The Linux Tips HOWTO

Paul Anderson, paul@geeky1.ebtech.net. Svensk översättning: Linus Åkerlund, uxm165t@tninet.se.
v3.1, 26 December 1997. Svensk översättning: 24 juni 1998.

Denna HOWTO innehåller sådana där tips och trick som är svåra att hitta och som gör Linux en aning trevligare.

Innehåll

1 Inledning

Vällkommen till **Linux Tips HOWTO**, en lista av trevliga små trick och optimeringar som gör Linux roligare. Allt jag har här just nu är tips som jag tagit direkt ur eget huvud och tips från den gamla Tips-HOWTOn (varför ta bort ett bra tips?). Så skicka alla dina favorit-trick och -tips till mig så att jag kan ta med dem i nästa Tips-HOWTO.

Paul Anderson *Ansvarig för Linux TIPS HOWTO*

`panderso@ebtech.net`

1.1 Översättarens anmärkningar

Precis som vanligt så kan det finnas tveksamma översättningar och sådant i det här dokumentet. Några kanske vill att jag ska översätta t.ex. variabler i skalprogram till svenska, vilket jag inte gjort. Alla som har synpunkter är välkomna att höra av sig till mig på `uxm165t@tninet.se`.

2 Korta tips

2.1 Praktiskt syslog-trick *Paul Anderson, ansvarig för Tips-HOWTOn*

Ändra i din /etc/syslog.conf och lägg in följande rad:

```
# Skicka allt till tty8
*.*                                     /dev/tty8
```

En sak att tänka på: *KOM IHÅG ATT ANVÄNDA TABBAR!* syslog gillar inte mellanslag...

2.2 Skalprogram för att läsa de där komprimerade HOWTOna *Didier Juges,* `dj@destin.nfds.net`

Från en nybörjare till en annan kommer här ett kort skalprogram som gör det enklare att läsa HOWTO-dokument. Mina HOWTOn finns i /usr/doc/faq/howto/ och är gzippade. Filnamnen är XXX-HOWTO.gz, där XXX är ämnet. Jag skapade följande skalprogram, vilket heter ”howto”, i katalogen /usr/local/sbin:

```
#!/bin/sh
if [ "$1" = "" ]; then
    ls /usr/doc/faq/howto | less
else
    gunzip -c /usr/doc/faq/howto/$1-HOWTO.gz | less
fi
```

När det anropas utan argument så visar det en lista över de tillgängliga HOWTOna. När det sedan anropas med första delen av ett filnamn (innan bindestrecket) som argument så zippar det upp (och behåller originalet intakt) och visar dokumentet.

För att t.ex. läsa Serial-HOWTO.gz, skriv:

$ howto Serial

2.3 Finns det tillräckligt med ledigt minne? *Hans Zoebelein,* `zocki@goldfish.cube.net`

Här kommer ett kort skalprogram, vilket då och då kollar om det finns tillräckligt med ledigt utrymme tillgängligt på vad som helst som dyker upp i mount (hårddiskar, cdrom, disketter osv.).

Om utrymmet tar slut så visas ett meddelande var Xte sekunde på skärmen och ett e-brev skickas för varje full enhet.

```
#!/bin/sh

#
# $Id: check_hdspace,v 1.18 1996/12/11 22:33:29 root Exp root $
#

#
# När jag gick mystiska felmeddelanden under en kompilering och
# tmp-filer fyllde upp mina diskar, skrev jag detta för att få en
# varning innan diskarna blir fulla.
#
# Om det här räddar dina servrar från att explodera, skicka ett
# e-brev till zocki@goldfish.cube.net och prisa mig.
# Om din brinner upp på grund av det så är jag ledsen, men jag
# har varnat dig: inga kompensationer.
# Om du vet hur man hanterar sed så kan du förlåta mig :)

#
# Stoppa in det och glöm det: Stoppa 'check_hdspace &' i rc.local.
# Kollar efter ledigt utrymme på enheter varje $SLEEPTIME sekunder.
# Du kan till och med kolla disketter och bandenheter. :)
# Om det finns mindre än $MINFREE (kb) utrymme, så echo-as en varning
# och ett e-brev skickas för varje aktuell enhet till $MAIL_TO_ME.
# Om det återigen finns mer fritt utrymme än vad som är satt som gräns
# så skickas ett nytt e-brev.
#

# ATT GÖRA: Olika $MINFREE för olika enheter.
# Rensa ut gammalt skräp ur /*tmp-kataloger om utrymmet tar slut.

DEVICES='/dev/sda2 /dev/sda8 /dev/sda9'        # enhet; lägg in diskar här
```

```
MINFREE=20480                                   # kb; varning under detta
SLEEPTIME=10                                    # sek; sov mellan kollarna
MAIL_TO_ME='root@localhost'                     # dåre; vem som får mail


# ------- inga ändringar behövs under denna rad (förhoppningsvis :) -------

MINMB=0
ISFREE=0
MAILED=""
let MINMB=$MINFREE/1024         # jepp, vi är strikta :)

while [ 1 ]; do
        DF="`/bin/df`"
                for DEVICE in $DEVICES ; do
                ISFREE=`echo $DF | sed s#.\*$DEVICE" "\*[0-9]\*""\*[0-9]\*" "\*## | sed s#" ".\*##`

                if [ $ISFREE -le $MINFREE ] ; then
                        let ISMB=$ISFREE/1024
                        echo  "VARNING: $DEVICE endast $ISMB mb ledigt." >&2
                        #echo "mera prylar här" >&2
                        echo -e "\a\a\a\a"

                        if [ -z  "`echo $MAILED | grep -w $DEVICE`" ] ; then
                                echo "VARNING: $DEVICE endast $ISMB mb ledigt.      (Gränsen satt till $MINMB mb)" \
                                | mail -s "VARNING: $DEVICE endast $ISMB mb ledigt!" $MAIL_TO_ME
                                MAILEDH="$MAILED $DEVICE"
                                MAILED=$MAILEDH
                                # lägg in annat, som städning av */tmp-
                                # kataloger, här
                        fi
                        elif [ -n  "`echo $MAILED | grep -w $DEVICE`" ] ; then
                                # Bort med mailed-markeraren om det finns
                                # tillräckligt med utrymme igen, så att vi
                                # kan skicka nya e-brev.
                                MAILEDH="`echo $MAILED  | sed s#$DEVICE##`"
                                MAILED=$MAILEDH
                        fi

                done
                sleep $SLEEPTIME

done
```

2.4 Verktyg för att rensa upp i dina loggfiler *Paul Anderson, ansvarig för Tips-HOWTOn*

Om du är so mjag så har du en lista med 250 prenumeranter, plus mer än 100 meddelanden per dag som kommer med UUCP. Tja, vad ska en hacker göra med sina enorma loggfiler? Installera chklogs. Chklogs är skrivet av Emilio Grimaldo, `grimaldo@panama.iaehv.ln`, och den senaste versionen (1.8) är tillgänglig via ftp från frp.iaehv.ln:/pub/users/grimaldo/chklogs-1.8.tar.gz. Det förklarar ganska bra självt hur du ska installera det (du ska naturligtvis även ta en titt på informationen i doc-katalogen). När du har isntallerat det ska du lägga till följande i crontab:

```
# Kör chklogs klockan 9 varje dag.
00 21 * * *       /usr/local/sbin/chklogs -m
```

När du ändå håller på, glöm inte att tala om för programmeraren vilket trevligt program det är :).

2.5 Praktiskt skalprogram för att städa upp core-filer *Otto Hammersmith,* `ohammers@cu-online.com`

Skapa en fil som heter rmcores (författaren kallar det handle-cores) med följande innehåll:

```
#!/bin/sh
USAGE="$0 <directory> <message-file>"

if [ $# != 2 ] ; then
        echo $USAGE
        exit
fi

 echo Tar bort...
find $1 -name core -atime 7 -print -exec rm {} \;

echo skickar e-brev
for name in `find $1 -name core -exec ls -l {} \; | cut -c16-24`
do
        echo $name
        cat $2 | mail $name
done
```

Sedan gör du så att ett cron-jobb kör det då och då.

2.6 Flytta kataloger mellan filsystem. *Alan Cox,* `A.Cox@swansea.ac.uk`

Ett snabbt sätt att flytta ett helt filträd från en disk till en annan är

```
(cd /käll/katalog && tar cf - . ) | (cd /dest/katalog && tar xvfp -)
```

[Byt från cd /käll/katalog; tar...osv. för att förhindra möjligheten att katalogen förstörs om en olycka inträffar. Tack till Jim Dennis, jadestar@rahul.net, för att han gjorde mig uppmärksam på detta. -Ansvarige.]

2.7 Ta reda på vilka kataloger som är störst. *Mick Ghazey,* `mghazey@miso.lowdown.com`

Har du någonsin funderat på vilka kataloger som är de största på din dator? Så här tar du reda på det:

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

2.8 Linux Gazette

Kudos till John Fisk som grundade Linux Gazette. Det är en suverän e-tidning och den är **GRATIS!!!** Vad mer kan du önska dig? Ta en titt på:

```
http://www.ssc.com/lg
```

Förresten så visar det sig att (1) LG utges nu varje månad och (2) John Fisk har inte längre hand om det, utan människorna på SSC.

2.9 Hänvisning till en patch till GNU Make 3.70 för att ändra VPATH-beteendet. *Ted Stern,* `stern@amath.washington.edu`

Jag vet inte om många har detta problem, men det finns en egenskap hos GNU Make version 3.70 som jag inte tycker om. Det är att VPATH beter sig udnerligt om du anger en absolut sökväg. Det finns en extremt stabil patch för att fixa detta, vilken du kan få från Paul D. Smith <`psmith@wellfleet.com`>. Han publicerar dokumentationen och patchen i nyhetsgruppen gnu.utils.bug efter varje uppdatering av GNU Make. Själv lägger jag till den här patchen och kompilerar om gmake på varje system jag har tillgång.

2.10 Hur får jag mitt system att inte köra fsck efter varje bootning? *Dale Lutz,* `dal@wimsey.com`

F: Hur får jag e2fsck att inte kolla min hårddisk varenda gång jag bootar om?

S: När du kompilerar om kärnan så markeras filsystemet som ”smutsigt” (dirty) och din hårddisk kommer då kollas efter varje ombootning. För att fixa detta kör du:

rdev -R /zImage 1

Detta fixar kärnan, så att den inte längre är övertygad om att filsystemet är smutsigt.

Obs: Om du använder lilo, lägg till `read-only` *till Linux-avdelningen i din konfigureringsfil för lilo (vanligtvis /etc/lilo.conf).*

2.11 Hur man undviker fsck orsakade av ”device busy” då man bootar om. *Jon Tombs,* `jon@gtex02.us.es`

Om du ofta får ”device busy”-fel som gör att filsystemet måste gå igenom en fsck, då du kör shutdown, så kommer här en enkel lösning:

Lägg till raden

```
mount -o remount,ro /mount.dir
```

till `/etc/rc.d/init.d/halt` eller `/etc/rc.d/rc.0`.

Gör detta för varje monterat filsystem utom /, innan anropet till umount -a. Detta betyder att om shutdown av någon anledning misslyckas med att döda alla processer och avmontera alla diskar så kommer de fortfarande att vara rena då du bootar om. Detta spar en massa tid för mig då jag bootar om.

2.12 Hur du hittar de största filerna på din hårddisk. *Simon Amor,* `simon@foobar.co.uk`

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

Om du har väldigt dåligt med utrymme så funkar även det följande, men kan ta en hel del tid:

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

2.13 Hur du skriver ut sidor med marginal för hålslag. ***Mike Dickey,*** mdickey@thorplus.lib.purdue.edu

```
#!/bin/sh
# /usr/local/bin/print
# en enkel formatterad utskrift, för att att låta någon
# ta hål på utskriften och sätta den i en pärm

cat $1 | pr -t -o 5 -w 85 | lpr
```

2.14 Ett sätt att söka igenom filträd efter ett visst reguljärt mönster. ***Raul Deluth Miller,*** rockwell@nova.umd.edu

Jag kallar detta skalprogram "forall". Använd det så här:

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

Här kommer forall:

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

2.15 Ett skalprogram för att städa upp efter program som skapar autosave- och backup-filer. ***Barry Tolnas,*** tolnas@nestor.engr.utk.edu

Här kommer ett enkelt skalprogram på två rader som rekursivt går ned genom kataloghierarkin och tar bort emacs autosave-filer (#) och backup-filer ( ), .o-filer och .log-filer som TeX skapar. Det komprimerar även .tex- och README-filer. Jag kallar det att "pressa ihop" mitt system.

```
#!/bin/sh
#SQUEEZE tar bort onödigt filer och komprimerar .tex- och README-filer
#Av Barry tolnas, tolnas@sun1.engr.utk.edu
#
echo pressar ihop $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 {} \;
```

2.16 Hur du tar reda på vilken process som äter mest minne. ***Simon Amor,*** simon@foobar.co.uk

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

-ELLER-

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

2.17 Ställa in vi för C-programmering *Paul Anderson,* `Ansvarig för Tips-HOWTOn`

Jag programmerar en massa C på min fritid och har tagit mig tid att ställa in vi så att det passar bra för C. Här är min .exrc:

```
set autoindent
set shiftwidth=4
set backspace=2
set ruler
```

Hur fungerar det? autoindent gör att vi automatiskt indenterar varje rad som följer en indenterad rad, shiftwidth ställer in tabblängden till 4 mellanslag, backspace slår på backspaceläget och ruler gör att det visar radnummer. Kom ihåg att du kan gå till en specifik rad, säg 20, genom att skriva:

```
vi +20 myfile.c
```

2.18 Varför hänger sig sendmail i fem minuter då man startar RedHat? *Paul Anderson,* `paul@geeky1.ebtech.net`

Det här är ett ganska vanligt problem, nästan så att det är en FAQ (ofta ställd fråga. övers.anm.). Jag vet inte om RedHat har fixat det här problemet i sin distribution, men annars kan du själv reparera det. Om du kollar i din /etc/hosts-fil så kommer du hitta något i stil med:

```
127.0.0.1               localhost       yourbox
```

När sendmail startas så kollar det din maskins namn (hostname) (i detta exempel är det yourbox). Det ser då att IP-numret för din yourbox är 127.0.0.1, vilket sendmail inte tycker om och gör kollen igen. Det fortsätter så med detta ett tag, tills det ger upp och avslutas. Att lösa detta är extremt enkelt, lägg bara in något i stil med

```
127.0.0.1               localhost
10.56.142.1             yourbox
```

i din /etc/hosts-fil.

2.19 Hur ställer jag in RedHat så att det använder färg-ls? *Paul Anderson,* `paul@geeky1.ebtech.net`

RedHats distribution kommer med färg-ls och varför de inte ställer in det för att använda det som standard är bortom min fattningsförmåga. Här kommer en lösning.

Skriv först eval ‘DIRCOLORS‘

Skriv sedan alias ls=’ls –color=yes’

Lägg in ’alias....’ i din /etc/bashrc

2.20 Hur tar jag reda på vilket bibliotek i /usr/lib som innehåller en viss funktion? *Pawel Veselow,* `vps@unicorn.niimm.spb.su`

Vad händer om du kompilerar och du har missat något bibliotek som måste länkas in? Allt gcc rapporterar är funktionsnamn... Här kommer ett enkelt kommando som gör att du kan hitta det du letar efter:

```
for i in *; do echo $i:;nm $i|grep tgetnum 2>/dev/null;done
```

tgetnum är namnet på den funktion du letar efter.

3 Detaljerade tips

3.1 Dela swap-partition mellan Linux och Windows. *Tony Acero,* ace3@midway.uchicago.edu

1. Formattera partitionen som en dos-partition och skapa swapfilen för Windows på den, men kör inte Windows än. (Du ska hålla swapfilen helt tom för tillfället så att den komprimeras bra.)

2. Boota Linux och sparar partitionen till en fil. Om partitionen var t.ex. /dev/hda8:

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

3. Komprimera filen dosswap; eftersom den i stort sett bara innehåller nollor så kommer den andra komprimeras bra.

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

4. Lägg till det följande till /etc/rc-filen för att förbereda och installera swaputrymmet under Linux: *XXXXX är antalet block i swappartitionen*

   ```
   mkswap /dev/hda8 XXXXX
   swapon -av
   ```

   Se till så att du lägger till en rad för swappartitionen i /etc/fstab.

5. Om ditt init/reboot-paket stödjer /etc/brc eller /sbin/brc, så lägg till det följande i /etc/brc. Annars kan du lägga till det för hand när du vill boota dos|os/2 och vill konvertera tillbaks swappartitionen till dos/windows-versionen:

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

Observera att detta bara skriver tillbaks de första 100 blocke till # partitionen. Enligt min erfarenhet räcker det.

> > Vilka är för- och nackdelarna med att göra detta?

Fördelar: du spar en hel del hårddiskutrymme.

Nackdelar: om steg 5 inte sker automatiskt så måste du komma ihåg att göra det för hand och dessutom gör det ombootningsprocessen en nanosekunde långsammare. :-)

3.2 Desperat undelete. *Michael Hamilton,* michael@actrix.gen.nz

Här kommer ett trick som jag varit tvungen att använda ett par gånger.

Desperata människors textfils-undelete.

Om du av misstag raderar en textfil, t.ex. e-post eller resultatet av en sen natts programmeringssession, så behöver inte allt vara förlorat. Om filen någonsin sparades på disk, alltså om den existerade i åtminstone 30 skunder, så kan dess innehåll fortfarande finnas kvar på partitionen.

Du kan använda kommandot grep för att leta igenom den råa partitionen efter innehållet i filen.

Nyligen tog jag t.ex. bort ett e-brev. Jag avslutade omedelbart all aktivitet som kunde modifiera den partitionen: i detta fall lät jag bara bli att spara filer och kompilera osv. I andra fall har jag faktiskt gått igenom de besvär det innebär att ta ner systemet till enanvändarläge (single user mode) och avmonterat filsystemet.

Sen använde jag kommandot egrep på partitionen: i mitt fall fanns e-brevet i /usr/local/home/michael/, så från utdatan från df kunde jag sluta mig till att det fanns på /dev/hdb5.

```
sputnik3:~ % df
  Filesystem         1024-blocks  Used Available Capacity Mounted on
  /dev/hda3              18621    9759     7901     55%   /
  /dev/hdb3             308852  258443    34458     88%   /usr
  /dev/hdb5             466896  407062    35720     92%   /usr/local

  sputnik3:~ % su
  Password:
  [michael@sputnik3 michael]# egrep -50 'ftp.+COL' /dev/hdb5 > /tmp/x
```

Jag är extremt försiktig när det gäller att fipplat med partitioner, så jag tog en paus för att se till att jag förstod kommandots syntax INNAN jag tryckte return. I det här fallet innehöll e-brevet ordet 'ftp', följt av en del text som följdes av ordet 'COL'. Meddelandet var runt 20 rader långt, så jag använde -50 för att få med alla rader runt den frasen. Förut har jag använt -3000 för att se till att alla rader i någon källkodsfil kom med. Jag omdirigerade utdatan från egrep till en annan partition - detta förhindrade att jag skrev över meddelandet jag letade efter.

Sedan använde jag strings för att hjälpa mig med att inspektera utdatan:

```
strings /tmp/x | less
```

Där fanns e-brevet!

Den här metoden är inte att lita på, allt eller en del av diskutrymmet kan redan ha blivit återanvänt.

Detta trick är antagligen bara användbart på ett enanvändarsystem. På system med flera användare och med mycket diskaktivitet så kan utrymmet du har frigjort mycket väl redan ha återanvänts. Och de flesta av oss kan inte bara ta ned maskinen för våra användare när vi måste hitta en fil vi tagit bort.

På det system jag kör hemma har detta trick varit användbart vid ungefär tre gånger under de senaste åren - vanligtvis när jag förstör en del av dagens arbetsresultat. Om det jag gjort klarar sig så långt att jag tycker att jag gjort anmärkningsvärda framsteg så brukar jag säkerhetskopiera det på disketter, så jag behöver inte använda detta trick så ofta.

3.3 Hur du använder immutable-flaggan. *Jim Dennis,* `jadestar@rahul.net`

Använda immutable-flaggan

Så fort du har installerat och konfigurerat ditt system så bör du gå igenom /bin, /sbin, /usr/bin, /usr/sbin och /usr/lib (och några av de andra katalogerna) och på ett sparsamt sätt använda 'chattr +i kommando'. Lägg också till det till kärnfilerna i rotkatalogen. Kör nu 'mkdir /etc/.dist' och kopiera allt från /etc/ dit (själv gör jag detta i två steg, med /tmp/etcdist.tar för att undvika rekursion). (Du kan även bara skapa /etc/.dist.tar.gz - och göra den immutable.)

Anledningen till allt detta är att beränsa skadan du kan göra när du är inloggad som root. Du kan inte skriva över filer av misstag och du kan inte göra systemet oanvändbart med ett mellanslag för mycket om du använder 'rm -rf' (du kan fortfarande förstöra en massa data - men dina bibliotek och binärfiler är säkrare).

Detta gör också en hel del intrång och "denial of service"-angrepp antingen omöjliga eller svårare (eftersom många av dem är beroende av att kunna skriva över någon fil genom ingrepp av något SUIDat program som *inte tillhandahåller ett visst skalkommando*).

Det enda otrevliga med detta är att när du kompilerar och kör 'make install' på olika typer av systembinärfiler. Å andra sidan så hindrar det också 'make install' från att skriva över filerna. När du glömmer att läsa Makefilen och chattr -i filen som ska skrivas över (och katalogerna till vilka du ska lägga till filer) så misslyckas make-kommandot, men då är det bara att använda kommandot chattr och köra det igen. Du kan också ta tillfället i akt och flytta dina gamla binärfiler, bibliotek och vad som helst till en .old/-katalog, eller byta namn på dem eller tar-a dem eller vad som helst.

3.4 Ett förslag om vad du ska stoppa in nya grejer. *Jim Dennis,* `jadestar@rahul.net`

Alla nya saker ska in i /usr/local eller /usr/local/'hostname'!

Om din distribution är en som lämnar /usr/local tomt, så skapa bara din /usr/local/src, /usr/local/bin osv. och använd dem. Om din distribution stoppar saker i /usr/local-trädet så är det kanske bäst att köra 'mkdir /usr/local/'hostname'' och ge 'wheel'-gruppen +w till den (jag gör den också SUID och SGID för att se till att varje medlem i wheel-gruppen bara kan greja med sina egna filer där och att alla filer som skapas tillhör wheel-gruppen).

Se nu till att vara disciplinerad och *ALLTID! ALLTID! ALLTID!* stoppa nya paket i /usr/local/src/.från/$VAR_JAG_FICK_DET_IFRÅN/ (för .tar eller vilken fil det nu var) och kompilera dem under /usr/local/src (eller .../$HOSTNAME/src). Se till att det installeras under local-trädet. Om det *absolut måste* installeras i /bin eller /usr/bin eller någon annanstans, så kan du skapa en länk från local-trädet till varje element som finns någon annanstans.

Anledningen till detta, även om det leder till mer jobb, är att det hjälper till att isolera det som ska säkerhetskopieras och återställas från distributions-mediet (nu för tiden är det oftast en CD). Genom att använda en /usr/local/.from-katalog upprätthåller du också en informell logg om varifrån källkoden kommit, vilket hjälper dig då letar efter nya versioner, och hjälper dig då du läser igenom säkerhetsmeddelandena i announcement-listor.

Ett av de system jag har hemma (det jag skriver detta på) sattes upp innan jag själv började arbeta enligt dessa principer. Jag "vet" fortfarande inte precis hur det skiljer sig från ett system från en "ren installering". Detta trots att jag har gjort väldigt lite med det systemet, inte konfigurerat det speciellt mycket och är den enda som använder det.

I kontrast till detta så har de system som jag satt upp på jobbet (sedan jag kastats in i rollen som systemadministratör) blivit konfigurerade på detta sätt - de har administrerats av många människor och MIS-typer och har gått igenom många stor uppgraderingar och paketinstalleringar. Trots detta har jag en klar uppfattning om vilka exakta element som lags in *efter* den ursprungliga installeringen och konfigureringen.

3.5 Konvertera alla filer i en katalog till små bokstäver. *Justin Dossey,* `dossey@ou.edu`

Jag la märke till några överdrivet svåra eller onödiga procedurer som rekommenderades i avdelningen med 2c-tips i nummer 12. Eftersom det är mer än ett, så skickar jag dem till dig:

```
#!/bin/sh
         # lowerit
         # konvertera alla filnamn i den aktuella katalogen till små
         # bokstäver. påverkar bara vanliga filer - ändrar inte namnen
```

```
# kataloger. frågar om lov innan det skriver över en fil.
for x in `ls`
  do
  if [ ! -f $x ]; then
    continue
    fi
  lc=`echo $x  | tr '[A-Z]' '[a-z]'`
  if [ $lc != $x ]; then
    mv -i $x $lc
  fi
  done
```

Oj. Det var ett långt skalprogram. Jag skulle inte skriva ett skalprogram för att göra det. Istället skulle jag använda detta kommando:

```
for i in * ; do [ -f $i ] && mv -i $i `echo $i | tr '[A-Z]' '[a-z]'`;
done;
```

direkt från kommandoraden.

Bidragsgivaren säger att han skrev skalprogrammet såsom han gjorde det för att det skulle bli lättare att förstå (se nedan).

I nästa tips, som handlar om att lägga till och ta bort användare, så gör Geoff allt rätt ända tills han kommer till sista steget. Boota om? Gosse, jag hoppas att han inte bootar om varenda gång han tar bort en användare. Allt du behöver göra är att utföra de två första stegen. Vilket slags processer kan den användaren ha igång, förresten? En irc-bot? Döda processerna med ett enkelt

```
kill -9 `ps -aux |grep ^<username> |tr -s " " |cut -d " " -f2`
```

I följande exempel är användarnamnet foo:

```
kill -9 `ps -aux |grep ^foo |tr -s " " |cut -d " " -f2`
```

När vi klarat av det, låt oss gå vidare till det bortglömda lösenordet för root.

Lösningen som gavs i Gazetten är den mest använda, men inte den enklaste. Till både LILO och loadlin kan man skicka med parametern ”single” för att boota direkt till standardskalet utan behöva skriva användarnamn eller lösenord. Därifrån kan man ändra eller ta bort vilket lösenord som helst, innan man skriver ”init 3” för att köra igång fleranvändarläget.

Antal ombootningar: 1 Det andra sättet Antal ombootningar: 2

Justin Dossey

3.6 Några tips för nya systemadministratörer *Jim Dennis,* `jadestar@rahul.net`

Skapa och upprätthåll en /README.‘hostname‘ och/eller en /etc/README.‘hostname‘. *(Eller kanske /usr/local/etc/README.‘hostname‘ - Ansvarige.)*

Från och med *dag ett* ska du absolut föra anteckningar i en loggfil. Du kan lägga in en rad med ”vi /README.$dollar;(hostname)” i roots /bash_logout. Ett annat sätt är att skriva ett su- eller sudo-skalprogram som gör något i stil med följande:

```
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 -
```

(använd typescript-kommandot för att skapa en sessionlogg och skapa en funktion för att automatisera tilläggen och uppdateringarna av loggen.)

Jag ska erkänna att jag inte har lagt in denna automatisering av princip - jag har hittills bara litat på självdisciplin. Jag har dock lekt med idén (till och med gått så långt som till att göra prototyper för skalprogrammen och skalfunktionerna som du ser dem i det här dokumentet). En sak som hindrar mig från att genomföra detta är script-kommandot självt. Jag tror jag kommer bli tvungen att plocka hem källkoden och lägga till några kommandoradsparametrar (för att pause/stoppa skalprogrammets "inspelning" av kommandoraden) innan jag börjar använda det.

Mitt sista förslag (för denna gång):

Roots sökväg ska bestå av 'PATH= /bin'.

Det är allt. Inget annat ska finnas i roots sökväg. Allt root gör ska finnas som en länk från /bin eller genom ett alias eller en skalfunktion, eller som ett skalprogram eller binärfil i /bin, eller ska skrivas in med den fulla sökvägen.

Detta gör alla som kör som root (ibland smärtsamt) medvetna om vilket förtroende han eller hon har för binärfiler. Den vise systemadministratören i ett fleranvändarsystem kollar då och då igenom sina /bin- och /.*history-filer efter mönster eller kryphål.

Den verkligt motiverade administratören kommer lägga märke till sekvenser som kan automatiseras, ställen där förnuftskollar kan läggas in och uppgifter för vilka "root"-privilegierna tillfälligt kan upphävas (köra igång editorer, MTAn och andra stora interaktiva program med goda inställningsmöjligheter som *kan* läggas in transparent eller i datafiler - som den kända ./.exrc för vi och emacs ./.emacs och det änn mer välkända $EXINIT och de inbäddade header/footer-makrona). De kommandona kan naturligtvis köras med något i stil med:

```
cp $data $någon_användares_hem/tmp
su -c $origkommando $diverse_argument
cp $någon_användares_hem/tmp $data
```

(...där det mer specifika beror på kommandot.)

Den sista sortens förbehåll är överdrivna om det handlar om en ensam användares arbetsstation - men de är väldigt bra av princip, om du är administratör av ett fleranvändarsystem - speciellt ett allmänt exponerat system (som de på netcom).

3.7 Hur du ställer in in xdms väljare för host-val. *Arrigo Triulzi,* a.triulzi@ic.ac.uk

1. Modifiera filen som kör igång xdm (oftast /etc/rc/rc.6 eller /etc/rc.local) så att den innehåller de följande raderna i avsnittet om xdm-start.

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

2. Editera /usr/lib/X11/cdm/Xservers och kommentera ut raden som startar servern på den lokala maskinen (d.v.s. startar 0:)
3. Boota om maskinen och du är klar.

Jag lägger till detta eftersom det, när jag desperat försökte konfigurera det för mitt eget subnät, tog mig ungefär en vecka att reda ut alla problemen.

Varning: med gamla SLS (1.1.1) kan du av någon anledning lämna ett -nodaemon efter xdm-raden - detta fungerar **INTE** med senare utgåvor.