Linux-Tips-HOWTO

Paul Anderson <paul@geeky1.ebtech.net>
prevedel Andrej Grauf <andrej.grauf@club.win-ini.si>

v3.6, junij 1998, prevod 22. december 1998

Ta HOWTO vsebuje namige in nasvete, ki jih boste težko našli drugje, naredijo pa Linux prijaznejši.

Kazalo

1 Uvod

Dobrodošli v Linux Tips-HOWTO, seznamu zanimivih zvijač in optimizacij, ki naredijo Linux bolj zabaven. Spis vsebuje samo nasvete iz moje glave in nasvete iz prejšnjega spisa Tips-HOWTO (Zakaj bi izpuščali spodobne nasvete?). Zato bom zelo vesel vaših najljubših namigov in nasvetov, ki jih bom tako lahko uvrstil v naslednji Tips-HOWTO.

Paul Anderson, vzdrževalec Linux TIPS HOWTO

`panderson@ebtech.net`

2 Krajši nasveti

2.1 Uporabna zvijača za Syslog. Paul Anderson, vzdrževalec spisa Tips-HOWTO

Uredite datoteko `/etc/syslog.conf` in vstavite naslednjo vrstico:

```
# Izpiši vse na tty8
                                        *.*       /dev/tty8
```

Opozorilo: NE POZABITE UPORABITI TIPKE TAB! Syslog ne mara presledkov.

2.2 Skripta za ogled stisnjenih HOWTO-jev. Didier Juges, dj@destin.nfds.net

Začetnik začetniku. Tu je kratka skripta, ki vam bo olajšala iskanje in ogled spisov howto. Moji HOWTO-ji so v imeniku `/usr/doc/faq/howto/` in so stisnjeni s programom `gzip`. Imena datotek so *XXX*`-HOWTO.gz`, kjer *XXX* predstavlja temo. V imeniku `/usr/local/sbin` sem ustvaril naslednjo skripto in jo poimenoval "`howto`":

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

Kadar skripto kličete brez argumenta, ta izpiše imenik s HOWTO-ji, ki so na voljo, kadar pa vnesete prvi del imena datoteke (pred vezajem), kot argument, samodejno raztegne in izpiše spis (original pri tem ostane nespremenjen).

Za ogled spisa Serial-HOWTO vnesite:

```
$ howto Serial
```

2.3 Ali je dovolj prostora na disku??? Hans Zoebelein, zocki@gold-fish.cube.net

V tem razdelku boste našli kratko skripto, ki bo od časa do časa preverila, ali je še dovolj prostora na napravah, ki jih lahko izpišete z ukazom `mount` (diski, cdrom, disketa,)

Če zmanjkuje prostora, se na zaslonu vsakih *X* sekund izpiše sporočilo, za vsako zapolnjeno napravo pa boste dobili tudi elektronsko sporočilo.

```
#!/bin/sh

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

#
# Ko so datoteke tmp zapolnile moje diske, sem začel prejemati skrivnostna
# sporočila o napakah. Zato sem spisal tole skripto, ki me opozori, preden
# se diski zapolnijo.
#
# Če vam te vrstice obvarujejo strežnik pred porušitvijo,
# pošljite elektronsko sporočilo z zahvalami na zocki@goldfish.cube.net.
# Če vam tale skripta povzroča težave, mi je strašno žal, ampak
# posvaril sem vas: brez pritožb.
# Če res obvladate delo s sed, mi prosim oprostite :)
#

#
# Poženi in pozabi: V rc.local vstavite 'check_hdspace &'.
# Preveri količino nezasedenega prostora na napravah vsakih $SLEEPTIME sekund.
# Preverite lahko tudi diskete in tračne naprave. :)
# Če je nezasedenega prostora manj kot $MINFREE (kb), bo sprožila opozorilo
# in poslala eno elektronsko sporočilo za vsako triggering napravo
# na $MAIL_TO_ME.
# Če je prostora več, kot je spodnja meja, podana z omejitvijo, pri
# kateri se sprožijo opozorila, se tudi opravilo elektronskega
# pošiljanja ponovno opremi.
#

# NALOGA: Različen $MINFREE za vsako napravo.
# Če zmanjka prostora na disku, iz imenikov /*tmp varno odstrani
# staro šaro.


DEVICES='/dev/sda2 /dev/sda8 /dev/sda9'    # naprava; tu navedite diske
# sproži opozorilo
SLEEPTIME=10                               # sec; mirovanje med preverjanjem
MAIL_TO_ME='root@localhost'               # fool; kdo prejme opozorilo
```

```
# ------ pod to vrstico ni potrebno vnašati sprememb (vsaj upam :) -------

MINMB=0
ISFREE=0
MAILED=""
let MINMB=$MINFREE/1024         # ja, mi smo striktni :)

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  "OPOZORILO: $DEVICE samo $ISMB mb prosto." >&2
              #echo "sem pridejo dodatne stvari" >&2
              echo -e "\a\a\a\a"

              if [ -z  "`echo $MAILED | grep -w $DEVICE`" ] ; then
                     echo "OPOZORILO: Na $DEVICE samo $ISMB mb prosto.      (Sprožilec je nastavljen na $MINMB mb
                     | mail -s "OPOZORILO: Na $DEVICE le $ISMB mb prosto!" $MAIL_TO_ME
                     MAILEDH="$MAILED $DEVICE"
                     MAILED=$MAILEDH
                     # sem vstavite še druga opravila, npr. čiščenje
                     # imenikov */tmp...
              fi
              elif [ -n  "`echo $MAILED | grep -w $DEVICE`" ] ; then
                      # Če je dovolj prostora, skripta odstrani
                      # oznako, da je bila elektronska pošta poslana.
                      # Pripravljeni smo na novo opravilo pošiljanja
                      # elektronske pošte.
                      MAILEDH="`echo $MAILED  | sed s#$DEVICE##`"
                      MAILED=$MAILEDH
              fi

      done
      sleep $SLEEPTIME

done
```

2.4 Pripomoček za odstranjevanje dnevnikov. Paul Anderson, vzdrževalec spisa Tips-HOWTO

Če ste takšni kot jaz, potem imate seznam s 430 naročniki, plus 100+ sporočil na dan preko UUCP. Torej, kaj naj hacker naredi s temi ogromnimi dnevniki? Namesti naj `chklogs`. Avtor programa `chklogs` je Emilio Grimaldo, `grimaldo@panama.iaehv.nl`, trenutno različico 1.8, ki sama po sebi daje dovolj informacij o poteku namestitve (seveda bi bilo dobro preveriti tudi informacije v imeniku `doc`), pa lahko najdete na `<ftp://ftp.iaehv.nl/pub/users/grimaldo/chklogs-1.8.tar.gz>`. Ko boste program namestili, dodajte vnos `crontab` v naslednji obliki:

```
# Vsak dan ob 21:00 zaženi chklogs.
00 21 * * *       /usr/local/sbin/chklogs -m
```

Ko ste že pri tem, ne pozabite omeniti avtorju, kako uporaben program je to :)

2.5 Priročna skripta za odstranjevanje odlagališč. Otto Hammersmith, ohammers@cu-online.com.

Ustvarite datoteko z imenom `rmcores` (avtor jo imenuje `handle-cores`) in naslednjo vsebino:

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

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

  echo Brišem...
find $1 -name core -atime 7 -print -type f -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
```

Naj jo opravilo `cron` zažene vsake toliko časa.

2.6 Prenašanje imenikov med datotečnimi sistemi. Alan Cox, A.Cox@swansea.ac.uk

Hiter način za premik celotne drevesne strukture datotek iz enega na drug disk

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

(Da bi se izognili izgubi imenikov v primeru nesreče, se premaknite iz `cd /source/directory; tar` itd. Hvala Jimu Dennisu, `jim@starshine.org`, ki me je opozoril na to. -Vzdrževalec)

2.7 Iskanje največjih imenikov. Mick Ghazey, mick@lowdown.com

Ali vas je že kdaj zanimalo, kateri imeniki so na vašem računalniku največji? To lahko ugotovite na naslednji način:

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

2.8 Glasilo za Linux

Zasluge ima John Fisk, avtor glasila za Linux. To je odličen e-zine plus in še **brezplačen**! Kaj več bi lahko še zahtevali? Oglejte si `<http://www.linuxgazette.com>`.

MMG, izkaže se, da (1) glasilo za Linux (LG) sedaj izhaja enkrat mesečno in (2) John Fisk ni več njegov urednik, saj sedaj zanj skrbi ekipa pri SSC.

2.9 Kako priti do popravka za program GNU make 3.70 in s tem preprečiti čudno obnašanje VPATH. Ted Stern, stern@amath.washington.edu.

Ne vem, ali ima veliko ljudi ta problem, vendar ima GNU `make` različice 3.70 "lastnost", ki mi ni všeč. Namreč to, da se `VPATH` čudno obnaša, če podate absolutno pot do datoteke. Obstaja zelo dober popravek, ki te težave odpravi in ga lahko dobite od Paula D. Smitha <`psmith@wellfleet.com`>. Po vsaki popravljeni izdaji programa GNU make, ustrezne popravke in dokumentacijo tudi objavi v novičarski skupini "`gnu.utils.lug`". Ta popravek uporabim in ponovno prevedem `gmake` na vsakem sistemu, do katerega imam dostop.

2.10 Kako preprečiti preverjanje sistema s programom fsck ob vsakem zagonu računalnika? Dale Lutz, dal@wimsey.com

V: Kako naj preprečim programu `e2fsckm`, da bi ob vsakem zagonu preverjal trde diske?

O: Ko ponovno zgradite jedro, je datotečni sistem označen kot "umazan", torej bo vaš disk pregledan vsakič, ko boste zagnali računalnik. To lahko preprečite, če zaženete:

```
rdev -R /zImage 1
```

S tem popravite jedro tako, da datotečnega sistema ne bo več obravnaval kot umazanega.

Opozorilo: Če uporabljate lilo, naj bo nastavitev za Linux v nastavitveni datoteki za lilo (običajno je to `/etc/lilo.conf`) namenjena šamo za branje" (read-only).

2.11 Kako se izogniti preverjanju, do katerega pride zaradi žasedenosti naprav". Jon Tombs, jon@gtex02.us.es

Če pri zaustavitvi sistema pogosto dobite sporočilo o zasedenosti naprav, kar povzroči, da se ob ponovnem zagonu izvede preverjanje datotečnega sistema, se lahko temu izognete na naslednji način:

V datoteko `/etc/rc.d/init.d/halt` ali `/etc/rc.d/rc.0` pred izvedbo ukaza `umount -a` dodajte vrstico

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

za vsak priključen datotečni sistem, razen za `/`. To pomeni, da če se pri zaustavitvi zaradi določenih razlogov ne prekinejo vsi procesi in se ne izvede odklop diskov, bodo ti ob ponovnem zagonu vseeno čisti. S tem med zagonom pridobim veliko časa.

2.12 Kako poiskati največje datoteke na trdem disku. Simon Amor, simon@foobar.co.uk

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

Tisti, ki ste zares na tesnem s prostorom, lahko uporabite tudi naslednji način. To bo trajalo nekaj časa, ampak odlično deluje:

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

2.13 Kako natisniti strani, da jih boste lahko kasneje preluknjali in vstavili v mapo. Mike Dickey, mdickey@thorplus.lib.purdue.edu

```
#!/bin/sh
# /usr/local/bin/print
# preprosta urejena oblika izpisa na list, ki ga boste tako lahko
# preluknjali in vstavili v mapo.

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

2.14 Način za iskanje določenih regularnih izrazov v drevesnih strukturah datotek, Raul Deluth Miller, rockwell@nova.umd.edu

To skripto imenujem 'forall' (za vse). Uporabite jo na naslednji način:

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

Pa še vsebina skripte 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 Skripta za odstranjevanje varnostnih kopij datotek in samodejno shranjenih datotek, ki jih tvorijo nekateri programi, Barry Tolnas, tolnas@nestor.engr.utk.edu

Tu je preprosta dvo-vrstična skripta, ki preišče ureditev imenika in odstrani emacs-ove samodejno shranjene (#) datoteke, varnostne kopije datotek (~), datoteke .o in datoteke .log, ki jih naredi TeX. Prav tako jo lahko uporabite za stiskanje datotek .tex in README. Na svojem sistemu sem jo imenoval 'squeeze' (stiskač).

```
#!/bin/sh
#SQUEEZE odstrani nepotrebne datoteke in stisne datoteke .tex in README
#Avtor 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 {} \;
```

2.16 Kako ugotoviti, kateri proces zavzema največ pomnilnika? Simon Amor, simon@foobar.co.uk

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

-ALI-

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

2.17 Priprava programa vi na programiranje v programskem jeziku C, Paul Anderson, Vzdrževalec spisa Tips-HOWTO

V prostem času veliko programiram v programskem jeziku C, zato sem pripravil program `vi`, tako da je primeren za programiranje v tem jeziku. Tukaj je moj `.exrc`:

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

Čemu služijo te nastavitve? `Autoindent` povzroči, da `vi` avtomatsko poravna vsako vrstico glede na poravnavo prve vrstice, `shiftwidth` nastavi razdaljo `^T` na 4 presledke, `backspace` nastavi način backspace, `ruler` pa povzroči, da `vi` izpisuje število vrstic. Da bi se pomaknili na določeno vrstico, npr. v dvajseto, uporabite:

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

2.18 Uporaba ctags za poenostavitev programiranja

Večina hackerjev ima `ctags` že nameščen na svojem računalniku, vendar ga ne uporabljajo. Zelo uporaben je lahko pri urejanju določenih funkcij. Predpostavimo da imate v eni izmed mnogih izvornih datotek v imeniku programa, ki ga pišete, neko funkcijo, ki bi jo radi posodobili. To funkcijo bomo poimenovali `foo()`. Prav tako ne veste, kje v izvorni datoteki ta funkcija je. V tem primeru vam lahko pomaga `ctags`. Ko ga zaženete, program v trenutnem imeniku ustvari datoteko z imenom `tags`, ki vsebuje seznam vseh funkcij, podatke, v katerih datotekah so te funkcije in kje so v datotekah said. Datoteka `tags` izgleda nekako tako:

```
ActiveIconManager     iconmgr.c     /^void ActiveIconManager(active)$/
AddDefaultBindings    add_window.c  /^AddDefaultBindings ()$/
AddEndResize          resize.c      /^AddEndResize(tmp_win)$/
AddFuncButton         menus.c       /^Bool AddFuncButton (num, cont, mods, func, menu, item)$/
AddFuncKey            menus.c       /^Bool AddFuncKey (name, cont, mods, func, menu, win_name, action)$/
AddIconManager        iconmgr.c     /^WList *AddIconManager(tmp_win)$/
AddIconRegion         icons.c       /^AddIconRegion(geom, grav1, grav2, stepx, stepy)$/
AddStartResize        resize.c      /^AddStartResize(tmp_win, x, y, w, h)$/
AddToClientsList      workmgr.c     /^void AddToClientsList (workspace, client)$/
AddToList             list.c        /^AddToList(list_head, name, ptr)$/
```

Da bi uredili npr. funkcijo `AddEndResize()` v programu `vim`, zaženite:

```
vim -t AddEndResize
```

S tem boste ustrezno datoteko odprli v urejevalniku, kazalec pa bo na začetku funkcije.

2.19 Zakaj se sendmail ob zagonu v distribuciji RedHat obesi za 5 minut? Paul Anderson, paul@geeky1.ebtech.net

To je precej pogost problem, skoraj že FAQ. Ne vem, če je RedHat odpravil ta problem v svoji distribuciji, lahko pa ga odpravite sami. Poglejte v datoteko `/etc/hosts` in ugotovili boste, da izgleda nekako tako:

```
127.0.0.1          localhost      vasaskatla
```

Ko se `sendmail` zažene, poišče ime gostitelja (v tem primeru vasaskatla). Potem ugotovi, da je IP za vasaskatla 127.0.0.1. `Sendmail` tega ne mara, zato prične z novim iskanjem. To nadaljuje tako dolgo, dokler končno ne obupa in konča. Odprava tega problema je zelo enostavna. Uredite datoteko `/etc/hosts` in jo preoblikujte nekako tako:

```
127.0.0.1            localhost
10.56.142.1          vasaskatla
```

2.20 Kako lahko preuredim distribucijo RedHat tako, da bom lahko uporabljal ukaz ls z barvami? Paul Anderson, paul@geekey1.ebtech.net

Distribucija RedHat vsebuje barvni `ls`, zakaj pa ne omogočijo uporabo barv kot privzeto nastavitev, pa presega moje znanje. Problem lahko rešite tako:

Najprej vnesite `eval 'DIRCOLORS'`.

Nato `alias ls='ls -color=auto'`.

In vstavite `'alias.....'` v datoteko `/etc/bashrc`.

2.21 vps@unicorn.miimm.spb.su Kako lahko ugotovim, katera knjižnica v /usr/lib vsebuje določeno funkcijo? Pawel Veselow,

Kaj storiti v primeru, če prevajate nek program in pozabite vključiti potrebno knjižnico? Vsa poročila gcc so funkcijska imena.Tu je preprost ukaz, ki bo našel, kar iščete:

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

Kjer je `tgetnum` ime funkcije, ki jo iščete.

2.22 Prevedel sem majhen testni program v C-ju, ko pa ga poženem, ne dobim nobenega rezultata!

Verjetno ste program prevedli v binarno datoteko z imenom `test`, mar ne? Linux vsebuje program z imenom `test`, ki preveri, ali je določen pogoj izpolnjen in rezultata nikoli ne izpiše na zaslon. Namesto vnosa `test` uporabite `./test`.

3 Podrobni nasveti

3.1 Deljenje izmenjalnih particij med Linuxom in Windows. Tony Acero, ace3@midway.uchicago.edu

1. Sformatirajte particijo kot particijo za DOS in na njej ustvarite izmenjalno datoteko za Windows, vendar operacijskega sistema Windows še ne zaženite (naj izmenjalna datoteka zaenkrat ostane še prazna, da jo je možno dobro stisniti).

2. Ponovno zaženite računalnik v Linuxu in shranite particijo v datoteko. Npr. če je bila to particija `/dev/hda8`:

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

3. Stisnite izmenjalno datoteko za DOS; ker je navidezno prazna, jo je mogoče zelo dobro stisniti:

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

4. Da bi pripravili in namestili izmenjalno datoteko v Linuxu, dodajte v datoteko /etc/rc naslednje: *XXXXX* je število blokov v navidezni particiji

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

   Prepričajte se, da ste v datoteko /etc/fstab dodali vnos za izmenjalno particijo.

5. V primeru da vaš osnovni/zagonski paket podpira /etc/brc ali /sbin/brc, v datoteko /etc/brc dodajte naslednje (drugače to opravite ročno, ko želite računalnik zagnati v DOSu|OS/2 in poskušate izmenjalno particijo pretvoriti nazaj v različico za dos/windows):

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

   # Ne pozabite, da se na ta način zapiše nazaj na particijo samo prvih 100 blokov. Empirično sem ugotovil, da to zadošča.

>>Kakšne so prednosti in pomanjkljivosti takšnega početja?

Prednost: prihranite precejšno količino prostora na disku. Pomanjkljivost: Če 5. korak ni avtomatiziran, boste morali to opraviti ročno, kar bo upočasnilo proces ponovnega zagona za nanosekundo :-)

3.2 Obnavljanje zbrisanih datotek. Michael Hamilton, michael@actrix.gen.nz

Tu je trik, ki sem ga morali uporabiti že nekajkrat.

Obnovitev zbrisane tekstovne datoteke obupanega uporabnika.

Če po nesreči odstranite tekstovno datoteko, npr. neko elektronsko sporočilo ali rezultate nočne programerske seanse, ni nujno, da je vse izgubljeno. Če je datoteka bila na disku in na njem ostala več kot 30 sekund, je njena vsebina mogoče še vedno na particiji diska.

Vsebino datoteke na particiji diska lahko poiščete s pomočjo ukaza grep.

Na primer, pred kratkim sem zbrisal del elektronskega sporočila. Takoj sem prekinil vse operacije, ki bi particijo lahko spremenile: v tem primeru sem se samo izognil shranjevanju datotek, prevajanju, itd. V drugih primerih sem sistem moral celo privesti na način enega samega uporabnika in odklopiti datotečni sistem.

Na particiji diska sem potem uporabil ukaz egrep: v mojem primeru je bilo elektronsko sporočilo v imeniku /usr/local/hime/michael. Iz izhoda ukaza df sem lahko ugotovil, da je ta na particiji /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:
[root@sputnik3 michael]# egrep -50 'ftp.+COL' /dev/hdb5 > /tmp/x
```

Kadar delam s particijami diska, sem zelo previden. Torej sem si vzel odmor in se pred potrditvijo prepričal, da res razumem pomen ukaza. V tem primeru je elektronsko sporočilo vsebovalo besedo 'ftp', ki ji je sledil tekst in na koncu še beseda 'COL'. Sporočilo je bilo dolgo okrog 20 vrstic, torej sem uporabil 50, da bi dobil vse vrstice okrog izraza. Včasih sem uporabljal tudi 3000 in s tem zagotovil, da sem dobil nazaj vse vrstice izvorne kode. Rezultat sem iz ukaza egrep usmeril na drugo particijo diska in s tem preprečil, da bi se sporočilo, ki ga iščem, prepisalo.

Izhod sem preiskal z uporabo ukaza strings:

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

in seveda našel elektronsko sporočilo.

Ta metoda ni zanesljiva. Mogoče je bilo nekaj, ali pa celoten prostor na disku, ponovno uporabljeno.

Ta zvijača je verjetno uporabna samo na sistemih z enim uporabnikom. Na sistemih z več uporabniki z visoko dejavnostjo diskov se je mogoče prostor na disku, ki ste ga sprostili, ponovno porabil. Prav tako večina izmed nas ne more kar tako preprečiti uporabe računalnika ostalim uporabnikom, kadar želimo obnoviti neko datoteko.

Na domačem sistemu me je ta zvijača rešila v približno treh primerih v zadnjih nekaj letih po navadi takrat, ko sem po pomoti zavrgel rezultate celodnevnega dela: če moje delo preživi do neke točke, kjer imam občutek, da sem naredil pomemben napredek, to delo shranim na disketo in te zvijače sploh ne uporabljam pogosto.

3.3 Kako uporabljati nespremenljive oznake. Jim Dennis, jadestar@rahul.net

Uporaba nespremenljivih oznak

Takoj po tem, ko boste namestili in nastavili vaš sistem, preglejte datoteke `/bin/sbin`, `/usr/bin`, `/usr/sbin`, `/usr/lib` (in še nekatere druge običajne osumljence) in brez predsodkov uporabite ukaz `chattr +i`. To vrstico dodajte tudi v datoteke jedra v korenskem imeniku. Naredite nov imenik '`mkdir /etc/.dist/`' in prekopirajte vsebino imenika `/etc` (to naredim v dveh korakih z uporabo `/tmp/etcdist.tar` in se s tem izognem podvajanju) v ta novi imenik. (Lahko bi tudi ustvarili imenik `/etc/.dist.tar.gz`) – in ga označili kot nespremenljivega.

Vzrok za vse to delo je zmanjšati škodo, ki bi jo lahko naredili, kadar boste prijavljeni kot korenski uporabnik. Na ta način ne boste mogli prepisati datotek z napačnim preusmeritvenim operatorjem in presledkom v ukazu '`rm -fr`' (še vedno lahko naredite veliko škodo sistema ne boste mogli poškodovati z nepravilno postavljenim vašim podatkom ampak vsaj vaše knjižnice in binarne datoteke bodo bolj zaščitene).

S tem boste prav tako onemogočili ali vsaj otežili uporabo številnih varnostnih in preklicnih storitev (ker večina temelji na prepisovanju datotek med izvajanjem programa SUID, ki *ne nudi poljubnih ukazov lupine*).

Edina neprijetnost se bo pojavila pri grajenju in izvajanju ukaza "`make install`" na različnih vrstah sistemskih binarnih datotek. Po drugi strani pa boste s tem preprečili, da bi '`make install`' prepisal datoteke. Kadar pozabite prebrati `Makefile` in `chattr -i`, datoteke, ki bodo prepisane (in imenike, katerim želite datoteke dodati) – in se ukaz `make` ne izvede, samo uporabite ukaz `chattr` in ga ponovno zaženite. Izkoristite lahko tudi priložnost in prestavite stare binarne datoteke, knjižnice oz. kar koli je že v imeniku `.old/`, jih preimenujete, stisnete ali počnete z njimi kar pač želite.

3.4 Nasveti, kam dati nove stvari. Jim Dennis, jadestar@rahul.net

Vse nove stvari se začnejo pod `/usr/local`! Ali v `/usr/local/'hostname'`. Če je vaša distribucija ena tistih, ki pusti imenik `/usr/local` prazen, tvorite svoje imenike `/usr/local/src`, `/usr/local/bin`, itd. in jih uporabljajte. V primeru da vaša distribucija v drevesno strukturo imenika `/usr/local` shrani stvari, boste verjetno uporabili ukaz '`mkdir /usr/local/'hostname'`' in mu podelili 'wheel' group `+w` (prav tako jih naredim SUID in SGID in s tem zagotovim, da lahko vsak član skupine wheel upravlja samo s svojimi datotekami in da vse ustvarjene datoteke pripadajo wheel skupini).

Disciplinirajte se in nove pakete *Vedno! Vedno! Vedno!* shranite v `/usr/local/src/.from/$KJERKOLI_SEM_JIH_DOBIL/` (za `.tar` oz. katere koli datoteke) in jih zgradite v imeniku `/usr/local/src` (ali `/$HOSTNAME/src`). Prepričajte se, da se namesti pod lokalno ureditev. Če je paket *nujno potrebno* namestiti v `/bin`, `/usr/bin` ali kjer koli drugje podajte symlink od lokalne ureditve do vsakega posameznega elementa.

Vzrok za to – čeprav je več dela – je pomoč pri osamitvi tistih stvari, za katere je potrebno narediti varnostne kopije, jih obnoviti ali ponovno namestiti pri opravilu popolne ponovne namestitve z distribucijskega medija (običajno s CD-ja). Z uporabo imenika `/usr/local/.from` vzdržujete tudi dnevnik z informacijami, kje ste izvorno kodo dobili – kar vam je v pomoč pri iskanju novih različic – je pa nujno potreben, kadar nadzorujete sezname objav glede varnosti.

Eden mojih domačih sistemov (s katerega ključem) je bil sestavljen preden sem začel uporabljati takšen sistem. Še vedno ne *poznam* vsega, po čemer se razlikuje od po namestitvi pripravljenega sistema. To se mi dogaja kljub temu, da zelo malo delam s konfiguracijo domačega sistema in da sem *edina* oseba, ki ga uporablja.

V primerjavi s tem so vsi sistemi, ki sem jih nastavil v službi (ko so mi zaupali vlogo sistemskega administratorja), nastavljeni na ta način – in čeprav so jih upravljali različni ljudje in so bili velikokrat nadgrajeni, pa vendar dobro vem kateri elementi so bili dodani *naknadno* po začetni namestitvi in nastavitvi.

3.5 Kako spremeniti vse velike črke v imenih datotek v majhne črke. Justin Dossey, dossey@ou.edu

V razdelku 2c nasvetov 12. izdaje sem opazil kar nekaj prezahtevnih ali nepotrebnih postopkov. Ker jih je več, vam jih pošiljam:

```
#!/bin/sh
# lowerit
# spremeni vse velike črke v imenih datotek v trenutnem imeniku v majhne črke
# uporabna samo za navadne datoteke--ne spreminja imen imenikov
# preden prepiše obstoječo datoteko, boste morali to potrditi
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
```

Uh, to je pa dolga skripta. Za izvajanje tega opravila jaz ne bi pisal skripte. Namesto tega bi v ukazni vrstici uporabil ukaz:

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

Avtor pravi, da je skripto zapisal v tej obliki zaradi večje razumljivosti (glej spodaj).

V naslednjem primeru, o dodajanju in odstranjevanju uporabnikov, Geoff dela dokaj dobro, razen v zadnjem koraku. Ponoven zagon? Resno upam, da avtor računalnika ne zaganja ponovno, vedno kadar odstrani uporabnika. Opraviti morate samo prva dva koraka. Kakšne procese pa naj bi uporabnik sploh uporabljal? Mogoče irc bot? Postopek ubijete z enostavnim ukazom

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

Primer; uporabniško ime je foo

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

S tem je delo opravljeno; gremo naprej na pozabljeno korensko geslo.

Rešitev, ki je navedena v glasilu, je najbolj univerzalna, ne pa tudi najenostavnejša. Z LILO in loadlin lahko vsak poda zagonski parameter šingle" in računalnik se neposredno zažene v privzeti lupini brez poziva za prijavo in geslo. V tem primeru lahko pred vnosom "init 3" za zagon več-uporabniškega načina, vsakdo spremeni ali zbriše katero koli geslo. Število ponovnih zagonov v tem primeru: 1, število ponovnih zagonov v drugih primerih: 2

Justin Dossey

3.6 Kako nadgraditi sendmail. Paul Anderson, paul@geekyl.ebtech.net

Začeli bomo z neobdelano, čisto izvorno kodo. Najprej si priskrbite izvorno kodo Sendmaila. Zajel sem verzijo 8.9.0, kar pa je zelo zahtevno opravilo, kot boste ugotovili. Kodo sem dobil na `<ftp://ftp.sendmail.org/pub/sendmail/sendmail.8.9.0.tar.gz>`.

Dolga je okrog 1 MB in glede na to, da uporabljam 8.7.6, mislim, da je vredna truda. Če bo to delovalo, boste nedvomno izvedeli, saj sicer ne bom mogel izdati nove HOWTO verzije :)

Ko ste kodo prenesli, jo razpakirajte. V trenutnem imeniku bom ustvaril imenik z imenom `sendmail-8.9.0`. Premaknite se v ta imenik ter preberite datoteki `README` in `RELESE_NOTES` (presenečeni boste nad opravljenimi nadgraditvami). Premaknite se sedaj v `src`. V tem imeniku boste opravili večino dela.

Kratko sporočilo: Sendmail je kratek, zelo uporaben, dobro napisan program. Binarna datoteka se na računalniku 5x86 133 z 32 MB RAM-a prevede v manj kot 5 minutah! Celotno prevajanje in namestitev (brez nastavitev) traja manj kot 15 minut!

Običajno na mojem sistemu ne zaganjam BIND, zaradi tega sem našel naslednje vrstice:

```
# ifndef NAMED_BIND
#  define NAMED_BIND    1     /* uporabi Berkeleyski internetni strežnik */
# endif
```

in spremenil 1 v 0:

```
# ifndef NAMED_BIND
#  define NAMED_BIND    0     /* uporabi Berkeleyski internetni strežnik */
# endif
```

Pri distribuciji Debian 1.3.1 se `db.h` glede na privzeto vrednost namesti v `/usr/include/db`, namesto v `/usr/include`, kjer jo `sendmail` poskuša najti. Pojdite v imenike `src`, `mailstats`, `makemap`, `praliases` in `smrsh` in tam izvedite naslednji ukaz:

```
./Build -I/usr/include/db
```

Ko boste s tem gotovi, `cd ..` in vnesite `make install`. Tako! Program sendmail različice 8.9.0 naj bi bil sedaj nameščen! V tem primeru sem predpostavil, da ste prej že opravili originalno nastavitev. Da bi na mojem sistemu vse potekalo gladko, ker gostim brezplačne poštne seznane za ljudi, ki uporabljajo `majordomo`, sem moral na začetek datoteke `/etc/sendmail.cf` dodati:

```
O DontBlameSendmail=forwardfileinunsafedirpath, forwardfileinunsafedirpathsafe
```

Dandanes je `sendmail` zelo občutljiv na dovoljenja imenikov in datotek ter bo javljal napake o imenikih in datotekah v privzetih imenih ali datotekah `.forward`, ki jih lahko spreminjajo skupine ali so prosto dostopne. Ker ni dobro odpraviti te občutljivosti, uporabljam v konzoli samo način enega uporabnika in se mi je zdelo, da je v redu tudi, če obstaja ta majhna varnostna luknja. YMMV

3.7 Nekaj nasvetov za nove sistemske administratorje, Jim Dennis, jadestar@rahul.net

Ustvarite in vzdržujte `/README.'hostname'` in/ali `/etc/README.'hostname'` [oz. če je le možno `/usr/local/etc/README.'hostname'` - Vzdrževalec.]

Od prvega dne upravljanja dalje sistem zapisuje stvari v omrežni dnevnik. Mogoče boste v root-ov /bash_logout vnesli vrstico "`vi /README.$(ime gostitelja)`". To lahko pravite tudi tako, da napišete skripto za `su` ali `sudo`, ki počne nekaj podobnega:

```
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 -
```

(Dnevnik ustvarite s pomočjo ukaza `typescript`. Prav tako ustvarite funkcijo, ki bo avtomatizirala dopolnjevala in posodabljala dnevnik.)

Priznati moram, da te avtomatizacije nisem vgradil do sedaj sem se zanašal na samodisciplino. Kakor koli že, s to zamislijo sem se igral (naredil sem celo prototip skripte in funkcije lupine, kot lahko vidite.) Nekaj, kar me pri tem moti, je ukaz '`script`' sam. Mislim, da bom moral zajeti izvore in dodati nekaj parametrov ukazne vrstice (za prekinitev/zaustavitev shranjevanja skript iz ukazne vrstice), preden bom začel uporabljati ta način.

Moj zadnji nasvet (v tem trenutku):

Korenska pot mora biti sestavljena iz '`PATH= /bin`'

To je to. Nič drugega v korenski poti. Za vse, kar počne korenski uporabnik, poskrbi symlink iz `/bin`, privzeto ime ali funkcija lupine, oziroma je skripta ali binarna verzija v `/bin` ali pa je izpisana z eksplicitno potjo.

S tem se vsi, ki se prijavljajo kot korenski uporabnik, zavedajo (včasih na zelo boleč način), kako zaupajo binarnim verzijam. Pameten sistemski administrator z več uporabniki bo redno pregledoval datoteke `/bin` in `/.*history`, da bi s tem našel vzorce in zančne napake.

Zares dobro motivirani sistemski administratorji bodo poiskali opravila, ki jih je mogoče avtomatizirati, mesta, kjer je možno vnesti logična preverjanja in opravila (zagon urejevalnikov, MTA in druge obsežne interaktivne programe, z izpopolnjenimi lastnostmi skript, ki jih je mogoče vnašati v prosojne ali podatkovne datoteke kot je neslavni `vi ./.exrc` in `emacs ./.emacs` in celo bolj zahrbten `$EXINIT` in vgrajeni makri v glavi/nogi), ki jih je potrebno opravljati brez korenskih privilegijev. Seveda je takšne vrste ukazov mogoče zaganjati s:

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

(kjer so specifikacije odvisne od ukaza).

Večina teh zadnjih varnostnih ukrepov prenese na domačo delovno postajo ali postajo z enim samim uporabnikom so pa zelo dober način za upravljanje večjega števila uporabnikov še posebej javnosti izpostavljeni sistemi (kot tisti pri netcom-u.)

3.8 Kako nastaviti xdm-ovega izbiralca za izbiro gostitelja. Arrigo Triulzi, a.triulzi@ic.ac.uk

1. Uredite datoteko, ki zažene `xdm` (najverjetneje je to datoteka `/etc/rc/rc.6` ali `/etc/rc.local`), tako da bo v razdelku za zagon `xdm` vsebovala naslednje vrstice:

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

2. Uredite datoteko `/usr/lib/X11/xdm/Xservers` in odstranite znak za komentar v vrstici, ki zažene strežnik na lokalnem računalniku (npr. z začetnim nizom `0:`)

3. Ponovno zaženite računalnik in delo je opravljeno.

To sem dodal zato, ker sem ga hotel nastaviti tudi za svojo podmrežo, pa sem za odpravo vseh problemov potreboval skoraj en teden.

Opozorilo: s starimi distribucijami SLS (1.1.1) lahko po vrstici za klic `xdm` izpustite `-nodaemon`: to NE deluje pri kasnejših izdajah.