The Linux Tips HOWTO

Paul Anderson, *paul@geeky1.ebtech.net* <mailto:paul@geeky1.ebtech.net> v2.4, 12 October 1996

이 howto는 알아내기 어려운 힌트들과 리눅스를 좀더 나이스하게 만 드는 tweeking을 포함하고 있다.

목 차

제 1 절 소개

리눅스를 좀더 재밌게 만들어주는 매우 좋은 트릭과 최적화의 리스트 인 리눅스 팁 하우투 문서에 오게된 것을 환영한다. 내가 여기에 기록 해 놓은 모든 것은 내 머리 꼭대기에서(?) 나온 팁들과 예전의 팁-하 우투 문서에 있던 것 들이다. 여러분이 좋아하는 힌트나 팁이 있다면 은 다음의 팁-하우투에 넣을

수 있도록 나에게 보내기 바란다. Paul Anderson Maintainer–Linux TIPS HOWTO *panderso@ebtech.net* `<mailto:panderso@ebtech.net>`

제 2 절 간단한 팁들

2.1 Handy Syslog Trick Paul Anderson, Tips-HOWTO maintainer.

당신의 /etc/syslog.conf를 수정하고 다음의 라인을 추가해라:

```
# Dump everything on tty8
*.*                                     /dev/tty8
```

경고 : TAB을 사용해야 한다는 것을 기억해라. syslog는 space를 싫 어한다...

2.2 Handy Script to Clean Up Corefiles.

ohammers@cu-online.com `<mailto:ohammers@cu-online.com>` Handy Script to Clean Up Corefiles. 다음과 같이 rmcores라 불리는 file을 만들어라.(필자는 그것을 handle-cores라 부른다) :

```
#!/bin/sh
USAGE="$0 <directory> <message-file>"
if [ $# != 2 ] ; then
        echo $USAGE
        exit
fi
 echo Deleting...
find $1 -name core -atime 7 -print -exec rm {} \;
echo e-mailing
for name in  `find $1 -name  core -exec  ls -l {}  \; | cut
-c16-24`
do
        echo $name
        cat $2 | mail $name
done
```

그리고 cron job이 그것을 종종 실행하도록 해라.

2.3 Moving directories between filesystems.

Alan Cox,*A.Cox@swansea.ac.uk* `<mailto:A.Cox@swansea.ac.uk>` file 트리 전체를 한 디스크로부터 다른 디스크로 빠르게 옮기는 방법

```
(cd /source/directory &&  tar cf  - . )  | (cd  /dest/directory
&& tar xvfp -)
```

불행이 있을 경우 디렉토리가 뭉게지는 것을 방지하기 위해서 cd /source/directory; tar...etc부터 변경해라. 나에게 알려준것에 대해 Jim Dennis에게, *jadestar@rahul.net* `<mailto:jadestar@rahul.net>`, 감사한다.

2.4 어떤 디렉토리가 가장 큰지 알아내기.

mghazey@miso.lowdown.com <mailto:mghazey@miso.lowdown.com> 당신의 컴퓨터에서 가장 큰 디렉토리는 무엇인지 궁금해 한적은 없는 가? 여기 알아낼 수 있는 방법이 있다.

```
du -S | sort -n
```

2.5 The Linux Gazette

영예는 Linux Gazette의 창시자인 John Fisk에게 돌아간다. 그것은 놀 라운 e-zine plus이며, 그것은 무료이다!! 더 이상 무엇을 물어볼것인 가. 이곳에서 확인하라. *www.redhat.com/lg* <http://www.redhat.com/lg> 그곳에 있는동안 John Fisk에게 e-zine LG가 얼마나 놀라운 것 인지 를 말해주는 노트를 남겨라.

2.6 Pointer to patch for GNU Make 3.70 to change VPATH behavor.

Ted Stern,*stern@amath.washington.edu* <mailto:stern@amath.washington.edu> Pointer to patch for GNU Make 3.70 to change VPATH behavor. 많은 사람들이 이러한 문제점을 지니고 있는지는 알수없지만, 내가 좋아하지 않는 3.70 버전의 GNU make의 특징이 있다. 그것은 절대 경 로를 줄 경우 VPATH가 재미있는 행동을 한다는 것이다. 이것을 수정 하는 절대적으로 믿을만한 패치가 있는데, Paul Smith<*psmith@wellfleet.com* <mailto:psmith@wellfleet.com>>으로부터 그것을 구할 수 있다. 또 그는 앞으로의 모든 새로운 버전의 make의 문서와 패치를 뉴스그룹 |gnu.utils.bug|에 올려놓을 것이다. 일반적으로, 나는 내가 억세스하는 모든 시스템에 이 패치를 적용하고 gmake를 다시 커파일한다.

2.7 어떻게 매 부팅시마다 fscking을 하는 것을 막을것인가.

Dale Lutz, *dal@wimsey.com* <mailto:dal@wimsey.com> Q: 매번 부팅할때마다 e2fsck가 디스크 체크를 하는 것을 어떻게 막 나? A: 커널을 다시만들 때, filesystem은 'dirty'하게 기록되고 그래서 당 신의 디스크는 매 부팅마다 체크될 것이다. 이것을 수정하려면 다음을 실행해라: rdev -R /zImage 1 이것은 더 이상 filesystem이 'dirty'하다고 납득하지 않도록 커널을 수 정해준다. 알아두기 : lilo를 사용한다면 lilo config file 내의 리눅스 셋업에 read-only를 추가해라. (흔히 /etc/lilo.conf)

2.8 부트시에 "device busy"에 의해 야기되는 fscks 피하기.

Jon Tombs, *jon@gtex02.us.es* <mailto:jon@gtex02.us.es> 리부팅할 때 filesystem이 fscks를 필요로 하게 만드는 셧다운시의 device busy 에러를 종종 만난다면, 여기 간단한 해결책이 있다: /etc/rc.d/init.d/halt 혹은 /etc/rc.d/rc.0에 다음의 라인을 추가한다. mount -o remount,ro /mount.dir 루트 '/'를 제외한 모든 마운트 된 filesystem에 있어서 umount -a를 호출하기 전에 이러한 처리를 해준다. 이것은 몇가지 이유에 있어서 shutdown이 모든 프로세스를 죽이고 디스크를 umount 하는데 실패하 더라도 reboot할 때 여전히 'clean'할 것을 의미한다. 나의 경우에는 reboot할 때 상당한 시간을 절약해준다.

2.9 하드디스크 상에서 가장 큰 파일 알아내기

Simon Amor, *simon@foobar.co.uk* <mailto:simon@foobar.co.uk>

```
ls -l | sort +4n
```

또는 하드디스크가 공간상의 문제가 있다면 이것은 시간은 걸리지만 잘 훌륭히 동작할 것이다.

```
    cd /
ls -lR | sort +4n
```

2.10 How to print pages with a margin for hole punching.

Mike Dickey, *mdickey@thorplus.lib.perdue.edu* <mailto:mdickey@thorplus.lib.perdue.edu>

```
#!/bin/sh
# /usr/local/bin/print
# a simple formatted printout, to enable someone to
# 3-hole punch the output and put it in a binder
cat $1 | pr -t -o 5 -w 85 | lpr
```

2.11 file 트리를 통하여 특정한 일상 표현 찾아내기.

Raul Deluth Miller, *rockwell@nova.umd.edu* <mailto:rockwell@nova.umd.edu> file 트리를 통하여 특정한 일상 표현 찾아내기 나는 이것을 'forall' 스크립트라고 부른다. 이렇게 사용해라:

```
forall /usr/include grep -i ioctl
 forall /usr/man grep ioctl
```

여기 forall이 있다:

```
#!/bin/sh
if [ 1 = `expr 2 \> $#` ]
then
        echo Usage: $0 dir cmd [optargs]
        exit 1
fi
dir=$1
shift
find $dir -type f -print | xargs "$@"
```

2.12 자동 세이브나 백업 file을 만드는 프로그램을 사용한뒤의 청소를 위한 스크립트.

Barry Tolnas, *tolnas@nestor.engr.utk.edu* <mailto:tolnas@nestor.engr.utk.edu> 자동 세이브나 백업 file을 만드는 프로그램. 여기 emacs의 자동-세이브(#)와 백업( ) file과 .o file 그리고 TEX.log file을 삭제하면서 반복적으로 디렉토리 계층구조를 감소시키는 간단한 two-liner가 있다. 이것은 또한 .tex와 README file을 압축하기도 한 다. 나는 내 시스템의 이것을 'sqeeze'라 부른다.

```
       #!/bin/sh
       #SQUEEZE removes unnecessary files  and compresses .tex
and README files
```

```
        #By Barry tolnas, tolnas@sun1.engr.utk.edu
        #
        echo squeezing $PWD
        find  $PWD \( -name \*~ -or -name \*.o -or -name \*.log
  -or -name \*\#\) -exec
        rm -f {} \;
        find $PWD \(  -name \*.tex -or  -name \*README\*  -or
  -name \*readme\* \) -exec gzip -9 {} \;
```

2.13 어떤 프로세스가 메모리를 가장 많이 잡아먹고 있는지 알아내기

simon@foobar.co.uk <mailto:simon@foobar.co.uk> 어떤 프로세스가 메모리를 가장 많이 잡아먹고 있는지 알아내기

```
      ps -aux | sort +4n
  -OR-
       ps -aux | sort +5n
```

제 3 절 세부적인 tip

3.1 Linux와 windows에서 swap 파티션 공유하기.

Tony acero, *ace3@midway.uchicagi.edu* <mailto:ace3@midway.uchicagi.edu>

1. 해당 파티션을 도스 파티션으로 포맷하고, 거기에 윈도우 swap file 을 생성한 뒤 아직 윈도우를 실행하지는 말아라. (여러분은 우선은 swap file이 완전히 비어있는 상태로 보존되기를 바랄 것이고, 그러므 로 그것은 잘 압축된다.)

2. Linux로 부팅한뒤에 그 파티션을 file로 저장한다. 예를들어 파티션 이 /dev/hda8이었다면

   ```
   dd if=/dev/hda8 of=/etc/dosswap
   ```

3. 그 도스 swap file을 압축한다. 그것은 가상적으로 모두 0이기 때문 에 매우 잘 압축될 것이다.

   ```
   gzip -9 /etc/dosswap
   ```

4. Linux에 swap 공간을 준비하고 설치하기위해 /etc/rc file에 다음의 라인을 추가한다.

   ```
   xxxxx는 swap 파티션의 block 개수이다.
    mkswap /dev/hda8 XXXXX
    swapon -av
   ```

   /etc/fstab file에 swap 파티션을 위한 엔트리를 추가했음을 확인하라.

5. init/reboot 패키지가 /etc/brc 혹은 /sbin/brc를 지원한다면 다음을 /etc/brc에 추가하고, 그렇지 않은 경우 dos | os/2로 부팅하고 싶을 때 그리고 swap 파티션을 dos/windows 버전으로 바꾸고 싶을 때 이것을 손으로 해주어라.

```
swapoff -av
 zcat /etc/dosswap.gz | dd of=/dev/hda8 bs=1k count=100
```

이것은 오직 처음의 100 블록 만을 파티션에 기록함을 알아두어라. 나는 경험적으로 100 이면 충분함을 알아내었다. >> 이것의 장점과 단점은 무엇인가. 장점 : 상당한 양의 디스크 스페이스를 절약할 수 있다. 단점 : step 5가 자동적으로 이루어지지 않을 경우, 여러분은 그것을 직접 해주어야 함을 기억해야 하며, reboot 프로세스가 조금 느려진다.

3.2 변경되지 않는 flag 사용하기.

Jim Dennis, *jadestar@rahul.net* `<mailto:jadestar@rahul.net>` 변경되지 않는 flag 사용하기 여러분의 시스템을 설치하고 환경을 설정한 바로 뒤에 /bin, /sbin, /usr/bin, and /usr/lib 그리고 다른 usual suspects로 간 뒤 'chattr +i command'를 자유로이 사용하라. 또한 그것을 root의 커널 file에 추가 해라. 이제 'mkdir /etc/.dist/'를 하고 /etc 의 모든 것을 그 디렉토리 로 복사해라. (나는 recursion을 피하기위해 이것을 /tmp/etcdist.tar를 사용하는 두 step을 사용한다.) (옵션으로 당신은 /etc/.dist.tar.gz을 만 들수도 있다)– 그리고 그것을 변경되지 않도록 만든다. 이런일을 하 는 이유는 root로 로그인 했을 때 당할수 있는 피해를 줄이기 위해서 이다. 당신은 stray redirection 오퍼레이션으로 오버라이트를 하지 않 을 것이며, 'rm -rf' 명령을 사용할 때 시스템을 사용불가능하게 하는 일이 없을 것이다. (여러분은 여전히 많은 피해를 입겠지만, libs와 bins는 좀더 안전해질 것이다) 이것은 다양한 보안과 denial of service exploits를 불가능하게 혹은 더욱 어렵게 만들 것이다. (그들의 많은 부분이 독자적인 shell을 제 공하지 않는 SUID 프로그램의 활동을 통해 file을 overwrite 하는데 의존하기 때문이다) 이것의 오직 한가지 불편한 점은 다양한 종류의 시스템 바이너리들을 'make install' 할 때이다. 반면에 그것은 또한 'make install'이 file을 overwrite하지 않도록 해준다. 여러분이 Makefile을 읽는 것을 깜박 잊 고 다시 기록될 file들을 chattr -i 할 때(그리고 당신이 file을 풀어놓 을 디렉토리도), make는 실패하며, chattr 명령을 사용하여 그것을 다 시 실행해야 한다. 당신은 자신의 기존의 bins, libs, 혹은 무엇이던간 에 .old/ 디렉토리로 옮길 기회를 얻게된다. 또는 그것들을 리네임하거 나 tar로 압축따위를 할수도 있다.

3.3 새로운 stuff를 어디에 넣을것인가에 대한 제안

Jim dennis, *jadestar@rahul.net* `<mailto:jadestar@rahul.net>` 새로운 stuff를 어디에 넣을것인가에 대한 제안

모든 새로운 stuff들은 /usr/local 혹은 /usr/local/'hostname'에서 시작 한다. 당신의 배포본이 /usr/local을 비워놓는 것중의 하나라면 /usr/local/src, /usr/local/bin 등을 만들고 그것을 사용해라. 만약 당신 의 배포본이 /usr/local 트리에 뭔가를 집어넣는다면, 당신은 'mkdir /usr/local/'hostname" 을 하고 그것에 'wheel' 그룹 +w를 부여하고 싶 을지 모른다. (나는 wheel 그룹의 모든 멤버들이 그 아래에서 오직 그 들 자신의 file만을 다루도록, 그리고 생성되는 모든 file이 'wheel' 그 룹에 속하도록 SUID와 SGID도 만들었다) 이제 새로운 패키지들을 /usr/local/src/.from/$WHEREVER_I_GOT_IT에 집어넣도록 항상 자 신을 훈련시켜라. 그리고 그들을 /usr/local/src에서(혹은 .../$HOSTNAME에서) build 해라. 그것이 로컬의 heirarchy 아래에 설 치되었음을 확인해라. 만약 그것이 절대적으로 /bin 또는 /sbin 아래에 설치가 되야 한다면, 어딘가에 존재할 때 로컬의 heirarchy로부터 각각 의 엘리먼트로의 심볼릭 링크를 만들어라. 이렇게 하는 이유는 – 그것이 번거로운 일임에도 불구하고 – 만일 의 경우에 백업되고, 리스토어되고 배포본으로부터 완저히 다시설치하 는 것 등을 막는데 도움을 주기 때문이다. /usr/local/.from 디렉토리를 사용함으로써, 소스가 나온곳의 비공식적 log를 유지할수도 있다 –이 것은 새로운 업데이트 사항을 찾을 때 도움을 준다 – . 또 security announcement 리스트를 모니터링 할 때 중요할수도 있다. 집에 있는 나의 시스템중 하나는 내 자신이 이러한 방책을 적용시키기 전에 구성된다. 나는 여전히 그것이 설치된 시스템으로서의 stock과 다른점이 무엇인지를 모른다. 이것은 내 집의 시스템에 거의 설정을 하지 않았다는 점과, 내가 그것을 사용하는 유일한 사람이라는 사실을 무시한 것이다. 반대로 내가 직장에서 설정 해놓은 시스템(내게 그곳의 시스템 관리자로서의 임무가 맡겨졌을 경우)은 모두 이런식으로 설정되어 있고, 많 은 청부인과 다른 MIS 사람들에 의해 관리되어 왔으며, 수많은 업그 레이드와 패키지 설치가 이루어졌다. 그럼에도 나에게는 초기의 설치 와 설정이 끝난뒤에 집어 넣게될 중요한 엘리먼트들에 대한 좋은 아이 디어가 있다.

3.4 시스템관리자를 위한 몇가지 팁.

Jim Dennis, *jadestar@rahul.net* <mailto:jadestar@rahul.net> 시스템관리자를 위한 몇가지 팁.

/README.'hostname'과 /etc/README.'hostname'을 만들고 관리해 라. 혹은 /usr/local/etc/README.'hostname'도 가능하다. 절대적으로, 시스템 관리를 시작한 첫날로부터 온라인 로그 file내에 기록을 해라. root의 /bash_logout에 | vi /README.$(hostname) | 과 같은 라인을 적어도 좋다. 이것을 하는 또 다른 방법은 su나 sudo에 다음과 같은 일을 하는 스크립트를 기록하는 것이다.

```
function exit \
        { unset exit; exit; \
          cat ~/tmp/session.$(date +%y%m%d) \
          >> /README.$(hostname) && \
          vi /README.$(hostname)
          }
script -a ~/tmp/session.$(date +%y%m%d)
/bin/su.org -
```

(session log를 만들기 위해 typescript를 사용하고, 자동으로 log를 appending과 updating을 할 함수를 만든다). 나는 내가 이 방식의 자동화 도구를 만들지 않았음을 인정할 것이다 – 나는 단지 여지껏 스스로의 훈련에 의존해 왔을 뿐이다. 그러나 나 는 장난삼아 그런 아이디어를 생각해 왔었다. (심지어는 이미 본 스크립트의 프로토타입이나 shell 기능의 문제에서 조차도.) 이것으로 나 를 다시 붙드는 것은 'script' 명령 그 자체이다. 나는 이것을 사용하기 전에 소스를 입수하고 몇가지 커맨드 라인 파라메터를 추가해야겠다고 생각한다(커맨드 라인으로부터 스크립트를 잠시중단하거나 정지시키기 위해서). 내가 마지막으로 제안하는 것은 : root의 'PATH=/bin'으로 구성되야 한다는 점이다. 바로 그렇다. 다른 무엇도 root의 path에 이어서는 안 된다. root가 하는 모든일은 /bin으로부터 혹은 alias나 shell 함수에 의 해서 심볼릭 링크를 받던가 또는 혹은 /bin 디렉토리내의 script나 binary이던가 또는 뚜렷한 패스로 타입되는 것이다. 이것은 누군가가 binary를 얼마나 신임하는지를 인식하고 root로서 실행하도록 만들어 준다. 멀티 유저 호스트의 현명한 관리자는 특정 패턴이나 loophole을 찾아내기 위해 자신의 /bin과 /*history를 정기적으로 살펴볼 것이다. 실제로 동기가 부여된 어드민은 자동화 될 수있는 sequence를 알아채 고, 온전한 체크가 삽입될 수 있는곳에 위치시키며, root의 특권을 일 시적으로 삼가기 위한 작업들을 한다(에디터를 설치하고, 정교한 스크 립트 구조를 가진 MTA의 또는 다른곳의 커다란 인터액티브 프로그 램을 설치한다. (그들을 불명예스러운 vi ./.exrc 그리고 emacs ./.emacs 그리고 교활한 $EXINIT 그리고 임베딩된 header/footer 매크 로처럼 transparent나 data file내에 임베딩될 수도 있다. 당연히 다음 과 같은 종류의 커맨드가 실행될 수 있다:

```
cp $data $some_users_home/tmp
su -c $origcommand $whatever_switches
cp $some_users_home/tmp $data
```

대개 마지막의 예방조치는 홈 또는 싱글 유저 워크스테이션 사용자에게 해당하는 것이다 – 그러나 그것은 멀티유저 시스템, 특히 대중에게 노출되어 있는 시스템의 관리자에게 도 유용한 것이다.

3.5 xdm의 host 선택자를 어떻게 설정할것인가.

Arrigo Triiulzi, *a.triulzi@ic.ac.uk* <mailto:a.triulzi@ic.ac.uk>

1. 대개 etc/rc/rc.6 혹은 /etc/rc.local에 xdm을 설치하는 file을 xdm startup 라인을 포함하 도록 다음과 같은 라인을 추가하라.

```
/usr/bin/X11/xdm
exec /usr/bin/X11/X -indirect hostname
```

2. /usr/lib/X11/xdm/Xserver를 수정하여 예를들어 starting 0와 같이 로컬 머신상에서 서버 를 시작하게하는 다음과 같은 라인을 언급하라.

3. 머신을 재부팅하라. 그러면 you are home and away.(무슨뜻인지..^^;) 내가 그것을 나의 서브넷에 맞게 설정하기 위해 필사적으로 노력하던 때, 모든 문제를 조사하기 위해서 거의 일주일이나 걸렸기 때문에 나 는 이것을 추가하였다.

경고 : 예전의 SLS(1.1.1)을 가지고서 몇가지 이유 때문에 xdm 라인 뒤에 -nodaemon을 남 겨놓을 수도 있다. 이것은 이후의 릴리즈에서는 동작하지 않는다.