Saving Space mini-HOWTO

By Guido Gonzato, *guido@ibogeo.df.unibo.it* <mailto:guido@ibogeo.df.unibo.it> v1.0.1, 7 April 1999

역자 : 이호연 *guru2k@kldp.org* <mailto:guru2k@kldp.org>

이 미니 하우투는 당신에게 당신의 리눅스가 설치된 곳에 이용할 수 있는 공간을 스케줄하는 것을 지시할 것이다. 특히 노트북 사용자를 기준으로 작성 하였다.

목 차

제 1 절 소개

나는 리눅스와 윈도우 95가 설치된 노트북 하나를 가지고 있다. 나는 윈도우 파티션에 500메가를, 리눅스에 240메가를 할당했다. 후자는 작게 보일 것이다. 거기에 레드햇 4.1 기반의 리눅스 시스템을 완벽히 알맞게 설치했고 몇몇 이용할 수 있는 많은 공간이 이상하게도 남았다.

당신이 공간이 필요하다면, 당신이 다음의 부분에서 찾을 것은 하드 디스크의 상당한 양의 공간을 비어있게 할 것이다. 오직 나는 당신에게 주의를 준다: 만약 어떤 것이 잘못되었다면 나에게 책임을 돌리지 마라! 우리는 나를 위해 좋은 작업을 하는 2개의 프로그램을 쓸 것이다. 그러나 본질적으로 위험하다. 당신은 경고를 받았다.

제 2 절 필요한 소프트웨어

필요한 것들:

완전한 리눅스 시스템 작업하기 (어떤 버전이라도 좋다);

sunsite.unc.edu:/pub/Linux/utils/compress <ftp://sunsite.unc.edu:/pub/Linux/utils/compress> 에서 찾을 수 있는 보통의 gzip 압축 유틸리티 혹은 bzip2중 하나; 이 툴은 압축을 gzip보다 잘 한다. 그러나 역시 조금 느리고 많은 메모리를 소모한다;

cdata.tvnet.hu/~ml/upx.html <http://cdata.tvnet.hu/~ml/upx.html> 과 *wildsau.idv.uni-linz.ac.at/mfx/upx.html* <http://wildsau.idv.uni-linz.ac.at/mfx/upx.html> 의 홈페이지에 있는 upx를 실행할 수 있는 압축 프로그램;

sunsite.unc.edu:/pub/Linux/libs/compression <ftp://sunsite.unc.edu:/pub/Linux/libs/compression> 에서 이용할 수 있는 zlibc 패키지. zlibc-X.X.tar.gz라고 불리며 X.X는 최신 버전이다.

다른 실행 할 수 있는 압축 프로그램이 있다. 하나는 gzexe (잊어버렸다), 잠시 upx가 이용할 수 있게 되기 전에 아마 최고였던 tcx; 아직 또 하나의 tzx, 이론 보다 더 나은 tcx. 그 것은 내가 스팍 PC에서 시도 했을 때 모든 것을 고정시켰다. 비록 나는 무엇이 잘못되었다고 확신하지는 않는다. 당신은 upx와 더 잘 고정할 수 있다; 그것은 믿을 수 있고, 쓸모 있고 매우 편리하다.

명쾌한 파일 시스템 압축 a la Stacker이 제공하는 커널 패치가 있다. 그러나 현재 이것은 아직 아무도 안정성과 믿을 수 있는 평가 되지 않았다. 안전의 목적은 더 나아가게 하기 위함이다.

제 3 절 순서

3.1 커널 소스 줄이기

당신은 커널 소스를 20메가 보다 더 작게 줄이거나 그것을 지워도 된다. 만약에 그렇다면, 나는 당신의 컴퓨터에서 한번 그리고 모두를 위해서 맞춰진 새로운 커널을 컴파일하기를 제안한다. 그후 소스를 없애라. 조심 하라.

당신의 컴퓨터가 정확히 설정되었다고 확신하지 않는다면 커널 소스를 지우는 것을 권하지 않는다. 그 밖에, C 프로그램을 컴파일 하기 위해 커널 인클루드 파일이 필요하다. 다시 생각하라!

그렇게 하려고 한다면, 당신의 컴퓨터에서 애플리케이션을 컴파일하기 원한다면 include/linux 디렉토리를 지우면 안된다.

3.2 애플리케이션 줄이기

지금, 당신이 정말로 필요한 소프트웨어를 결정해라. 몇몇은 장황하다고 입증 되었다: 예를 들면, 당신이 emacs 없이 살 수 없다고 확신하는가? 당신은 대신 jed를 쓸수 있다.

당신이 보존하기 원하는 것을 결정하라. 몇몇 전반적인 동작:

gcc는 상당히 큰 패키지이다. 커널과 리눅스 바이너리가 생성되지 않은 모든 애플리케이션을 다시 컴파일 하는데 필요하다. 물론, C나 포트란 (f2c 혹은 g77) 로 프로그램을 만들었다면 그것 또한 필요하다; 당신의 지우기 전에 깊이 생각하라. lcc 같은 다른 컴파일러도 좋다. 그러나 gcc 레벨이 아니다. 나는 그것을 보존하기를 제안한다;

X11은 아주 크다. 그러나 가지고 있기에 좋다. 당신이 그것을 포기하지 못하겠다면, 적어도 작게 만들려고 시도하라: 오직 X 서버만 정당하고, 하나의 간단한 윈도우 매니저, 오직 하나의 xterm, 100 dpi 폰트는 안되고, 그외;

Tex과 첨부된 패지키는 정말 매우 크다. LaTeX과 TeX를 포기하는 것은 많은 공간을 절약한다; 당신이 dvitty와 dvivga를 쓴다면, xdvi나 ghostview같은 X11 프리뷰어를 포기하는 것은 가능하다. X11 dviware가 필요로하는 것을 제거하는 것은 X11를 장황하게 만들 수 있다;

게임은 절대로 "필요" 하지 않다.

3.3 바이너리 줄이기

바이너리의 크기를 감소시키기 시작하자. /usr/bin 과 문제의 명령으로 위치를 바꾸자.

```
machine:/usr/bin# strip *
```

바이너리의 임베디드 기호와 같이 없어질 것이다. /usr/X11R6/bin 과 다른 실행 가능한 디렉토리에서 반복해라 (TeX와 gcc 바이너리를 잊어 버리지 마라). 그러나 /sbin, /bin 혹은 /usr/sbin 에서는 하지 마라!

3.4 Upx로 압축하기

먼저, upx를 설치하고 문서를 읽어라. 그 후 /usr/bin으로 가서 upx * 명령으로 그 것을 실행시켜라; 모든 실행 가능한 것을 압축할 것이다. 다른 디렉토리에서도 이 작업을 반복하라.

새 패키지를 설치할 때 마다 실행 가능한 파일을 압축하는 것을 기억하라!

3.5 Gzip으로 압축하기

한번 압축된 다른 파일이 많이 있다. /usr/doc/ 에서도 시작하자; 디렉토리를 옮기고 명령을 실행하라

```
machine:/usr/doc# find . -type f -exec gzip -9 {} \; 2> /dev/null
```

당신이 새 패키지를 설치할 때 마다 docs를 압축하는 것을 기억하라!

TeX용 문서가 포함된 디렉토리에서도 이 작업을 반복하가(시스템의 /usr/lib/texmf/texmf/doc/. 확실하면 이 디렉토리를 완전히 제거하라.

지금, zlibc를 설치하고 그 것을 컴파일 할 것이다. 당신의 시스템이 나의 것과 비슷하다면, 컴파일 프로세스는 libc(정적)이 없어졌다고 할 것이다. 침착하고, 당신은 당신이 필요한 uncompress.o 라고 불리는 파일을 찾을 것이다. 그 것을 /usr/local/lib로 옮기고 당신의 /etc/profile 파일에 이 라인을 추가하라:

```
export LD_ELF_PRELOAD=/usr/local/lib/uncompress.o
```

지금, 당신은 gzip으로만 압축을 할 수 있다. 그러나 또한 데이타 파일: 그 것을 사용할 수 있음에도 불구하고 애플리케이션은 그것을 쓸 것이다. 이론적으로, 트릭은 대부분의 애플리케이션과 작업을 해야한 하지만 당신의 성공의 단계를 습관적으로 변경할 것이다. 나의 것은 아주 활용 되지 않았다.

제 4 절 진짜 살아있는 예제

이 것은 내가 나의 컴퓨터의 하나의 방법을 적용시킨 것이다. 사용하기 전에, df는 내가 398,798 1024–blocks라고 보고했다:

나는 커널 소스와 커널 해더를 제거 하지 않았다;

나는 몇몇 애플리케이션과 모든 게임을 지웠다. 그러나 나는 X11, X11 개발툴, C와 포트란 개발툴, Tcl/Tk, 네트워킹 툴과 그외 다른 표준 애플리케이션들을 지웠다. df는 244,669 블록을 사용한다고 보고했다;

나는 /usr/bin과 /usr/X11R6/bin과 /usr/lib/texmf/bin/i586-linux와 /usr/lib/gcclib/i386-linux/2.7.2.1에서 upx를 실행시켰다. 226,270 블록을 사용한다고 보고했다;

나는 /usr/doc와 /usr/lib/texmf/tex/mf/dos의 문서를 압축했다: 198,745 블록을 사용한다고 보고했다.

합계를 구하면, 나는 398,798 블록을 198,745 블록으로 만들었다. 재료를 생각해보라. 당신은 200,000 블록을 절약할 수 있다! gzip 대신에 bzip2를 사용했다면 더 많은 공간이 남았을 것이다.

평균적으로 볼때, 시작할 때 조심하고 필요한 애플리케이션을 설치하고, 실행 가능한 파일과 문서들은 20 메가 정도를 절약해 준다. 노트북에서는, 이 것은 생명 구조자가 될 수 있다.

(역자주 : 이하의 부분은 저작권 관련 부분으로 번역을 하지 않도록 하겠습니다.)

제 5 절 The End

5.1 Copyright

5.2 Disclaimer

"Saving Space mini-HOWTO"was written by Guido Gonzato, *REMOVE_MEguido@ibogeo.df.unibo.it* `<mailto:REMOVE_MEguido@ibogeo.df.unibo.it>`.

This document is provided "as is". I put great effort into writing it as accurately as I could, but you use the information contained in it at your own risk. In no event shall I be liable for any damages resulting from the use of this work.

Feedback is welcome. For any requests, suggestions, flames, etc., feel free to contact me.

Enjoy Linux and life,

Guido =8-)