 조언(Some Advice for Disaster Recovery)

여러분은 각각의 컴퓨터에 대한 ZIP disk와 여러분이 만든 출력물들을 안전한 곳에 보관해야 한다. 또한 이것들의 복사본을 다른 백업 저장소에 보관해야 한다. 다른 장소에 보관하는 주된 이유는 재해 복구를 가능하게 하는 것이고, 하드웨어를 교체한 각각의 호스트를 복구하는 것은 재해 복구의 한 부분이다.

여러분은 몇 개의 tomsrtbt (http://www.toms.net/rb) 플로피와 ZIP 드라이브 또한 다른 저장소에 보관해야 할 것이다.

여러분은 아마도 이 HOWTO 또한 몇 부를 복사해야 할 것이며, 여러분의 사이트에만 특별히 적용되는 여러 해설과 주석들을 첨부해서 보관해야 할 것이다.

10. 이제는 무엇을?(What Now?)

이 HOWTO는 한 컴퓨터에 대한 실험으로부터 나온 결과이다. 당연히 첫 번째 백업 단계에서 백업이 필요한 디렉토리와 파일을 알 수 있을 것이다. 나는 첫 단계에서 X를 저장하고 복구하는 것은 다루지 않았다. 또한 Intel 이 외의 프로세서에 대해서도 전혀 다루지 않았다.

여러분이 자신의 컴퓨터에 이 스크립트를 시험하고 개선시켜 피드백을 준다면 감사하겠다. 또한 백업 소프트웨어 판매회사가 자신의 제품의 최소 백업을 어떻게 할 것인가를 문서화해 준다면 고맙겠다. 또한 나는 리눅스 공동체가 밤에 조금 더 편히 잠드는 것을 보고 싶다.

10.1. 해야할 일들(To Do)

자원자를 환영한다. 시작하기 전에 다른 이가 이미 작업하고 있지 않은지 나와 함께 점검해야 할 것이다.

- 분할영역 편집기- 다른 하드드라이브의 분할영역 경계를 조정하거나, 다른 geometry에 대해 같은 작업을 하는 것. 또는, 하나의 하드 드라이브 내에서 분할영역의 크기를 조절하는 것. GUI는 아마도 좋은 생각일 것이다. FSF의 `parted` (http://www.gnu.org/software/parted)는 또한 하나의 대안이 될 것으로 보인다. 그것은 기존의 분할영역의 크기를 재조정할 수 있으나 몇 가지 제약이 있다.
- `make.fdisk`는 현재 하나의 스크립트이다. 마운트 명령어를 다른 스크립트로 분리해서 `make.dev.hda` 같이 실행가능하게 만들어야 한다. 그리고 몇 가지 장난을 칠 수 있게 리부팅한다(내가 한번도 들어보지 못한 OS를 위한 분할 영역을 만든다 든지). 또는 `parted` (http://www.gnu.org/software/parted)를 실행시킨 후 tomsrtbt (http://www.toms.net/rb)를 실행하기 위해 리부팅하고 모든 리눅스 분할 영역을 마운트하고 계속한다.
- `make.fdisk`는 현재 몇몇 FAT 분할 영역만을 인식한다. 그래서, 다른 것들도 인식 가능하게 `make.fdisk`에 코드를 추가하거나 출력 파일에서 그들을 재구성하도록 적당한 사용 설명을 만든다.
- ext3fs에 대해서도 유사한 일들. ext3fs는 ext2의 저널링 버전이다.
- FAT12나 FAT16 분할영역에 대해서는 포맷을 실행하지 않으며 분할 영역 내부로 아무것도 써 넣지 않는다. 그래서 DOS 6.x에는 혼란을 주지 않는다. 문제에 대한 설명을 위해서는 **fdisk**에 대한 주의 사항을 보라.
- The second stage backup and restore scripts currently use gzip for compression. They could benefit from using bzip2 instead. This change will require saving the libbz2* libraries in /usr/lib instead of libz.
- ext2/3 파일 시스템으로 ZIP 디스크에 쓰는 스크립트.
- 인간이 사용하는 다른 언어로의 번역
- loadlin이나 비슷한 프로그램이 어떤 영향을 미치는지 조사
- GRUB을 위한 수정
- CD-ROM을 사용하기 위한 스크립트의 변화. tomsrtbt (http://www.toms.net/rb)로 부팅이 가능하고 그 나머지 부분에는 일단계 복구 자료를 포함하고 있는 CD-ROM이 만들어 지는 것.

11. 스크립트들(The Scripts)

스크립트가 무엇을 하는가에 대한 설명은 스크립트의 시작 부분의 주의 사항을 보라.

11.1. 1단계(First Stage)

11.1.1. `make.fdisk`

백업시 실행되는 이 스크립트는 복구시에도 실행되어 아래의 `make.dev.hda`와 비슷한 스크립트를 만들어 낸다. 이것은 아래의 `dev.hda`와 유사한 자료 파일도 만들어 낸다. 스크립트의 이름과 만들어지는 자료 파일의 이름은 이 스크립트에 매개변수로 주어지는 장치에 따라 달라진다. 복구시에 실행되는 그 스크립트는 하드드라이브에 분할영역을 새로이 만들어 낸다. `make.fdisk`는 아래의 `save.metadata`로부터 이름이 유래했다.

```
#! /usr/bin/perl

# A perl script to create a script and input file for fdisk to
# re-create the partitions on the hard disk, and format the Linux and
# Linux swap partitions. The first parameter is the fully qualified
# path of the device of the hard disk, e.g. /dev/hda. The two
# resulting files are the script make.dev.x and the data file dev.x
# (where x is the hard drive described, e.g. hda, sdc). make.dev.x is
# run at restore time to rebuild hard drive x, prior to running
# restore.metadata. dev.x is the input file for fdisk.

# Time-stamp: <2002-09-10 10:18:44 root make.fdisk>

# Copyright 2001 through the last date of modification Charles Curley
# except for the subroutine cut2fmt.

# cut2fmt Copyright (c) 1998 Tom Christiansen, Nathan Torkington and
# O'Reilly & Associates, Inc.  Permission is granted to use this code
# freely EXCEPT for book publication.  You may use this code for book
# publication only with the explicit permission of O'Reilly &
# Associates, Inc.

# This program is free software; you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by the
# Free Software Foundation; either version 2 of the License, or (at your
# option) any later version.

# This program is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
# General Public License for more details.

# You should have received a copy of the GNU General Public License along
```

```
# with this program; if not, write to the Free Software Foundation, Inc.,
# 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

# In addition, as a special exception, Tom Christiansen, Nathan
# Torkington and O'Reilly & Associates, Inc.  give permission to use
# the code of this program with the subroutine cut2fmt (or with
# modified versions of the subroutine cut2fmt that use the same
# license as the subroutine cut2fmt), and distribute linked
# combinations including the two.  You must obey the GNU General
# Public License in all respects for all of the code used other than
# the subroutine cut2fmt.  If you modify this file, you may extend
# this exception to your version of the file, but you are not
# obligated to do so.  If you do not wish to do so, delete this
# exception statement and the subroutine cut2fmt from your version.

# You can also contact the Free Software Foundation at http://www.fsf.org/

# Changes:

# 2002 09 08: Added minimal support for ext3fs. We now detect mounted
# ext3fs partitions & rebuild but with no options. The detection
# depends on the command line "dumpe2fs <device> 2>/dev/null | grep -i
# journal" producing no output for an ext2fs, and output (we don't
# care what) for an ext3fs.

# This could stand extension to support non-default ext3 options such
# as the type of journaling. Volunteers?

# 2002 07 25: Bad block checking is now a command line option (-c) at
# the time the product script is run.

# 2002 07 03: Corrected the mechanism for specifying the default
# drive.

# 2001 11 25: Changed the way mke2fs gets its bad block
# list. badblocks does not guess at the block size, so you have to get
# it (from dumpe2fs) and feed it to badblocks. It is simpler to just
# have mke2fs call badblocks, but you do loose the ability to have a
# writing test easily. -- C^2

# 2001 11 25: Changed the regex that extracts partition labels from
# the mount command. This change does not affect the results at all,
# it just makes it possible to use Emacs' perl mode to indent
# correctly. I just escaped the left bracket in the regex. -- C^2

# Discussion:

# fdisk will spit out a file of the form below if you run it as "fdisk
# -l".
```

```
# root@tester ~/bin $ fdisk -l /dev/hda

# Disk /dev/hda: 64 heads, 63 sectors, 1023 cylinders
# Units = cylinders of 4032 * 512 bytes

#    Device Boot    Start       End    Blocks   Id  System
# /dev/hda1             1         9     18112+  83  Linux
# /dev/hda2            10      1023   2044224    5  Extended
# /dev/hda5            10       368    723712+  83  Linux
# /dev/hda6           369       727    723712+  83  Linux
# /dev/hda7           728       858    264064+  83  Linux
# /dev/hda8           859       989    264064+  83  Linux
# /dev/hda9           990      1022     66496+  82  Linux swap

# What fdisk does not do is provide output suitable for later
# importing into fdisk, a la sfdisk. This script parses the output
# from fdisk and creates an input file for fdisk. Use the input file
# like so:

# fdisk /dev/hdx < dev.hdx

# For the bare metal restore package, this script also builds a script
# that will execute the above command so you can run it from your zip
# disk. Because the bare metal restore scripts all are in /root/bin,
# the data file and script created by this script are also placed
# there. The same script also creates appropriate Linux file systems,
# either ext2fs, or Linux swap. There is limited support for FAT12 and
# FAT16. There is no support right now (hint, hint) for FAT32. For
# anything else, you’re on your own.

# Note for FAT32: According to the MS KB, there are more than one
# reserved sectors for FAT32, usually 32, but it can vary. Do a search
# in M$’s KB for "boot sector" or BPB for the gory details. For more
# info than you really need on how boot sectors are used, see
# http://support.microsoft.com/support/kb/articles/Q140/4/18.asp

# You can also edit dev.x to change the sizes of partitions. Don’t
# forget, if you change the size of a FAT partition across the 32MB
# boundary, you need to change the type as well! Run "fdisk /dev/hda"
# or some such, then the l command to see the available partition
# types. Then go ahead and edit dev.x appropriately. Also, when moving
# partition boundarys with hand edits, make sure you move both logical
# and extended partition boundaries appropriately.

# Bad block checking right now is a quick read of the partition. A
# writing check is also possible but more difficult. You have to run
# badblocks as a separate command, and pass the bad block list to
# mke2fs in a file (in /tmp, which is a ram disk). You also have to
```

```
# know how large the blocks are, which you learn by running
# dumpe2fs. It gets messy and I haven’t done it yet. You probably
# don’t need it for a new hard drive, but if you have had a hard drive
# crash on you and you are reusing it (while you are waiting for its
# replacement to come in, I presume), then I highly recommend it. Let
# me know how you do it.

# For more information contact the author, Charles Curley, at
# http://w3.trib.com/~ccurley/.


# cut2fmt figures out the format string for the unpack function we use
# to slice and dice the output from fdisk. From Christiansen and
# Torkington, Perl Cookbook 5.

sub cut2fmt {
    my (@positions) = @_;
    my $template    = ”;
    my $lastpos     = 1;

    foreach $place (@positions) {
$template .= "A" . ($place - $lastpos) . " ";
$lastpos = $place;
    }

    $template .= "A*";
    return $template;
}


# Begin main line code.



# Provide a default device.

# print "\$ARGV[0] is $ARGV[0].\n";

$device = defined ($ARGV[0]) ? $ARGV[0] : "/dev/hda";

# print "Device is $device.\n";

# prepare format string.
$fmt = cut2fmt (11, 19, 24, 34, 45, 49);
# print "Format string is $fmt.\n\n";

# define fields in the array @_.
$dev = 0;
$bootable = 1;
```

```
$firstcyl = 2;
$lastcyl = 3;
$parttype = 5;
$partstring = 6;

$target = "\/target";

$outputfilename = $device;
$outputfilename =~ s/\//./g;
$outputfilename = substr ($outputfilename, 1, 100);

$outputfilepath = "\/root\/bin\/";


# Make a hash of the labels.
$mpid = open (MOUNT, "mount -l |") or die "Couldn't fork: $!\n";
while (<MOUNT>) {
    if ($_ =~ /^$device/i) { # is this a line with a partition in it?
#      print $_; # print it just for grins
split;
if ($_[6] ne "") { # only process if there actually is a label
    $_[6] =~ s/[\[\]]//g; # strike [ and ].
    $labels{$_[0]} = $_[6];
#     print "The label of file device $_[0] is $labels{$_[0]}.\n";
}


# We only mount if it's ext2fs or ext3fs and read and write.

if ($_[4] =~ /ext[23]/ and $_[5] eq "(rw)" ) {
            $mountpoints{$_[2]} = $_[0];
#           print "$_[2] is the mountpoint for $mountpoints{$_[2]}.\n";
}
    }
}
close (MOUNT);


$fpid = open (FDISK, "fdisk -l $device |") or die "Couldn't fork: $!\n";

open (OUTPUT, "> $outputfilepath${outputfilename}")
    or die "Couldn't open out-
put file $outputfilepath${outputfilename}.\n";

while (<FDISK>) {
    if ($_ =~ /^$device/i) { # is this a line with a partition in it?
#    print $_; # print it just for grins
chop; # kill trailing \r
@_ = unpack ($fmt, $_);
```

```
# now strip white spaces from cylinder numbers
@_[$firstcyl] =~ s/[ \t]+//;
@_[$lastcyl] =~ s/[ \t]+//;
@_[$parttype] =~ s/[ \t]+//;

$partnumber = substr(@_[$dev], 8, 10); # get partition num-
ber for this line
# just for grins
#   print "  $partnumber, @_[$firstcyl], @_[$lastcyl], @_[$partstring]\n";

# Here we start creating the input to recreate the partition
# this line represents.

print OUTPUT "n\n";
if ($partnumber < 5) {
    # primary Linux partition
    if (@_[$parttype] == 83) {
print OUTPUT "p\n$partnumber\n@_[$firstcyl]\n";
if (@_[$firstcyl] ne @_[$lastcyl]) { # in case it’s all on one cylinder
    print OUTPUT "@_[$lastcyl]\n";
}

# Now detect if this is an ext3 (journaling)
# partition. We do this using dumpe2fs to dump the
# partition and grepping on "journal". If the
# partition is ext2, there will be no output. If it is
# ext3, there will be output, and we use that fact to
# set a command line switch. The command line switch
# goes into an associative array (hash) so we don’t
# have to remember to reset it to the null string when
# we’re done.

$dpid = open (DUMPE2FS, "dumpe2fs @_[$dev] 2>/dev/null | grep -
i journal |")
    or die "Couldn’t fork: $!\n";
while (<DUMPE2FS>) {
#      print "Dumpe2fs: $_";
    $ext3{$_[$dev]} = "-j ";
    last;
}
close (DUMPE2FS);

if ($labels{@_[$dev]}) { # do we have a label?
    $format .= "echo\necho formatting $checking@_[$dev]\n";
    $format .= "mke2fs $ext3{$_[$dev]}\$blockcheck -
L $labels{@_[$dev]} @_[$dev]\n\n";
} else {
    $format .= "echo\necho formatting $checking@_[$dev]\n";
```

```
    $format .= "mke2fs $ext3{$_[$dev]}\$blockcheck @_[$dev]\n\n";
}

# extended partition
    } elsif (@_[$parttype] == 5) {
print OUTPUT "e\n$partnumber\n@_[$firstcyl]\n";
if (@_[$firstcyl] ne @_[$lastcyl]) {
    print OUTPUT "@_[$lastcyl]\n";
}

# primary Linux swap partition
    } elsif (@_[$parttype] == 82) {
print OUTPUT "p\n$partnumber\n@_[$firstcyl]\n";
if (@_[$firstcyl] ne @_[$lastcyl]) {
    print OUTPUT "@_[$lastcyl]\n";
}
print OUTPUT "t\n$partnumber\n82\n";
$format .= "echo Making @_[$dev] a swap partition.\n";
$format .= "mkswap \$blockcheck @_[$dev]\n\n";

# primary mess-dos partition. We don’t handle FAT32,
# which requires a command line switch for mkdosfs.
    } elsif ( @_[$parttype] == 1 || @_[$parttype] == 4 || @_[$part-
type] == 6 ) {
print ("dd if=@_[$dev] of=$outputfilepath$outputfilename$partnumber");
print (" bs=512 count=1\n");
system ("dd if=@_[$dev] of=$outputfilepath$outputfilename$partnumber bs=512 count=1");
print OUTPUT "p\n$partnumber\n@_[$firstcyl]\n";
if (@_[$firstcyl] ne @_[$lastcyl]) { # in case it’s all on one cylinder
    print OUTPUT "@_[$lastcyl]\n";
}
print OUTPUT "t\n$partnumber\n@_[$parttype]\n";
$format .= "echo\necho formatting $checking@_[$dev]\n";
$format .= "mkdosfs \$blockcheck @_[$dev]\n";
$format .= "# restore FAT boot sector.\n";
$format .= "dd if=$outputfilename$partnumber of=@_[$dev] bs=512 count=1\n\n";

    } else {
# anything else partition
print OUTPUT "p\n@_[$firstcyl]\n";
if (@_[$firstcyl] ne @_[$lastcyl]) {
    print OUTPUT "@_[$lastcyl]\n";
}
print OUTPUT "t\n$partnumber\n@_[$parttype]\n";
    }

} else {
    # logical Linux partition
    if (@_[$parttype] == 83) {
```

```
print OUTPUT "l\n@_[$firstcyl]\n";
if (@_[$firstcyl] ne @_[$lastcyl]) {
    print OUTPUT "@_[$lastcyl]\n";
}

# Now detect if this is an ext3 (journaling)
# partition. We do this using dumpe2fs to dump the
# partition and grepping on "journal". If the
# partition is ext2, there will be no output. If it is
# ext3, there will be output, and we use that fact to
# set a command line switch. The command line switch
# goes into an associative array (hash) so we don’t
# have to remember to reset it to the null string when
# we’re done.

$dpid = open (DUMPE2FS, "dumpe2fs @_[$dev] 2>/dev/null | grep -
i journal |")
    or die "Couldn’t fork: $!\n";
while (<DUMPE2FS>) {
#      print "Dumpe2fs: $_";
    $ext3{$_[$dev]} = "-j ";
    last;
}
close (DUMPE2FS);

if ($labels{@_[$dev]}) { # do we have a label?
    $format .= "echo\necho formatting $checking@_[$dev]\n";
    $format .= "mke2fs $ext3{@_[$dev]}\$blockcheck -
L $labels{@_[$dev]} @_[$dev]\n\n";
} else {
    $format .= "echo\necho formatting $checking@_[$dev]\n";
    $format .= "mke2fs $ext3{@_[$dev]}\$blockcheck @_[$dev]\n\n";
}

# logical Linux swap partition
    } elsif (@_[$parttype] == 82 ) {
print OUTPUT "l\n@_[$firstcyl]\n";
if (@_[$firstcyl] ne @_[$lastcyl]) {
    print OUTPUT "@_[$lastcyl]\n";
}
print OUTPUT "t\n$partnumber\n82\n";
$format .= "echo Making @_[$dev] a swap partition.\n";
$format .= "mkswap \$blockcheck @_[$dev]\n\n";

# primary mess-dos partition. We don’t handle FAT32,
# which requires a command line switch for mkdosfs.
    } elsif ( @_[$parttype] == 1 || @_[$parttype] == 4 || @_[$part-
type] == 6 ) {
print ("dd if=@_[$dev] of=$outputfilepath$outputfilename$partnumber");
```

```
print (" bs=512 count=1\n");
system ("dd if=@_[$dev] of=$outputfilepath$outputfilename$partnumber bs=512 count=1");
print OUTPUT "p\n$partnumber\n@_[$firstcyl]\n";
if (@_[$firstcyl] ne @_[$lastcyl]) { # in case it's all on one cylinder
    print OUTPUT "@_[$lastcyl]\n";
}
print OUTPUT "t\n$partnumber\n@_[$parttype]\n";
$format .= "echo\necho formatting $checking@_[$dev]\n";
$format .= "mkdosfs \$blockcheck @_[$dev]\n";
$format .= "# restore FAT boot sector.\n";
$format .= "dd if=$outputfilename$partnumber of=@_[$dev] bs=512 count=1\n\n";

    } else {
# anything else partition
print OUTPUT "l\n@_[$firstcyl]\n";
if (@_[$firstcyl] ne @_[$lastcyl]) {
    print OUTPUT "@_[$lastcyl]\n";
}
print OUTPUT "t\n$partnumber\n@_[$parttype]\n";
    }

}

# handle bootable partitions
if (@_[$bootable] =~ /\*/) {
    print OUTPUT "a\n$partnumber\n";
}
    }
}

print OUTPUT "v\nw\n";

close (OUTPUT);
close (FDISK);


open (OUTPUT, "> ${outputfilepath}make.$outputfilename")
    or die "Couldn't open out-
put file ${outputfilepath}make.$outputfilename.\n";

print OUTPUT <<FINIS;
#! /bin/sh

# A script to restore the partition data of a hard drive and format
# the partitions. Created at bare metal backup time by the Perl script
# make.fdisk.

# Copyright 2001 through the last date of modification Charles Curley.
```

```
# This program is free software; you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by the
# Free Software Foundation; either version 2 of the License, or (at your
# option) any later version.

# This program is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
# General Public License for more details.

# You should have received a copy of the GNU General Public License along
# with this program; if not, write to the Free Software Foundation, Inc.,
# 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

# You can also contact the Free Software Foundation at http://www.fsf.org/

# For more information contact the author, Charles Curley, at
# http://w3.trib.com/~ccurley/.

export blockcheck=\$1;

if [ "\$blockcheck" != "-c" ] && [ -n "\$blockcheck" ]
then
    echo "\${0}: automated restore with no human interaction."
    echo "\${0}: -c: block check during file system making."
    exit 1;
fi

FINIS

# Clean the old partition table out.
print OUTPUT "dd if=/dev/zero of=$device bs=512 count=2\n\nsync\n\n";

print OUTPUT "fdisk $device \< $outputfilename\n\nsync\n\n";
print OUTPUT $format;

# Now build the mount points on the root and other partitions.

# We have a hash of mount points and devices in %mountpoints. However,
# we have to process them such that directories are built on the
# appropriate target partition. E.g. where /usr/local is on its own
# partition, we have to mount /usr before we build /usr/local. We can
# ensure this by sorting them. Shorter mount point paths will be built
# first. We can’t sort a hash directly, so we use an array.

# We build commands to create the appropriate mount points and then
# mount the partitions to the mount points. This is in preparation for
# untarring the contents of the ZIP disk, done in restore.metadata.
```

```
foreach $point ( sort keys %mountpoints) {
    print OUTPUT "\n# $point is the mount-
point for $mountpoints{$point}.\n";
    print OUTPUT "mkdir $target$point\n";
    print OUTPUT "mount $mountpoints{$point} $target$point\n";
}


close (OUTPUT);

# These scripts are dangerous & should only be visible to root.

chmod 0700, "${outputfilepath}make.$outputfilename";
chmod 0600, "${outputfilepath}$outputfilename";
```

11.1.2. `make.dev.hda`

이 스크립트는 위의 `make.fdisk`에 의해서 만들어지는 파일 중 하나의 예이다. 자료 파일을 `dev.hda`와 같이 사용하며, 분할영역을 만들고 파일시스템을 만든다. 복구시에 가장 처음으로 실행되는 스크립트이다.

만약 여러분이 예를 들어 새 분할 영역을 추가하기 위해서 `dev.hda`를 편집할 만큼 용기가 있어 시도한다면, 이 스크립트 또한 편집해야 한다.

만약 여러분이 분할 영역에 대해 파일 시스템을 쓰려할 때 손상된 블록에 대하여 점검하기를 원한다면, "-c" 명령 옵션을 사용하라.

```
#! /bin/sh

# A script to restore the partition data of a hard drive and format
# the partitions. Created at bare metal backup time by the Perl script
# make.fdisk.

# Copyright 2001 through the last date of modification Charles Curley.

# This program is free software; you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by the
# Free Software Foundation; either version 2 of the License, or (at your
# option) any later version.

# This program is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
# General Public License for more details.

# You should have received a copy of the GNU General Public License along
# with this program; if not, write to the Free Software Foundation, Inc.,
# 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
```

```
# You can also contact the Free Software Foundation at http://www.fsf.org/

# For more information contact the author, Charles Curley, at
# http://w3.trib.com/~ccurley/.

dd if=/dev/zero of=/dev/hda bs=512 count=2

sync

fdisk /dev/hda < dev.hda

sync

echo
echo formatting and checking /dev/hda1
mkdosfs -c /dev/hda1
# restore FAT boot sector.
dd if=dev.hda1 of=/dev/hda1 bs=512 count=1

echo
echo formatting and checking /dev/hda2
badblocks -c 128 -o /tmp/dev.hda2 /dev/hda2
mke2fs -L /boot -l /tmp/dev.hda2 /dev/hda2

echo
echo formatting and checking /dev/hda5
badblocks -c 128 -o /tmp/dev.hda5 /dev/hda5
mke2fs -L / -l /tmp/dev.hda5 /dev/hda5

mkswap -c /dev/hda6



# / is the mountpoint for /dev/hda5.
mkdir /target/
mount /dev/hda5 /target/

# /boot is the mountpoint for /dev/hda2.
mkdir /target/boot
mount /dev/hda2 /target/boot
```

11.1.3. `dev.hda`

이 자료 파일은 복구시에 사용된다. `make.dev.hda` 스크립트에 의해 **fdisk**로 연결된다. 이 파일은 백업시에 `make.fdisk` 스크립트에 의해서 만들어진다. **fdisk**에 익숙한 사람들은 내용이 **fdisk**의 명령어나 사용되는 값(실린더 수 같은)이라는 것을 알 수 있을 것이다. 그러므로, 이

파일을 편집함으로써, 분할 영역의 크기를 바꾸고 새 분할영역을 더하는 것이 가능하다. 그래서 마지막에서 두 번째 명령어가 하드 드라이브에 쓰기 전에 분할 영역을 확인하는 것, **v**가 되는 것이다.

```
n
p
1
1
29
t
1
6
a
1
n
p
2
30
44
n
e
3
45
1023
n
l
45
944
n
l
945
1023
t
6
82
v
w
```

11.1.4. `save.metadata`

이 스크립트는 백업의 한 부분으로 실행되는 첫 스크립트이다. 이것은 위의 `make.fdisk`를 호출한다. 만약 여러분이 백업할 것이 SCSI 하드 드라이브이거나 혹은 여러 개의 하드 드라이브라면, 적절하게 `make.fdisk`를 호출하도록 이 스크립트를 편집하라.

```
#! /bin/sh

# A script to save certain meta-data off to the boot partition. Useful for
```

```
# restoration.

# Time-stamp: <2002-08-23 16:30:05 ccurley save.metadata>

# Copyright 2000 through the last date of modification, Charles Curley.

# This program is free software; you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by the
# Free Software Foundation; either version 2 of the License, or (at your
# option) any later version.

# This program is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
# General Public License for more details.

# You should have received a copy of the GNU General Public License along
# with this program; if not, write to the Free Software Foundation, Inc.,
# 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

# You can also contact the Free Software Foundation at http://www.fsf.org/


# 2002 07 01: Went to bzip2 to compress the archives, for smaller
# results. This is important in a 100MB ZIP disk. Also some general
# code cleanup.

# 2002 07 01: The function crunch will tar and BZIP2 the
# archives. This is cleaner than the old code, and has better safety
# checking.


# For more information contact the author, Charles Curley, at
# http://w3.trib.com/~ccurley/.


# Crunch: A function to compress the contents of a directory and put
# the archive onto the ZIP disk.

# The first parameter is the name of the archive file to be
# created. The backup location, $zip, will be prepended and the
# extension, "tar.bz2" will be appended.

# All following parameters will be taken as additional directories or
# files to be put into the archive.

function crunch {
```

```
if [ -z "$1" ] || [ -z "$2" ]
# Checks if parameter #1 or #2 is zero length.
then
   echo "-Parameter #1 or #2 is missing.-
"  # Also if no parameter is passed.
   return 1
else
   local file=$1 # The archive file to create
   shift # Discard the file name
   local dirs=$@ # The director[y|ies] to archive
   local tarcmd="tar -cjf" # The tar command.

   $tarcmd  $zip/$file.tar.bz2 $dirs # do it!!

   error=$? # Preserve the exit code

   if [ $error != 0 ] # Did we fail?
   then # Yes
      echo "Tar failed with error $error"
      echo $tarcmd $zip/$file.tar.bz2 $dirs
      exit $error # return tar's exit code as ours
   fi

   return 0 # For error testing if needed.
fi
}

# Begin the main line code
export zip="/mnt/zip"; # Where we will put archives
#  export save="/mnt/save";

echo "saving hard drive info"
make.fdisk /dev/hda

# back up RPM metadata

echo "Verifying RPMs."

rpm -Va > /etc/rpmVa.txt

echo "Finished verifying RPMs; now mounting the ZIP drive."

# Make sure we have the ZIP drive mounted.
umount $zip
modprobe ppa # Driver for 100MB parallel port ZIP disk
mount /dev/sda1 # It should have ext2fs on partition 1.

# clean it all out
rm -r $zip/*
```

```
mkdir $zip/lost+found

echo "Building the ZIP drive backups."

# These are in case we need to refer to them while rebuilding. The
# rebuilding process should be mostly automated, but you never
# know....

fdisk -l /dev/hda > $zip/fdisk.hda

ls -al /mnt > $zip/ls.mnt.txt
ls -al / > $zip/ls.root.txt

mkdir $zip/etc;
cp -p /etc/* $zip/etc

cd /

# Build our minimal archives on the ZIP disk. These appear to be
# required so we can restore later on.

crunch boot boot
crunch root root
crunch etc etc
crunch lib lib

crunch usr.sbin usr/sbin
crunch usr.bin usr/bin
crunch sbin sbin
crunch bin bin
crunch dev dev

# Now optional saves.

# arkeia specific:
# crunch arkeia usr/knox

# save these so we can use ssh for restore. *crack* for RH 7.0 login
# authentication.
crunch usr.lib usr/lib/*crack* usr/lib/libz* usr/lib/libssl* usr/lib/libcrypto*

# save the scripts we used to create the ZIP disk and the ones we will
# use to restore it.
mkdir $zip/root.bin
cp -p /root/bin/* $zip/root.bin
rm $zip/root.bin/*~ $zip/root.bin/#*#

echo "Testing our results."
find $zip -iname "*.bz2" | xargs bunzip2 -t
```

```
# Not a normal part of the process: we duplicate the ZIP disk onto an
# NFS mount elsewhere.

#  echo "Backing the ZIP drive to the NFS mount."

#  umount $save
#  mount $save

#  rm -r $save/zip
#  mkdir $save/zip
#  cp -pr $zip $save

df -m
```

11.1.5. `restore.metadata`

이 스크립트는 일단계 복구로서 ZIP disk에서 metadata를 복구한다.

```
#! /bin/sh

# A script to restore the meta-data from the ZIP disk. This runs under
# tomsrtbt only after partitions have been rebuilt, file systems made,
# and mounted. It also assumes the ZIP disk has already been
# mounted. Mounting the ZIP disk read only is probably a good idea.

# Time-stamp: <2002-09-10 11:05:43 ccurley restore.metadata>

# Copyright 2000 through the last date of modification Charles Curley.

# This program is free software; you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by the
# Free Software Foundation; either version 2 of the License, or (at your
# option) any later version.

# This program is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
# General Public License for more details.

# You should have received a copy of the GNU General Public License along
# with this program; if not, write to the Free Software Foundation, Inc.,
# 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

# You can also contact the Free Software Foundation at http://www.fsf.org/

# 2002 07 01: Went to bzip2 to compress the archives, for smaller
# results. This is important in a 100MB ZIP disk. Also some general
```

```
# code cleanup.

# For more information contact the author, Charles Curley, at
# http://w3.trib.com/~ccurley/.


zip="/mnt"; # Where we mount the zip drive.
target="/target"; # Where the hard drive to restore is mounted.

ls -lt $zip # Warm fuzzies for the user.

cd $target

# Restore the archived metadata files.
for archive in $( ls $zip/*.bz2 ); do
echo $archive
ls -al $archive
bzip2 -dc $archive | tar -xf -
done

# Build the mount points for our second stage restoration and other
# things.

# If you boot via an initrd, make sure you build a directory here so
# the kernel can mount the initrd.

for dir in mnt mnt/save mnt/zip mnt/cdrom mnt/floppy mnt/imports proc; do
mkdir $target/$dir
done

chmod a-w $target/proc # Restore /proc's read-only permissions

# Restore the scripts we used to create the ZIP disk and the ones we will
# use to restore it. These should be the latest & greatest in case we had
# to do any editing during 1st stage restore.
cp -p $zip/root.bin/* $target/root/bin

# Now install the boot sector.
chroot $target /sbin/lilo -C /etc/lilo.conf

df -m
```

11.1.6. `first.stage`

This script runs the entire first stage restore with no operator intervention.

If you want `make.dev.hda` to check for bad blocks when it puts a file system on the partitions, use a "-c" command line option.

```
#! /bin/sh

# A master script to run the other, detailed scripts. Use this script
# only if you want no human intervention in the restore process. The
# only option is -c, which forces bad block checking during formatting
# of the partitions.

# Time-stamp: <2002-07-25 14:22:03 ccurley first.stage>

# Copyright 2002 through the last date of modification Charles Curley.

# This program is free software; you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by the
# Free Software Foundation; either version 2 of the License, or (at your
# option) any later version.

# This program is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
# General Public License for more details.

# You should have received a copy of the GNU General Public License along
# with this program; if not, write to the Free Software Foundation, Inc.,
# 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

# You can also contact the Free Software Foundation at http://www.fsf.org/

export blockcheck=$1;

if [ "$blockcheck" != "-c" ] && [ -n "$blockcheck" ]
then
    echo "${0}: automated restore with no human interaction."
    echo "${0}: -c: block check during file system making."
    exit 1;
fi

cd /mnt/root.bin # just in case we aren't already where we should be.

for drive in $( ls make.dev.* ); do
    ./$drive $blockcheck;
done

restore.metadata
```

11.2. 2단계(Second Stage)

이 스크립트는 백업하거나 복구할 컴퓨터에서 실행한다.

11.2.1. `back.up.all`

이 스크립트는 NFS 마운트를 이용해서 다른 컴퓨터에 저장한다. tape 드라이브나 다른 매체에 저장하도록 할 수 있다.

```
#! /bin/sh

# Back up the entire system to another computer’s drive. To make this
# work, we need a convenient chunk of disk space on the remote computer we
# can nfs mount as /mnt/save.

# Time-stamp: <2001-11-19 09:19:35 ccurley back.up.all>

# Copyright 2000 through the last date of modification Charles Curley.

# This program is free software; you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by the
# Free Software Foundation; either version 2 of the License, or (at your
# option) any later version.

# This program is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
# General Public License for more details.

# You should have received a copy of the GNU General Public License along
# with this program; if not, write to the Free Software Foundation, Inc.,
# 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

# You can also contact the Free Software Foundation at http://www.fsf.org/

# For more information contact the author, Charles Curley, at
# http://w3.trib.com/~ccurley/.

save="/mnt/save"

# Make sure it’s there
umount $save
mount $save

cd /

rm $save/tester.tar.old.gz
mv $save/tester.tar.gz $save/tester.tar.old.gz

# save everything except /mnt, /proc, and nfs mounted directories.
```

```
time tar cf - / --exclude /mnt --exclude /proc --exclude $save\
    | gzip -c > $save/tester.tar.gz
```

11.2.2. `back.up.all.ssh`

이 스크립트는 `back.up.all`과 동일하게 동작한다. 단지 NFS 대신 ssh을 이용한다는 것이 다를 뿐이다.

```
#! /bin/sh

# Back up the entire system to another computer's drive. To make this
# work, we need a convenient chunk of disk space on the remote
# computer. This version uses ssh to do its transfer, and compresses
# using bz2. This means this script has to know more about the other
# computer, which does not make for good modularization.

# Time-stamp: <2001-11-19 09:19:48 ccurley back.up.all.ssh>

# Copyright 2000 through the last date of modification Charles Curley.

# This program is free software; you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by the
# Free Software Foundation; either version 2 of the License, or (at your
# option) any later version.

# This program is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
# General Public License for more details.

# You should have received a copy of the GNU General Public License along
# with this program; if not, write to the Free Software Foundation, Inc.,
# 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

# You can also contact the Free Software Foundation at http://www.fsf.org/

# For more information contact the author, Charles Curley, at
# http://w3.trib.com/~ccurley/.

save="/backs/tester"
backup_server="charlesc"

# rotate the old backups. Do it all in one line to minimze authentica-
tion overhead.
ssh $backup_server "rm $save/tester.tar.old.bz2; mv $save/tester.tar.bz2 \
    $save/tester.tar.old.bz2"
```

```
# save everything except /mnt, /proc, and squid directories.

time tar cf - / --exclude /mnt --exclude /proc --exclude /var/spool/squid\
    | ssh $backup_server "bzip2 -9 > $save/tester.tar.bz2"
```

11.2.3. `restore.all`

이 스크립트는 `back.up.all` 스크립트를 이용해서 백업한 경우 복구하기 위해 사용하는 것이다.

```
#! /bin/sh

# A script to restore all of the data from an nfs mount. This is our final
# stage restore.

# Time-stamp: <2001-11-19 09:21:22 ccurley restore.all>

# Copyright 2000 through the last date of modification Charles Curley.

# This program is free software; you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by the
# Free Software Foundation; either version 2 of the License, or (at your
# option) any later version.

# This program is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
# General Public License for more details.

# You should have received a copy of the GNU General Public License along
# with this program; if not, write to the Free Software Foundation, Inc.,
# 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

# You can also contact the Free Software Foundation at http://www.fsf.org/

# For more information contact the author, Charles Curley, at
# http://w3.trib.com/~ccurley/.

export save="/mnt/save"

mount $save

cd /
gunzip -dc $save/tester.tar.gz | tar -xpkf -

rm /var/run/*.pid

lilo
```

11.2.4. `restore.all.ssh`

이 스크립트는 `back.up.all.ssh`을 이용하여 백업했을 때 사용하는 복구 스크립트이다.

```
#! /bin/sh

# A script to restore all of the data using ssh and bunzip2. This is
# our final stage restore.

# Copyright 2000 through the last date of modification Charles Curley.

# Time-stamp: <2001-11-19 09:21:33 ccurley restore.all.ssh>

# This program is free software; you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by the
# Free Software Foundation; either version 2 of the License, or (at your
# option) any later version.

# This program is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
# General Public License for more details.

# You should have received a copy of the GNU General Public License along
# with this program; if not, write to the Free Software Foundation, Inc.,
# 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

# You can also contact the Free Software Foundation at http://www.fsf.org/

# For more information contact the author, Charles Curley, at
# http://w3.trib.com/~ccurley/.

save="/backs/tester/"
backup_server="charlesc"

cd /

ssh $backup_server "cat $save/tester.tar.bz2" | bunzip2 | tar -xpkf -

rm /var/run/*.pid

lilo
```

11.3. 백업 서버 스크립트(Backup Server Script)

위의 SSH 스크립트는 보안에 있어 문제가 있을 수 있다. 방화벽 위에서 스크립트들이 실행된다면, 방화벽은 SSH을 통하여 백업 서버에 접근을 허가해야만 한다. 이 경우에 있어, 영리

한 크래커는 백업서버를 크랙할 수도 있을지 모른다. 백업서버에서 백업과 복구 스크립트를 실행하고, 백업 서버로 하여금 방화벽에 접근하도록 하는 것이 좀더 안전한 방법이 될 수 있을 것이다. 이상이 왜 이 스크립트들이 존재하는 가에 대한 이유이다. 이것들을 `get.x`와 `restore.x`로 이름을 바꾸어라. 여기서, `x`는 대상 컴퓨터의 이름이다. 대상 컴퓨터의 호스트 이름을 사용할 수 있도록($target 변수의 초기화) 스크립트들을 편집하거나 명령어 줄에 인수로 사용할 수 있도록 스크립트를 다시 써야 한다.

이 스크립트들은 단지 일단계 백업과 복구만이 아니라, 대상 컴퓨터를 완전히 백업하고 복구한다. 또한 만약 오류가 있는 ZIP disk를 교체할 필요가 있을 경우에, `get.tester` 스크립트는 ZIP disk 또한 백업한다.

나는 이 스크립트들을 주기적으로 사용한다.

11.3.1. `get.tester`

```
#! /bin/sh

# Back up another computer’s drive to this system. To make this work, we
# need a convenient chunk of disk space on this computer. This version
# uses ssh to do its transfer, and compresses using bz2. This version was
# developed so that the system to be backed up won’t be authenticated to
# log onto the backup computer. This script is intended to be used on a
# firewall. You don’t want the firewall to be authenticated to the backup
# system in case the firewall is cracked.

# Time-stamp: <2002-08-23 16:12:35 ccurley get.tester>

# Copyright 2000 through the last date of modification Charles Curley.

# This program is free software; you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by the
# Free Software Foundation; either version 2 of the License, or (at your
# option) any later version.

# This program is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
# General Public License for more details.

# You should have received a copy of the GNU General Public License along
# with this program; if not, write to the Free Software Foundation, Inc.,
# 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

# You can also contact the Free Software Foundation at http://www.fsf.org/

# For more information contact the author, Charles Curley, at
# http://w3.trib.com/~ccurley/.
```

```
# 2002 07 01: We now set the path on the target to the zip drive with
# a variable. This fixes a bug in the command to eject the zip disk.

# 2002 07 01: The zip disk archives are now in bzip2 format, so this
# script has been changed to reflect that.



# The host name of the computer to be backed up.
target=tester
zip=/mnt/zip

echo Backing up $target

echo Aging the ZIP disk backups.

rm -r $target.old.zip

mv $target.zip $target.old.zip

ssh $target "modprobe ppa ; mount -r $zip"

echo Copying the ZIP disk.

# -r for recursive copy, -p to preserve times and permissions, -q for
# quiet: no progress meter.

scp -qpr $target:$zip $target.zip

du -hs $target.zip


echo Aging the archives

rm $target.tar.old.bz2

mv $target.tar.bz2 $target.tar.old.bz2


echo Backing up $target to the backup server.

ssh $target tar -cf - / --exclude /mnt --exclude /proc --
exclude /var/spool/squid\
    | bzip2 -9 | cat > $target.tar.bz2

echo Testing the results.
find . -iname "*.bz2" | xargs bunzip2 -t

ssh $target "eject $zip"
```

11.3.2. `restore.tester`

```
#! /bin/sh

# A script to restore all of the data to tester via ssh. This is our final
# stage restore.

# Time-stamp: <2001-11-19 09:22:01 ccurley restore.tester>

# Copyright 2000 through the last date of modification Charles Curley.

# This program is free software; you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by the
# Free Software Foundation; either version 2 of the License, or (at your
# option) any later version.

# This program is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
# General Public License for more details.

# You should have received a copy of the GNU General Public License along
# with this program; if not, write to the Free Software Foundation, Inc.,
# 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

# You can also contact the Free Software Foundation at http://www.fsf.org/

# For more information contact the author, Charles Curley, at
# http://w3.trib.com/~ccurley/.

# The host name of the computer to be restored.

target=tester

bunzip2 -dc $target.tar.bz2 | ssh $target "cd / ; tar -xpkf - "

ssh $target lilo
```

12. 자료 출처(Resources)

특별한 순서는 없다. 여러분이 혼자서 알아보기를 원한다면 이것들을 보라. 여기에 소개된 목록이 뭔가를 추천한다는 의미 는 아니다. 또한, 사실 많은 것을 사용해 보지 못했으므로 특별히 언급할 것도 많지 않다.

- W. Curtis Preston (http://www.oreilly.com/catalog/unixbr/author.html)의 아주 훌륭한 저작인 *Unix Backup & Recovery* (http://www.oreilly.com/catalog/unixbr/). 이 책에서 Bare Metal Recovery를 시작했다. 이 책을 매우 추천한다. 나의 책에 대한 평가를 읽어 보기 바란다 (http://www2.linuxjournal.com/lj-issues/issue78/3839.html).
-

  작은 리눅스 배포판의 목록. (http://www.fokus.gmd.de/linux/linux-distrib-small.html)
- tomsrtbt (http://www.toms.net/rb), "The most Linux on 1 floppy disk." Tom은 다른 작은 리눅스 배포판에 대한 link를 가지고 있다.
- 리눅스 문서화 프로젝트 (http://www.tldp.org/). 특히 "*LILO, Linux Crash Rescue HOW-TO*"를 주의해서 볼 것.
- The Free Software Foundation's `parted` (http://www.gnu.org/software/parted) for editing (enlarging, shrinking, moving) partitions.
- 파티션을 백업하기 위한Partition Image (http://www.partimage.org/).

  웹페이지에서 보면, "Partition Image는 Linux/Unix 유틸리티로서, 분할영역의 정보를 다양한 포맷(아래를 볼것)으로 저장할 수 있다. 그리고 여러개의 파일로 쪼개어 여러개의 플로피에 저장하게 만들 수 있다.... 분할 영역은 버전 0.6.0 이후로 네트워크를 통해서 저장될 수 있다. "
- Hugo Rabson의 Mondo (http://www.microwerks.net/~hugo/): "파일 시스템의 일부 또는 전부를 담고 있는 하나 또는 그 이상의 부팅가능한 응급 복구 CD(또는 tape+플로피)를 만든다. 재해 발생으로 인한 자료 손실의 경우, 여러분은 Bare Metal로 부터 복구를 해야할 것이다."
- 유망해 보이는 Beeblebrox Project (http://beeblebrox.sourceforge.net/)

A. GNU Free Documentation License

Version 1.1, March 2000

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document’s license notice.

H. Include an unaltered copy of this License.

I. Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document’s Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

11. How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.