Ελληνικό Linux Tips HOWTO

Paul Anderson, paul@geeky1.ebtech.net v3.1, 26 Δεκεμβρίου 1997

Αυτό το HOWTO περιέχει υποδείξεις και πειραματισμούς, που είναι δύσκολο να βρεθούν και κάνουν το Linux ομορφότερο.

Contents

1 Εισαγωγή

Καλώς ήρθατε στο **Linux Tips HOWTO**, μια λίστα από συμπαθητικά τρυκ και βελτιστοποιήσεις που θα κάνουν το Linux πιο ευχάριστο. Όλα όσα έχω εδώ είναι υποδείξεις που έβγαλα απ'το κεφάλι μου, και συμβουλές από το παλιό Tips-HOWTO (γιατί, άλλωστε, να βγάλω έξω καθώς πρέπει συμβουλές, σωστά;). Στείλτε μου, λοιπόν, όλες τις αγαπημένες σας μικρές χρήσιμες συμβουλές και οδηγίες ώστε να μπορέσω να τις ενσωματώσω στο επόμενο Tips-HOWTO.

Paul Anderson *Συντηρητής–Linux TIPS HOWTO*

`panderso@ebtech.net`

2 Σύντομες Υποδείξεις

2.1 Πρόχειρο Syslog Τρυκ *Paul Anderson, Συντηρητής του Tips-HOWTO*

Ανοίξτε το αρχείο /etc/syslog.conf, και προσθέστε την παρακάτω γραμμή:

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

ΠΡΟΕΙΔΟΠΟΙΗΣΗ: *ΜΗΝ ΞΕΧΑΣΕΤΕ ΝΑ ΧΡΗΣΙΜΟΠΟΙΗΣΕΤΕ TABS!* στο syslog δεν αρέσουν τα κενά...

2.2 Script για να βλέπετε τα συμπιεσμένα HOWTOs. *Didier Juges,* `dj@destin.nfds.net`

Απ' τον ένα αρχάριο στον άλλο, προέκυψε αυτό το μικρό script που διευκολύνει το ψάξιμο και το διάβασμα κειμένων howto. Τα howto αρχεία μου βρίσκονται στο /usr/doc/faq/howto/ και είναι συμπιεσμένα. Τα ονόματα των αρχείων είναι XXX-HOWTO.gz, όπου XXX είναι το θέμα του howto. Δημιούργησα το script με όνομα "howto" στον κατάλογο the /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
```

Όταν καλείται χωρίς παραμέτρους, εμφανίζει ο κατάλογος των διαθέσιμων αρχείων howto. Όταν καλείται με παράμετρο το πρώτο τμήμα του ονόματος του αρχείου (πριν την παύλα), το αποσυμπιέζει (κρατώντας το αρχικό αρχείο ανέπαφο) και μετά εμφανίζει το κείμενο.

Για παράδειγμα, για να δείτε το κείμενο Serial-HOWTO.gz , γράψτε:

$ howto Serial

2.3 Υπάρχει αρκετός ελεύθερος χώρος;;; *Hans Zoebelein,* zocki@goldfish.cube.net

Ακολουθεί ένα μικρό script το οποίο ελέγχει περιοδικά αν υπάρχει αρκετός ελεύθερος χώρος διαθέσιμος σε οτιδήποτε εμφανίζεται στο mount (δίσκοι, cdrom, δισκέτα...)

Αν ο χώρος εξαντλείται, τυπώνεται ένα μήνυμα κάθε X δευτερόλεπτα στην οθόνη και στέλνεται 1 μήνυμα με mail για κάθε γεμάτη συσκευή.

```
#!/bin/sh

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

#
# Since I got mysterious error messages during compile when
# tmp files filled up my disks, I wrote this to get a warning
# before disks are full.
#
# If this stuff saved your servers from exploding,
# send praising email to zocki@goldfish.cube.net.
# If your site burns down because of this, sorry but I
# warned you: no comps.
# If you really know how to handle sed, please forgive me :)
#

#
# Shoot and forget: Put 'check_hdspace &' in rc.local.
# Checks for free space on devices every $SLEEPTIME sec.
# You even might check your floppies or tape drives. :)
# If free space is below $MINFREE (kb), it will echo a warning
# and send one mail for each triggering device to $MAIL_TO_ME.
# If there is more free space than trigger limit again,
# mail action is also armed again.
#

# TODO: Different $MINFREE for each device.
# Free /*tmp dirs securely from old junk stuff if no more free space.
```

```
DEVICES='/dev/sda2 /dev/sda8 /dev/sda9'        # device; your put disks here
MINFREE=20480                                  # kb; below this do warning
SLEEPTIME=10                                   # sec; sleep between checks
MAIL_TO_ME='root@localhost'                    # fool; to whom mail warning


# ------- no changes needed below this line (hopefully :) -------

MINMB=0
ISFREE=0
MAILED=""
let MINMB=$MINFREE/1024         # yep, we are strict :)

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  "WARNING: $DEVICE only $ISMB mb free." >&2
                        #echo "more stuff here" >&2
                        echo -e "\a\a\a\a"

                        if [ -z  "`echo $MAILED | grep -w $DEVICE`" ] ; then
                                echo "WARNING: $DEVICE only $ISMB mb free.      (Trigger is set to $MINMB
                                | mail -s "WARNING: $DEVICE only $ISMB mb free!" $MAIL_TO_ME
                                MAILEDH="$MAILED $DEVICE"
                                MAILED=$MAILEDH
                                # put further action here like cleaning
                                # up */tmp dirs...
                        fi
                        elif [ -n  "`echo $MAILED | grep -w $DEVICE`" ] ; then
                                # Remove mailed marker if enough disk space
                                # again. So we are ready for new mailing action.
                                MAILEDH="`echo $MAILED  | sed s#$DEVICE##`"
                                MAILED=$MAILEDH
                        fi

                done
                sleep $SLEEPTIME

done
```

2.4 Βοηθητική εφαρμογή για να καθαρίζετε τα log αρχεία σας. *Paul Anderson, Συντηρητής του Tips-HOWTO* >

Αν είστε σαν κι εμένα, έχετε μια λίστα με 250 εγγεγραμμένους, και 100+ μηνύματα να έρχονται καθημερινά μέσω UUCP. Λοιπόν, τι μπορεί να κάνει ένας χάκερ με αυτά τα τεράστια logs; Να στήσει το chklogs. Το Chklogs γράφτηκε από τον Emilio Grimaldo, `grimaldo@panama.iaehv.nl`, και η τρέχουσα έκδοση 1.8 είναι διαθέσιμη στο ftp.iaehv.nl:/pub/users/grimaldo/chklogs-1.8.tar.gz. Η διαδικασία εγκατάστασης είναι απλή (θα διαβάσετε, φυσικά, τις πληροφορίες στον υποκατάλογο sub). Όταν το εγκαταστήσετε, προσθέστε μια εγγραφή crontab σαν την παρακάτω:

```
# Run chklogs at 9:00PM daily.
00 21 * * *       /usr/local/sbin/chklogs -m
```

Μια και το χρησιμοποιήσατε, μην ξεχάσετε να αναφέρετε στον συγγραφέα πόσο ωραίο προγραμματάκι είναι αυτό :)

2.5 Πρόχειρο Script για να καθαρίζετε τα Core αρχεία. *Otto Hammersmith,* ohammers@cu-online.com

Δημιουργήστε ένα αρχείο με όνομα rmcores (ο συγγραφέας το αποκαλεί handle-cores) που να περιέχει τον παρακάτω κώδικα:

```
#!/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
```

Δημιουργήστε μια εργασία cron να το τρέχει περιοδικά.

2.6 Μεταφέροντας καταλόγους μεταξύ σύστημα αρχείων s. *Alan Cox,* A.Cox@swansea.ac.uk

Γρήγορος τρόπος να μεταφέρετε ένα ολόκληρο δένδρο αρχείων από ένα δίσκο σε έναν άλλο

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

[ΑΑλλάξτε από το /source/directory. Κάντε tar....κλπ. ώστε να αποφύγετε την πιθανότητα να αχρηστεύσετε το directory σε περίπτωση καταστροφής. Ευχαριστώ τον Jim Dennis, jadestar@rahul.net, που μου το είπε. -Συντ.]

2.7 Βρείτε τους μεγαλύτερους καταλόγους που έχετε. *Mick Ghazey,* mghazey@miso.lowdown.com

Έχετε αναρωτηθεί ποτέ ποιοι είναι οι μεγαλύτεροι κατάλογοι στον υπολογιστή σας; Ορίστε πώς θα το μάθετε.

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

2.8 To Linux Gazette

Δόξα και τιμές στον John Fisk, το δημιουργό του Linux Gazette. Είναι ένα εξαιρετικό ηλεκτρονικό περιοδικό και επιπλέον, είναι **ΔΩΡΕΑΝ!!!** Τι περισσότερο μπορείτε να ζητήσετε; Δείτε μόνοι σας:

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

Επί την ευκαιρία, φαίνεται ότι (1) το LG βγαίνει σε μηνιαία έκδοση, και (2) δεν το συντηρεί πια ο John Fisk , αλλά οι συνεργάτες της SSC.

2.9 Δείκτης στο patch για το GNU Make 3.70 για να αλλάξετε τη συμπεριφορά του VPATH. *Ted Stern,* `stern@amath.washington.edu`

Δεν ξέρω αν έχουν πολλοί άνθρωποι αυτό το πρόβλημα, αλλά υπάρχει ένα ”χαρακτηριστικό” του GNU make έκδοση 3.70 το οποίο δε μου αρέσει. Είναι ότι το VPATH συμπεριφέρεται περίεργα αν δώσετε το απόλυτο όνομα αρχείου. Υπάρχει ένα εξαιρετικά πλήρες patch που διορθώνει το πρόβλημα, το οποίο μπορείτε να βρείτε από τον Paul D. Smith <`psmith@wellfleet.com`>. Αυτός στέλνει την τεκμηρίωση και το patch μετά από κάθε επανέκδοση του GNU make στο newsgroup gnu.utils.bug. Γενικά, εφαρμόζω αυτό το patch και κάνω ξανά compile το gmake σε κάθε σύστημα που έχω πρόσβαση.

2.10 Πώς σταματώ το σύστημα μου να κάνει fsck σε κάθε επανεκκίνηση; *Dale Lutz,* `dal@wimsey.com`

Q: Πώς σταματώ το e2fsck να ελέγχει το δίσκο μου κάθε φορά που κάνω boot.

A: Όταν ξαναχτίσετε τον πυρήνα, το σύστημα αρχείων μαρκάρεται σαν ’βρώμικο’ και έτσι ο δίσκος σας θα ελέγχεται σε κάθε εκκίνηση. Για να το φτιάξετε αυτό τρέξτε το:

rdev -R /zImage 1

Αυτό διορθώνει τον πυρήνα από το να πιστεύει ότι το σύστημα αρχείων είναι βρώμικο.

Σημείωση: Αν χρησιμοποιείτε lilo, τότε προσθέστε το `read-only` *στην εγκατάσταση του στο αρχείο ρύθμισης του lilo (συνήθως στο /etc/lilo.conf)*

2.11 Πώς να αποφύγετε τα fscks που προκαλούνται από ”device busy” την ώρα της επανεκκίνησης. *Jon Tombs,* `jon@gtex02.us.es`

Αν βλέπετε συχνά λάθη device busy κατά το shutdown που κάνουν το σύστημα αρχείων να χρειάζεται fsck κατά την επανεκκίνηση, ορίστε μια μικρή διόρθωση :

Προσθέστε την παρακάτω γραμμή στο `/etc/rc.d/init.d/halt` ή `/etc/rc.d/rc.0`

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

για όλα τα προσαρτημένα (mounted) συστήματα αρχείων εκτός του /, πριν την κλήση umount -a. Αυτό σημαίνει ότι αν, για κάποιο λόγο, το shutdown αποτύχει να σκοτώσει όλες τις διεργασίες και αποπροσαρτήσει τους δίσκους θα είναι ακόμη καθαροί στην επανεκκίνηση. Έτσι εξοικονομώ πολύ χρόνο στην επανεκκίνηση.

2.12 Πώς να βρείτε τα μεγαλύτερα αρχεία στο δίσκο σας.

Simon Amor, `simon@foobar.co.uk`

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

Η, γι' αυτούς που είναι πραγματικά scrunched για χώρο, μπορεί να αργεί λίγο αλλά δουλεύει μια χαρά:

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

2.13 Πώς να τυπώσετε σελίδες και να αφήσετε περιθώριο για το διακορευτή. *Mike Dickey,* mdickey@thorplus.lib.purdue.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
```

2.14 Ενας τρόπος για να ψάχνετε μέσα σε δένδρα αρχείων για μια συγκεκριμένη κανονική έκφραση. *Raul Deluth Miller,* rockwell@nova.umd.edu

Ονομάζω αυτό το script 'forall'. Χρησιμοποιήστε το ως εξής :

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

Να το script 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 Ένα script για να καθαρίζει τα αρχεία autosave και backup που δημιουργούν διάφορα προγράμματα. *Barry Tolnas,* tolnas@nestor.engr.utk.edu

Ένα απλό πρόγραμμα δύο-γραμμών που κατεβαίνει αναδρομικά την ιεραρχία ενός καταλόγου αφαιρώντας τα αρχεία emacs auto-save (#) και backup (~), τα αρχεία .o , και τα TeX .log. Επίσης συμπιέζει τα αρχεία .tex και τα README. Το έχω ονομάσει 'squeeze' στο σύστημά μου.

```
#!/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 {} \;
```

2.16 Πώς να βρείτε ποια διεργασία απαιτεί την περισσότερη μνήμη. *Simon Amor,* simon@foobar.co.uk

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

-H-

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

2.17 Εξοπλίστε το vi για προγραμματισμό σε C , *Paul Anderson,*Tips-HOWTO Συντηρητής

Έχω προγραμματίσει πολύ σε C στον ελεύθερό μου χρόνο, και έδωσα χρόνο για να εξοπλίσω το vi ώστε να είναι φιλικό με τη C. Ορίστε το αρχείο μου .exrc:

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

Τι κάνει αυτό; Το autoindent αναγκάζει το vi να κάνει indent αυτόματα κάθε γραμμή που ακολουθεί την πρώτη που έγινε indent, το shiftwidth θέτει την απόσταση του ^T (tab) σε 4 κενά, το backspace θέτει το backspace mode, και το ruler δείχνει τον αριθμό γραμμής. Να θυμάστε, ότι για να πάτε σε ένα συγκεκριμένο αριθμό γραμμής, ας πούμε την 20, χρησιμοποιήστε το:

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

2.18 Γιατί το sendmail κολλάει επί 5 λεπτά κατά την εκκίνηση στο RedHat; *Paul Anderson,* paul@geeky1.ebtech.net

Αυτό είναι ένα σχεδόν κοινό πρόβλημα, σε σημείο να γίνει FAQ. Δεν ξέρω αν το RedHat διόρθωσε αυτό το bug στη διανομή τους, αλλά μπορείτε να το διορθώσετε και μόνοι σας. Αν κοιτάξετε στο αρχείο /etc/hosts , θα δείτε ότι μοιάζει κάπως σαν :

```
127.0.0.1               localhost       yourbox
```

Όταν ξεκινά το sendmail, ψάχνει στο hostname που έχετε (σε αυτή την περίπτωση, το yourbox). Εκεί βρίσκει ότι το IP του yourbox είναι 127.0.0.1, αυτό δεν αρέσει στο sendmail, και ψάχνει πάλι. Συνεχίζει με αυτό για λίγο μέχρις ότου να εγκαταλείψει και να σταματήσει. Το να διορθώσετε αυτό το πρόβλημα είναι πολύ απλό, ανοίξτε το αρχείο /etc/hosts και αλλάξτε το σε κάτι σαν το:

```
127.0.0.1               localhost
10.56.142.1             yourbox
```

2.19 Πώς ρυθμίζω το RedHat για να χρησιμοποιεί την color-ls; *Paul Anderson,* paul@geeky1.ebtech.net

Η διανομή RedHat έρχεται με color-ls, παρόλαυτά δεν ξέρω γιατί δεν τη ρυθμίζουν για χρήση με χρώματα. Να πώς θα το διορθώσετε.

Πρώτα, γράψτε eval ‘DIRCOLORS‘

Μετά, alias ls='ls –color=yes'

Και προσθέστε το 'alias.....' στο αρχείο /etc/bashrc

2.20 Πώς βρίσκω ποια βιβλιοθήκη στο /usr/lib έχει μια συγκεκριμένη συνάρτηση; *Pawel Veselow,* `vps@unicorn.niimm.spb.su`

Τι κάνετε αν κάνετε compile και δε κάνατε link μια βιβλιοθήκη που χρειάζεται; Όλα τα gcc reports είναι ονόματα συναρτήσεων... Να μια εντολή δείγμα που θα βρει αυτό που ψάχνετε:

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

όπου tgetnum είναι το όνομα της συνάρτησης που ψάχνετε.

3 Λεπτομερείς Υποδείξεις

3.1 Μοιράζοντας τα swap partitions μεταξύ Linux και Windows. *Tony Acero,* `ace3@midway.uchicago.edu`

1. Φορμάρετε το partition σαν dos partition, και δημιουργήστε το αρχείο swap των Windows σε αυτό, αλλά μην τρέξετε ακόμη τα windows. (Θέλετε να κρατήσετε το αρχείο swap τελείως κενό για την ώρα, ώστε να συμπιεστεί καλά).

2. Ξεκινήστε linux και σώστε το partition σε ένα αρχείο. Για παράδειγμα αν το partition ήταν το /dev/hda8:

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

3. Συμπιέστε το αρχείο swap του dos. Μια και είναι σχεδόν όλα 0 θα συμπιεστεί πολύ καλά

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

4. Προσθέστε το παρακάτω στο αρχείο /etc/rc για να προετοιμάσετε και να στήσετε το swap χώρο κάτω από Linux:

   XXXXX είναι ο αριθμός των μπλοκς του swap partition

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

   Σιγουρευτείτε ότι προσθέσατε μια γραμμή για το swap partition στο αρχείο /etc/fstab

5. Αν το πακέτο σας init/reboot υποστηρίζει /etc/brc ή /sbin/brc προσθέστε τα παρακάτω στο /etc/brc, αλλιώς κάντε το με το χέρι όταν θέλετε να ξεκινήσετε σε dos|os/2 και θέλετε να μετατρέψετε το swap partition πίσω στην έκδοση για dos/windows:

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

Note that this only writes the first 100 blocks back to the partition. I've found empirically that this is sufficient

>> Ποιά είναι τα πλεονεκτήματα και τα μειονεκτήματα για το παραπάνω;

Πλεονέκτημα: εξοικονομείτε ουσιαστικό χώρο από το δίσκο σας.

Μειονέκτημα: αν το βήμα 5 δε γίνεται αυτόματα, πρέπει να θυμηθείτε να το κάνετε με το χέρι, και επιβαρύνει τη διαδικασία εκκίνησης κατά ένα nanosecond :-)

3.2 Απελπισμένα Undelete. *Michael Hamilton,* michael@actrix.gen.nz

Να ένα τρυκ που έπρεπε να χρησιμοποιήσω μερικές φορές.

undelete αρχείου κειμένου για απελπισμένους.

Αν κατά λάθος σβήσατε ένα text αρχείο, για παράδειγμα, κάποιο email, ή τα αποτελέσματα μιας ολονύχτιας προγραμματιστικής συνεδρίας , μπορεί να μη χάθηκαν όλα. Αν το αρχείο που φτιάξατε στο δίσκο, ήταν εκεί π.χ για περισσότερο από 30 δευτερόλεπτα, μπορεί τα περιεχόμενά του να βρίσκονται ακόμη στην κατάτμηση του δίσκου.

Μπορείτε να χρησιμοποιήσετε την εντολή grep για να ψάξει το raw disk partition για τα περιεχόμενα του αρχείου.

Για παράδειγμα, πρόσφατα, έσβησα κατά λάθος ένα κομμάτι ενός email. Έτσι, σταμάτησα αμέσως οποιαδήποτε ενέργεια που μπορούσε να μεταβάλλει το partition: σε αυτή την περίπτωση απλά απέφυγα να αποθηκεύσω αρχεία ή να κάνω μεταγλώττιση (compile) κλπ. Σε άλλες περιπτώσεις, μπήκα πραγματικά στον κόπο να κατεβάσω το σύστημα σε single user mode, και να αποδεσμεύσω (unmount) το σύστημα αρχείων.

Χρησιμοποίησα τότε την εντολή egrep στο partition του δίσκου: στην περίπτωσή μου το email μήνυμα ήταν στο /usr/local/home/michael/, έτσι από το output του df, είδα ότι ήμουν στο /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
```

Τώρα είμαι υπερβολικά προσεκτικός όταν παίζω με τις κατατμήσεις των δίσκων, έτσι σταμάτησα να βεβαιωθώ ότι καταλαβαίνω τη σύνταξη της εντολής ΠΡΙΝ να πατήσω το πλήκτρο return. Σε αυτή την περίπτωση το email περιείχε τη λέξη 'ftp' ακολουθούμενη από κάποιο text που το ακολουθούσε το 'COL'. Το μήνυμα ήταν περίπου 20 γραμμές, έτσι χρησιμοποίησα το -50 για να φέρω όλες τις λέξεις γύρω από τη φράση. Στο παρελθόν είχα χρησιμοποιήσει το -3000 για να είμαι σίγουρος ότι θα πάρω όλες τις γραμμές από κάποιο πηγαίο κώδικα. Κατεύθυνα την έξοδο της egrep σε ένα διαφορετικό partition δίσκου- έτσι απέφυγα να γράψω πάνω από το παλιό μήνυμα που έψαχνα.

Μετά χρησιμοποίησα την εντολή strings για να με βοηθήσει να εξετάσω την έξοδο

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

Το email ήταν πραγματικά εκεί.

Αυτή η μέθοδος δεν είναι για να βασιστείτε σε αυτή, όλος, ή μέρος, του χώρου του δίσκου μπορεί να έχει ήδη ξαναχρησιμοποιηθεί.

Αυτό το τρυκ είναι πιθανά χρήσιμο μόνο σε συστήματα που δουλεύει ένας χρήστης. Σε συστήματα πολλαπλών χρηστών με υψηλή δραστηριότητα δίσκου, ο χώρος που ελευθερώσατε μπορεί να έχει ήδη χρησιμοποιηθεί. Ακόμη περισσότερο, δεν μπορούμε να τραβάμε το χαλί κάτω από τους χρήστες μας κάθε φορά που χρειαζόμαστε να ανακτήσουμε ένα αρχείο.

Στο σύστημα που έχω σπίτι, αυτό το τρυκ φάνηκε βολικό σε τρεις περίπου περιπτώσεις τα τελευταία χρόνια - συνήθως όταν σβήνω τυχαία μερικές μέρες δουλειάς. Αν ό,τι δουλεύω φτάνει σε ένα σημείο όπου νιώθω ότι έχω κάνει σημαντική βελτίωση, κάνω αντίγραφο ασφαλείας σε ένα floppy, έτσι δε χρειάζομαι αυτό το τρυκ και πολύ συχνά.

3.3 Πώς να χρησιμοποιήσετε το σταθερό flag. *Jim Dennis,* `jadestar@rahul.net`

Χρησιμοποιήσετε το σταθερό Flag

Αμέσως μόλις εγκαταστήσετε και ρυθμίσετε το σύστημά σας πηγαίνετε στα /bin, /sbin/, /usr/bin, /usr/sbin και /usr/lib (και μερικά of the other usual suspects και κάντε γενναιόδωρη χρήση της 'chattr +i εντολής'. Επίσης προσθέστε τα αρχεία του πυρήνα στο root. Κάντε 'mkdir /etc/.dist/' αντιγράψτε τα όλα από το /etc/ on down (αυτό το κάνω σε δύο βήματα χρησιμοποιώντας το /tmp/etcdist.tar για να αποφύγω την αναδρομή) σε εκείνο τον κατάλογο. (Προαιρετικά μπορείτε να δημιουργήσετε το /etc/.dist.tar.gz) – και να το μαρκάρετε σα σταθερό.

Ο λόγος για όλα αυτά είναι να ελαττώσουμε τη ζημιά που μπορεί να προκαλέσετε αν μπείτε σαν root. Δε θα γράψετε πάνω από αρχεία με ένα τελεστή ανακατεύθυνσης (redirection operator) stray, και δε θα κάνετε το σύστημα άχρηστο με ένα stray κενό σε μια εντολή 'rm -fr' (μπορείτε ακόμη να κάνετε πολλές ζημιές στα δεδομένα σας – αλλά οι βιβλιοθήκες και τα εκτελέσιμα θα είναι πιο ασφαλή.

Αυτό επίσης κάνει ένα πλήθος προβλημάτων ασφάλειας και άρνησης εξυπηρέτησης (denial of service) είτε αδύνατο είτε πιο δύσκολο να δημιουργηθούν (μια και πολλοί βασίζονται στο να γράφουν πάνω από ένα αρχείο μέσω των λειτουργιών κάποιου προγράμματος SUID που *δεν παρέχει μια αυθαίρετη γραμμή εντολών*).

Ο μόνος μπελάς σε αυτό είναι όταν χτίζετε τον πυρήνα και κάνετε 'make install' σε διάφορα είδη εκτελέσιμων του συστήματος. Από την άλλη, εμποδίζει επίσης το 'make install' να γράφει πάνω από τα αρχεία. Όταν ξεχάσετε να διαβάσετε το Makefile και να κάνετε chattr -i τα αρχεία που πρόκειται να γραφτούν από πάνω (και οι κατάλογοι στους οποίους θέλετε να προσθέσετε αρχεία) – το make αποτυγχάνει, χρησιμοποιείτε απλά την εντολή chattr και το ξανατρέχετε. Μπορείτε επίσης να αρπάξετε αυτή την ευκαιρία για να μεταφέρετε τα παλιά σας bin's, lib's, ή οτιδήποτε σε ένα κατάλογο .old/ ή να του αλλάξετε όνομα ή να τα κάνετε tar ή ότι άλλο.

3.4 Μια πρόταση για το πού να τοποθετήσετε νέα αρχεία. *Jim Dennis,* `jadestar@rahul.net`

Όλα τα νέα αρχεία μπαίνουν κάτω από το /usr/local! ή /usr/local/'hostname'

Αν η διανομή σας αφήνει το /usr/local άδειο τότε δημιουργήστε το /usr/local/src, /usr/local/bin κλπ. και χρησιμοποιήστε τα. Αν η διανομή σας εγκαθιστά προγράμματα στο δένδρο /usr/local τότε μπορεί να θέλετε να κάνετε 'mkdir /usr/local/'hostname'' και να δώσετε το 'wheel' group +w σε αυτό (το κάνω επίσης SUID και SGID για να εξασφαλίσω ότι κάθε μέλος του wheel group μπορεί να ανακατεύεται με τα δικά του αρχεία από εκεί και κάτω, και ότι όλα τα αρχεία που δημιουργήθηκαν θα ανήκουν στο 'wheel' group.

Τώρα πειθαρχήστε τον εαυτό σας και *ΠΑΝΤΑ! ΠΑΝΤΑ! ΠΑΝΤΑ!* βάλτε τα νέα πακέτα κάτω από το /usr/local/src/.from/$ΟΠΟΥ_ΤΑ_ΕΒΑ•Α/ (για τα .tar ή οποιαδήποτε αρχεία) και χτίστε τα κάτω από το /usr/local/src (ή .../$HOSTNAME/src). Σιγουρευτείτε ότι εγκαθίσταται κάτω από την τοπική ιεραρχία. Αν *πρέπει υποχρεωτικά* να εγκατασταθεί πίσω στο /bin ή /usr/bin ή κάπου αλλού – βάλτε ένα symlink από την τοπική ιεραρχία σε κάθε στοιχείο του από να το βάλετε οπουδήποτε αλλού.

Ο λόγος γι' αυτό – ακόμη κι αν απαιτεί περισσότερη δουλειά – είναι ότι βοηθά να απομονώσετε ό,τι πρέπει να γίνει αντίγραφο ασφαλείας και restore ή να ξαναεγκαταστηθεί σε περίπτωση πλήρους επανεγκατάστασης από τα distribution media (συνήθως CD στις μέρες μας). Χρησιμοποιώντας το /usr/local/.from directory μπορείτε να κρατήσετε ένα ανεπίσημο ημερολόγιο (log) από που προέρχονται οι πηγαίοι κώδικες – το οποίο βοηθάει όταν ψάχνετε για νέες αναβαθμίσεις – και μπορεί να είναι πολύ αποφασιστικό όταν κάνετε monitore τις λίστες ανακοινώσεων σχετικά με ασφάλεια.

Ε-να από τα συστήματά μου στο σπίτι (αυτό από το οποίο με το οποίο κάνω και τις συνδέσεις μου) συναρμολογήθηκε πριν να υιοθετήσω αυτή την τακτική για τον εαυτό μου. Ακόμη δε "γνωρίζω" όλες τις διαφορές από το σύστημα που αφήνεται "όπως εγκαταστάθηκε". Αυτό είναι παρά το γεγονός ότι έχω κάνει πολύ λίγες ρυθμίσεις στο σύστημα του σπιτιού μου και είμαι το *μόνο* άτομο που το χρησιμοποιεί.

Αντίθετα, τα συστήματα που έχω στήσει στη δουλειά (όταν απέκτησα εκεί το ρόλο του διαχειριστή συστήματος) έχουν όλα ρυθμιστεί με αυτόν τον τρόπο – έχουν διαχειριστεί από πολλούς ανάδοχους και άλλους ανθρώπους του MIS, και έκαναν ένα μεγάλο αριθμό αναβαθμίσεων και εγκαταστάσεων πακέτων. Έχω μια πολύ καλή ιδέα για το ποια ακριβή στοιχεία τοποθετήθηκαν στο σύστημα *μετά* την αρχική εγκατάσταση και ρύθμιση.

3.5 Μετατρέψτε όλα τα ονόματα των αρχείων ενός καταλόγου σε μικρά γράμματα. *Justin Dossey,* dossey@ou.edu

Παρατήρησα μερικές υπερβολικά δύσκολες ή μη απαραίτητες διαδικασίες που προτείνονται στο τμήμα 2c tips του τεύχους 12. Μια και είναι περισσότερες από μία, τις στέλνω σε σένα:

```
#!/bin/sh
         # lowerit
         # convert all file names in the current directory to lower case
         # only operates on plain files--does not change the name of directories
         # will ask for verification before overwriting an existing file
         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
```

Ωπ, αυτό είναι μεγάλο script. Δε θα έγραφα τέτοιο script για να κάνω αυτό. Αντίθετα, θα χρησιμοποιούσα αυτή την εντολή:

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

στη γραμμή εντολών.

Ο άνθρωπος που έκανε αυτή τη συνεισφορά, λέει ότι έγραψε το script με αυτό τον τρόπο ώστε να είναι κατανοητό (δείτε παρακάτω).

Στην επόμενη υπόδειξη, αυτή σχετικά με την προσθήκη και διαγραφή χρηστών, ο Geoff τα πάει μια χαρά μέχρι το τελευταίο βήμα. Επανεκκίνηση; Ωχ, ελπίζω να μη χρειάζεται επανεκκίνηση του συστήματος κάθε φορά που διαγράφει ένα χρήστη. Ο,τι χρειάζεται να κάνετε είναι τα δύο πρώτα βήματα. Τι είδους διεργασίες θα έχει αφήσει ο χρήστης, άλλωστε; Ένα irc bot; Σκοτώστε τις διαδικασίες απλά με την

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

Παράδειγμα, το username είναι foo

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

Τώρα που φροντίσαμε κι αυτό, ας προχωρήσουμε στο ξεχασμένο κωδικό του root.

Η λύση που δίνεται στο Gazette είναι η πιο κοινή, αλλά όχι και η πιο εύκολη. Με το LILO και το loadlin, μπορείτε να δώσετε την παράμετρο για boot σε περιβάλλον ενός χρήστη (single) για αυτόματη εκκίνηση στο προκαθορισμένο shell χωρίς login ή password prompt. Από εκεί, μπορείτε να αλλάξετε ή να αφαιρέσετε passwords πριν να γράψετε "init 3" να ξεκινήσετε σε κατάσταση πολλαπλών χρηστών (multiuser mode). Αριθμός Επανεκκινήσεων: 1 Με τον άλλο τρόπο Αριθμός Επανεκκινήσεων: 2

Justin Dossey

3.6 Μερικές υποδείξεις για νέους sysadmins. *Jim Dennis,* jadestar@rahul.net

Δημιουργήστε και συντηρήστε ένα /README.'hostname' και/ή ένα /etc/README.'hostname' *[Η πιθανά /usr/local/etc/README.'hostname' -Maint.]*

Οπωσδήποτε, κρατήστε σημειώσεις από *την πρώτη μέρα* της διαχείρισης του συστήματος σε ένα online log αρχείο. Μπορείτε να φτιάξετε ένα vi /README.$(hostname). Μια γραμμή στο αρχείο του root /bash_logout. Ένας άλλος τρόπος να το κάνετε αυτό είναι να γράψετε su ή ένα sudo script που κάνει κάτι σαν το παρακάτω:

```
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 -
```

(χρησιμοποιήστε την εντολή typescript για να δημιουργήστε ένα session log και δημιουργήστε μια συνάρτηση που θα προσθέτει και θα ενημερώνει αυτόματα το log).

Παραδέχομαι ότι δεν έχω υλοποιήσει αυτό την πολιτική αυτόματης ενημέρωσης – έχω βασιστεί ως τώρα στην αυτοπειθαρχία μου ως τώρα. Παρόλαυτά, έχω παίξει με αυτή την ιδέα (μέχρι το σημείο να φτιάξω ένα πρότυπο δείγμα (prototype) των scripts και των shell συναρτήσεων όπως τις βλέπετε). Ένα πράγμα που με κρατάει είναι η ίδια η 'script' εντολή. Νομίζω ότι θα πρέπει να πάρω τον κώδικα και να προσθέσω δυο παραμέτρους στη γραμμή εντολών (για να διακόπτω/σταματώ το script να καταγράφει από τη γραμμή εντολών) πριν να δεσμευτώ ότι θα το χρησιμοποιώ).

Η τελευταία μου πρόταση (γι' αυτό το γύρο):

Το path του Root θα πρέπει να περιέχει το 'PATH= /bin'

Αυτά και τίποτα άλλο στο path του root. Ο,τι κάνει ο root παρέχεται από ένα symlink από το /bin ή από ένα alias ή shell ρουτίνα, ή είναι ένα script ή ένα εκτελέσιμο στο /bin, ή γράφεται με το πλήρες path.

Αυτό κάνει τον καθένα που τρέχει σαν root να γνωρίζει (μερικές φορές επώδυνα) πόσο εμπιστεύεται τα εκτελέσιμα. Ο γνωστικός admin ενός host πολλών χρηστών θα κοιτάει περιοδικά μέσω του δικού του /bin και των αρχείων /.*history ψάχνοντας patterns και παράθυρα.

Ο πραγματικά δραστηριοποιημένος admin θα εντοπίσει ακολουθίες που μπορούν να αυτοματοποιηθούν, μέρη στα οποία έλεγχοι μπορούν/πρέπει να εισαχθούν , και διαδικασίες για τις οποίες "root" τα δικαιώματα θα πρέπει να αποφευχθούν προσωρινά (η εκτέλεση διαφόρων επεξεργαστών, MTA's και άλλα μεγάλα διαλογικά (interactive) προγράμματα με περίπλοκα χαρακτηριστικά scripting τα οποία *μπορεί* να ενσωματωθούν σε διαφανή αρχεία ή αρχεία δεδομένων – όπως το αισχρό vi ./.exrc και το emacs ./.emacs και το ακόμη περισσότερο ύπουλο $EXINIT και οι ενσωματωμένες μακροεντολές επικεφαλίδων (header/footer macros)). Φυσιολογικά οι εντολές αυτού του τύπου μπορούν να τρέξουν με κάτι σαν:

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

(...όπου οι λεπτομέρειες εξαρτώνται από την εντολή).

Αυτές οι τελευταίες κυρίως προφυλάξεις υπερβολικές για το σπίτι ή το "single" σταθμό εργασίας του χρήστη – αλλά είναι πολύ καλή τακτική για τη διαχείριση ενός συστήματος πολλαπλών χρηστών – ειδικά για ένα σύστημα δημόσια εκτεθειμένο (όπως αυτό στο netcom).

3.7 Πώς να ρυθμίσετε το xdm's chooser για επιλογή host. *Arrigo Triulzi,* `a.triulzi@ic.ac.uk`

1. Ενημερώστε το αρχείο που φορτώνει το xdm πολύ πιθανά το /etc/rc/rc.6 ή /etc/rc.local) έτσι ώστε να περιέχει τις παρακάτω γραμμές στο τμήμα εκκίνησης του xdm.

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

2. Ανοίξτε το /usr/lib/X11/xdm/Xservers και βάλτε σε σχόλια τη γραμμή που ξεκινά το server στο τοπικό μηχάνημα (π.χ. ξεκινώντας με 0:)
3. Επανεκκινήστε το μηχάνημα και you're home and away.

Αυτό το πρόσθεσα όταν προσπαθούσα απεγνωσμένα να το εγκαταστήσω για το δικό μου subnet εδώ χρειάστηκε περίπου μια βδομάδα για να κατευνάσω όλα τα προβλήματα.

Προειδοποίηση: με το παλιό SLS (1.1.1) για κάποιο λόγο μπορείτε να αφήστε ένα -nodaemon μετά τη γραμμή xdm – αυτό **ΔΕ** δουλεύει για μεταγενέστερες εκδόσεις.

3.8 Περί της Ελληνικής Μετάφρασης

Για σχόλια, παρατηρήσεις, διορθώσεις, ενημερώσεις περί της Ελληνικής μετάφρασης αυτού του HOWTO, στείλτε email στην Βούλα Σανιδά : `voulariba@hellug.gr` ή στον Ευριπίδη Παπακώστα : `evris@hellug.gr` και εναλλακτικά : `ep@rocketmail.com`