*Nat Makarévitch* mailto:nat@nataa.fr.eu.org

# Astuces Linux HOWTO

v3.6, Juin 1998 traduction du 2 janvier 1999

Ce document contient ces astuces et réglages difficiles à trouver qui rendent Linux un peu plus sympathique.

## Contents

# 1 Introduction

Ce document est le **Linux Astuces HOWTO** (titre original : **Linux Tips HOWTO**), une liste de trucs et d'optimisations bien pratiques, qui contribuent à rendre Linux plus agréable. Tout ce qui est ici sort ou bien de ma tête, ou bien de l'ancien Astuces HOWTO (après tout, pourquoi enlever des astuces qui marchent ?) Alors envoyez-moi vos astuces préférées (NdT : en anglais !) pour que je puisse les inclure dans la prochaine version du Linux Astuces HOWTO.

Paul Anderson *Rédacteur de "Linux TIPS HOWTO"*

`panderso@ebtech.net`

# 2 Astuces simples

## 2.1 Un truc pratique pour syslog. *Paul Anderson, rédacteur du Linux Astuces HOWTO.*

Editez le ficher `/etc/syslog.conf` et ajoutez-y la ligne suivante :

```
# Tout envoyer sur tty8
*.*                                     /dev/tty8
```

Attention : PENSEZ À UTILISER DES TABULATIONS ! Syslog n'aime pas les espaces.

## 2.2 Un script pour afficher les HOWTO compactés. *Didier Juges,* `dj@destin.nfds.net`.

De débutant à débutant, voici un petit script qui facilite la lectures des howto. Mes howto sont dans `/usr/doc/faq/howto/` et sont compressés avec `gzip`. Les fichiers s'appellent `XXX-HOWTO.gz`, où XXX est le titre. J'ai appelé le script suivant `howto` et je l'ai placé dans `/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
```

Appelé sans argument, il affiche la liste des howto disponibles. Quand on lui passe en argument la première partie du nom du fichier (avant le trait d'union), il décompacte le document (en laissant l'original intact) et l'affiche à l'écran.

Par exemple, pour afficher le document `Serial-HOWTO.gz`, tapez :

```
$ howto serial
```

## 2.3 Reste-t-il assez de place libre ? *Hans Zoebelein,* `zocki@goldfish.cube.net`.

Voici un script qui vérifie à intervalles réguliers qu'il reste de la place sur tout ce qui est monté (disques durs, CDROM, disquettes...)

En cas de pénurie d'espace libre, un message est affiché à l'écran toutes les X secondes et un courrier électronique est envoyé pour chaque périphérique qui déborde.

```
#!/bin/sh

#
# $Id: Tips-HOWTO.sgml,v 1.1.1.1 2003/01/03 02:38:54 traduc Exp $
#

#
# Depuis que j'ai ete confronte a des
# messages d'erreur mysterieux pendant les compilations
# quand les fichiers temporaires remplissaient mes disques, j'ai
# ecrit ca pour etre averti avant que les disques
# ne soient pleins.
#
# Si ca a empeche vos serveurs d'exploser, envoyez
# les courriers de remerciement a zocki@goldfish.cube.net.
# Si votre site flambe a cause de ca, desole
# mais je vous avais prevenu: c'est votre probleme
# Si vous savez vraiment vous servir de sed, excusez moi :)
#

#
# Lancez-le et vous pouvez l'oublier: mettez "check_hdspace &"
# dans rc.local. Il verifie l'espace libre toutes les
# $SLEEPTIME secondes. Vous pouvez meme surveiller vos
# disquettes et vos bandes.  :)
# Si l'espace libre est inferieur a $MINFREE (Ko),
# le script va afficher un message d'avertissement et envoyer un
# courrier a $MAIL_TO_ME pour chaque peripherique
# concerne. Des qu'il y a a nouveau plus de place
# libre que la limite, le systeme d'envoi de courrier est
# reamorce.
#

# RESTE A FAIRE:
# Des $MINFREE differents pour chaque peripherique
# Nettoyer les repertoires /*tmp des vieilleries en cas de
#          penurie d'espace.


DEVICES='/dev/sda2 /dev/sda8 /dev/sda9' # vos disques
MINFREE=20480                           # la limite
SLEEPTIME=10                            # secondes entre deux verifications
MAIL_TO_ME='root@localhost'             # la personne a avertir


# ------- rien a changer en dessous (j'espere :) -------

MINMB=0
ISFREE=0
MAILED=""
let MINMB=$MINFREE/1024         # oui, on fait ca bien :)

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 mb)" \
                                | mail -s "WARNING: $DEVICE only $ISMB mb free!" $MAIL_TO_ME
                                MAILEDH="$MAILED $DEVICE"
                                MAILED=$MAILEDH
                                # rajoutez ce qu'il reste a faire
                                # par exemple nettoyer les */tmp
                        fi
                        elif [ -n  "`echo $MAILED | grep -w $DEVICE`" ] ; then
                                # Enlever le marqueur de courrier si
                                # l'espace disponible remonte
                                # au-dessus de la limite. Pour pouvoir
                                # envoyer un nouveau message en cas de
                                # besoin.
                                MAILEDH="`echo $MAILED  | sed s#$DEVICE##`"
                                MAILED=$MAILEDH
                        fi

                done
                sleep $SLEEPTIME

done
```

### 2.4 Un utilitaire pour nettoyer vos fichiers journaux (logs). *Paul Anderson, rédacteur du Linux Astuces HOWTO.*

Si vous êtes comme moi, vous avez une liste de diffusion avec 430 inscrits et plus de 100 messages qui arrivent tous les jours par UUCP. Qu'est-ce qu'un bidouilleur peut bien faire avec ces énormes fichiers journaux ? Il peut installer `chklogs`. `chklogs` a été écrit par Emilio Grimaldo, `grimaldo@panama.iaehv.nl`, et la version 1.8 actuelle est disponible sur `ftp.iaehv.nl:/pub/users/grimaldo/chklogs-1.8.tar.gz`. C'est très simple à installer (il faut bien sûr lire le contenu du répertoire `doc`). Une fois le paquetage installé, rajoutez une entrée à votre crontab :

```
# Lance chklogs tous les jours a 21h
00 21 * * *         /usr/local/sbin/chklogs -m
```

Pendant que vous y êtes, n'oubliez pas de dire à l'auteur à quel point vous appréciez son logiciel :)

### 2.5 Un script pratique pour nettoyer les fichiers `core`. *Otto Hammersmith.*

Créez un fichier `rmcores` (l'auteur l'appelle `handle-cores`) contenant ceci :

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

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

 echo Deleting...
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
```

Et utilisez cron pour le lancer à intervalles réguliers.

## 2.6 Déplacement de répertoires inter partitions Linux (filesystems). *Alan Cox,* A.Cox@swansea.ac.uk.

```
(cd /repertoire_source && tar cf - . ) | (cd /repertoire_cible && tar xvf -)
```

*[ Et pas cd /répertoire_ source; tar...etc., qui laisse la possibilité de bousiller un répertoire en cas de problème. Merci à Jim Dennis, jim@starshine.org, qui me l'a signalé. -le rédacteur ]*

## 2.7 Trouver les plus gros répertoires. *Mick Ghazey.*

Vous vous êtes déjà demandé quels étaient les plus gros répertoires sur votre machine ? Voici une façon de les trouver.

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

## 2.8 La Linux Gazette.

Bravo et merci à John Fisk, le créateur de la Linux Gazette. C'est un excellent magazine en ligne, qui plus est **GRATUIT !** Que demander de plus ? Vous pouvez le trouver à l'adresse :

```
http://www.linuxgazette.com
```

Au fait, il s'avère que (1) la LG est maintenant mensuelle et (2) elle n'est plus maintenue par John Fisk, mais par l'équipe de SSC.

## 2.9 Indication permettant de résoudre le problème posé par le VPATH du GNU make version 3.7. *Ted Stern,* stern@amath.washington.edu.

J'ignore si ce problème concerne de nombreux utilisateurs mais l'une des caractéristiques de la version 3.7 du GNU make ne m'enthousiasme pas. Il s'agit du comportement d'un VPATH sur répertoire absolu. Un robuste patch corrige cela, vous pourrez l'obtenir auprès de Paul D. Smith <psmith@wellfleet.com> [1]. Ce

[1] Veuillez rédiger votre courrier en anglais ! NDT

dernier poste dans le groupe gnu.utils.bug un article contenant ce patch et sa documentation après parution de chaque nouvelle version du GNU make. En ce qui me concerne... il est installé sur tous les systèmes auxquels j'ai accès !

## 2.10 Comment interdire à ma machine de lancer fsck après chaque démarrage ? *Dale Lutz,* `dal@wimsey.com`.

Réponse : Après recompilation du noyau le système de fichiers est considéré comme non vérifié ("marked as dirty"), ce qui implique que fsck sera mis en action lors de chaque démarrage. Pour éviter cela lancer :

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

Cela modifie le noyau qui, dès lors, considère que le système de fichiers est sain.

Note : Ajoutez, si vous employez LILO, `read-only` à la section de l'image de boot de votre fichier de configuration LILO (souvent nommé /etc/lilo/config ou /etc/lilo.conf).

## 2.11 Comment éviter les lancements de fsck, au boot, dûs au "device busy" ? *Jon Tombs,* `jon@gtex02.us.es`.

Si votre système connaît de fréquentes erreurs de type "device busy" au démarrage qui laissent le système de fichiers dans un état exigeant un `fsck`, veuillez suivre les recommandations suivantes :

Ajoutez, au fichier `/etc/rc.d/init.d/halt` ou `/etc/rc.d/rc.0`, la ligne

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

pour tous vos systèmes de fichiers montés, sauf la racine, avant l'invocation de umount -a. Cela signifie que si, pour une quelconque raison, "shutdown" ne parvient pas à tuer tous les processus puis démonter les partitions ces dernières seront malgré tout considérées comme saines lors du redémarrage. Cette astuce a considérablement écourté le temps de démarrage de mon système !

## 2.12 Comment trouver les plus gros fichiers sur votre disque dur. *Simon Amor,* `simon@foobar.co.uk`.

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

Pour ceux d'entre vous qui sont vraiment à l'étroit, ça prend du temps, mais ça marche bien :

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

## 2.13 Comment imprimer sur des pages avec marges ? *Mike Dickey,* `mdickey@thorplus.lib.purdue.edu`.

```
#!/bin/sh
# /usr/local/bin/print
# Une simple sortie formatee pour permettre de
# perforer les feuilles afin de les mettre dans un classeur

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

## 2.14 Méthode permettant de rechercher des expressions rationnelles dans des fichiers. *Raul Deluth Miller,* `rockwell@nova.umd.edu`.

Je désigne, par "expressions rationnelles", les `regexp` de "grep" et consorts.

J'ai appelé ce script "forall" et l'utilise ainsi :

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

Voici le script forall:

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

## 2.15 Un script pour faire le ménage derrière les programmes qui créent des fichiers de sauvegarde.

Voici un petit script de deux lignes qui parcourt une arborescence et qui y efface les fichiers de sauvegarde (# et ~) d'emacs, les fichiers .o, et les fichiers .log de TeX. Il compacte également les fichiers .tex et README. Sur mon système, je l'ai appelé "squeeze".

```
#!/bin/sh
#SQUEEZE efface les fichiers superflus et compacte les fichiers .tex
#et README.
#Par Barry tolnas, tolnas@sun1.engr.utk.edu
#
echo nettoyage de $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 Comment trouver le processus qui occupe le plus de mémoire. *Simon Amor.*

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

-OU-

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

## 2.17 Configuration de `vi` pour la programmation en C. *Paul Anderson, rédacteur du Linux Astuces HOWTO.*

Je passe beaucoup de temps à programmer en C, et j'ai pris le temps de configurer vi pour me faciliter la tâche. Voici le contenu de mon fichier `.exrc` :

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

Qu'est-ce que ça fait ? `autoindent` force vi à indenter automatiquement toutes les lignes qui suivent la première ligne indentée, `shiftwidth` impose une taille de 4 espaces pour ˆT, `backspace` configure la touche d'espacement arrière, et `ruler` force l'affichage des numéros de lignes. Notez que pour placer le curseur sur une ligne donnée, par exemple la ligne 20, vous pouvez utiliser :

```
vi +20 monfichier.c
```

## 2.18 Utilisation de ctags pour faciliter la programmation

Beaucoup de bidouilleurs ont déjà ctags sur leur machine, mais ne s'en servent pas. Cela peut être très pratique pour éditer des fonctions spécifiques. Supposez que vous avez une fonction dans l'un des nombreux fichiers sources contenus dans un répertoire pour un programme que vous êtes en train d'écrire, et que vous voulez éditer cette fonction pour faire une mise à jour. Appelons cette fonction foo(). Vous ne savez pas non plus où elle se trouve dans le fichier source. C'est là que ctags peut être très pratique. Quand vous le lancez, ctags crée un fichier nommé `tags` dans le répertoire courant, qui contient la liste de toutes les fonctions, le fichier source dans lequel elles se trouvent et leur emplacement dans ce fichier source. Le fichier `tags` ressemble à ça :

```
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)$/
AddToList       list.c  /^AddToList(list_head, name, ptr)$/
```

Pour éditer, par exemple, AddEndResize() avec vim, tapez :

```
vim -t AddEndResize
```

Cela va ouvrir le bon fichier dans l'éditeur et placer le curseur au début de la fonction.

## 2.19 Pourquoi sendmail se bloque-t-il pendant 5 minutes au démarrage d'une Red Hat ? *Paul Anderson.*

C'est un problème assez courant, presque au point d'en faire une FAQ. Je ne sais pas si Red Hat corrige l'erreur dans sa distributions, mais vous pouvez réparer ça vous-même. Si vous regardez dans votre fichier `/etc/hosts`, vous allez trouver quelque chose qui ressemble à ça :

```
127.0.0.1       localhost       votremachine
```

Quand sendmail démarre, il fait une recherche sur le nom de votre machive (`votremachine` dans l'exemple). Ensuite, il trouve que l'adresse IP de la machine est 127.0.0.1; sendmail n'aime pas ça et recommence la recherche. Il continue comme ça pendant un moment avant d'abandonner. Corriger ce problème est très facile : éditez votre fichier `/etc/hosts` et mettez-y quelque chose comme ça :

```
127.0.0.1        localhost
10.56.142.1      votremachine
```

### 2.20 Comment configurer une Red Hat pour avoir `ls` en couleurs? *Paul Anderson,* `paul@geeky1.ebtech.net`.

La distribution Red Hat est fournie avec color-ls (ls en couleurs), mais je n'arrive pas à comprendre pourquoi ils ne le configurent pas pour utiliser les couleurs par défaut. Voici une façon d'arranger ça.

Commencez par taper `eval ‘DIRCOLORS‘`

Puis `alias ls='ls -color=auto'`

Enfin, mettez la ligne "`alias .....`" dans votre `/etc/bashrc`.

### 2.21 Comment trouver quelle bibliothèque de /usr/lib contient une fonction donnée ? *Pawel Veselow.*

Vous êtes en train de compiler un programme et vous avez oublié de lier une bibliothèque nécessaire? Et gcc qui ne donne que les noms des fonctions manquantes... Voici une commande pour trouver ce que vous cherchez :

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

Remplacez `tgetnum` par le nom de la fonction que vous cherchez.

### 2.22 J'ai compilé un petit programme en C, mais quand je le lance, je ne vois aucun résultat !

Vous avez compilé le programme et créé un programme appelé `test`, non? Linux a déjà un programme `test`, qui teste si une certaine condition est vraie et qui n'affiche aucun résultat à l'écran. Pour lancer votre programme `test`, tapez `./test`.

## 3 Astuces détaillées

### 3.1 Linux et Windows peuvent utiliser une même partition pour le swap ! *Tony Acero,* `ace3@midway.uchicago.edu`.

1. Formater la partition sous DOS puis y disposer le fichier d'échange de Windows. Ne pas employer Windows tout de suite afin de laisser ce fichier complètement "vide" pour faciliter son compactage.

2. Démarrer Linux et sauver ce fichier dans un fichier. Exemple (cas d'une partition de "swap" commun nommée /dev/hda8) :

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

3. Compacter le fichier de swap :

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

4. Ajouter au fichier /etc/rc la ligne suivante afin de préparer et installer la partition de swap lorsqu'elle est employée par Linux : *XXXXX représente ici le nombre de blocs que compte la partition de swap*

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

   Ajoutez une ligne destinée à cette partiton de swap dans le fichier /etc/fstab

5. Si les programmes init et shutdown employés utilisent /etc/brc ajouter à ce fichier les lignes suivantes :

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

   Dans le cas contraire il vous faudra invoquer ces commandes avant chaque fin de session Linux (placer ces commandes dans un script...)

Note : dd ne traite que 100 blocs car j'ai empiriquement déterminé que rien ne sert d'en écrire davantage !

>> Quels sont les avantages et inconvénients de cette méthode ?

Avantages : gain d'espace disponible sur le disque !

Inconvénients : si l'étape de restauration du fichier d'échange Windows n'est pas automatique il ne faudra pas négliger, sous Linux et avant chaque redémarrage "vers" Windows, de lancer les commandes chargées de cette remise en place.

## 3.2 Récupération de fichiers effacés. *Michael Hamilton,* `michael@actrix.gen.nz`.

Voici une astuce dont j'ai eu besoin à quelques reprises.

La récupération d'un fichier texte par une personne désespérée.

Si vous effacez un fichier texte par accident, par exemple un courrier électronique ou le produit d'une nuit de programmation, tout n'est pas perdu. Si le fichier a eu le temps d'aller jusqu'au disque, c'est à dire s'il a existé pendant plus de 30 secondes, il est possible que son contenu se trouve encore sur la partition.

Vous pouvez le rechercher dans la partition en utilisant la commande grep.

Par exemple, récemment, j'ai effacé un courrier électronique par accident. J'ai immédiatement cessé toute activité qui aurait pu modifier le contenu de la partition : je me suis abstenu de sauvegarder quoi que ce soit, de compiler quoi que ce soit, etc. En d'autres occasions, je suis allé jusqu'à passer le système en mode mono-utilisateur et démonter le système de fichiers.

J'ai ensuite utilisé la commande egrep sur la partition : dans mon cas, le message se trouvait dans `/usr/local/home/michael/`, et donc d'après la sortie de df, dans `/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
```

Je suis extrêmement prudent quand je manipule des partitions, donc j'ai bien pris le temps de m'assurer que je comprenais la syntaxe de cette commande AVANT de presser la touche Entrée. Dans ce cas, le message contenait la mot "ftp", puis un peu de texte suivi du mot "COL". Le message faisait une vingtaine de lignes, donc j'ai utilisé -50 pour avoir toutes les lignes assez proches de la phrase. Il m'est déjà arrivé d'utiliser -3000 pour être sûr de réperer toutes les lignes d'un code source. J'ai redirigé le sortie de egrep vers une autre partition pour éviter d'écraser le message que je recherchais.

J'ai ensuite utilisé la commande strings pour examiner le résultat.

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

Effectivement, le message était là.

Cette méthode peut ne pas être efficace si tout ou partie de l'espace disque a déjà été réutilisé.

Cette astuce n'est probablement utilisable que sur un système mono-utilisateur. Sur un système multi-utilisateurs avec beaucoup d'activité sur les disques, l'emplacement que vous avez libéré peut très bien déjà avoir été réutilisé. Et pour la plupart nous ne pouvons pas nous permettre d'enlever la machine de sous les pieds de nos utilisateurs dès que nous avons besoin de récupérer un fichier.

Sur mon système personnel, cette astuce a été bien pratique à environ trois occasions ces quelques dernières années - généralement après que j'ai détruit accidentellement une partie de mon travail du jour. Si ce que je fais survit assez longtemps pour progresser de façon significative, je le sauvegarde sur une disquette, donc je n'ai pas souvent besoin de ce truc.

## 3.3 Comment utiliser le marqueur d'immutabilité. *Jim Dennis,* `jadestar@rahul.net`.

Utilisez le marqueur d'immutabilité.

Juste après avoir installé et configuré votre système, faites un tour dans `/bin`, `/sbin`, `/usr/bin`, `/usr/sbin`, `/usr/lib` et autres, et n'hésitez pas à vous servir de la commande "`chattr +i`". Appliquez-la aussi aux fichiers du noyau à la racine. Maintenant, "`mkdir /etc/.dist`" et copiez-y toute l'arborescence contenue dans `/etc` (je le fais en deux étapes en utilisant `/tmp/etcdist.tar` pour éviter la récursion). (Vous pouvez aussi vous contenter de `/etc/.dist.tar.gz`). Et placez-y un marqueur d'immutabilité.

Tout cela sert à limiter les dégâts que vous pouvez faire en tant que root. Vous éviterez ainsi d'écraser des fichiers avec une redirection mal contrôlée, et vous ne risquez pas de rendre le système inutilisable à cause d'une espace mal placée dans une commande "`rm -fr`" ; vous pouvez toujours faire très mal à vos données, mais vos binaires et vos bibliothèques seront à l'abri.

De plus, cela prévient, ou du moins complique, l'exploitation d'un certain nombre de trous de sécurité ; en effet, beaucoup d'attaques de ce type écrasent un fichier au moyen d'un quelconque programme SUID qui *ne permet pas d'exécuter une commande arbitraire.*

Le seul inconvénient se présente à l'installation de divers logiciels système. D'un autre côté, ça empêche l'écrasement accidentel de fichiers par "`make install`". Si vous oubliez de lire le Makefile et d'appliquer `chattr -i` aux fichiers qui doivent être écrasés (et aux répertoires auxquels vous voulez ajouter des fichiers), le make échoue, et il suffit d'utiliser chattr avant de le relancer. Vous pouvez aussi en profiter pour déplacer vos anciens binaires, bibliothèques et autres dans un répertoire `.old/`, les renommer, les archiver ou autre.

## 3.4 Une suggestion quant à l'endroit où mettre ce que vous rajoutez.

Tout ce que vous rajoutez doit se trouver sous `/usr/local` ou `/usr/local/`hostname``!

Si votre distribution laisse /usr/local vide, créez /usr/local/src, /usr/local/bin, etc. et utilisez-les. Si votre distribution met des choses dans /usr/local, créez /usr/local/‘hostname‘ et donnez-lui le mode +w pour le groupe wheel (en plus, je le rends SUID et SGID pour m'assurer que les membres du groupe wheel ne peuvent toucher qu'à leurs propres fichiers et que tous les nouveaux fichiers vont appartenir au groupe wheel).

Maintenant, forcez-vous à *TOUJOURS* placer les nouveaux paquetages sous /usr/local/src/.from/$OU_JE_L_AI_EU (pour les fichiers .tar ou autres) et à les compiler sous /usr/local/src (ou .../$HOSTNAME/src). Assurez-vous qu'ils s'installent sous la hiérarchie locale. Si quelque chose *doit obligatoirement* être installé dans /bin ou /usr/bin ou autre, créez un lien symbolique depuis la hiérarchie locale vers tout ce qui est installé ailleurs.

La raison de tout ça, même si ça représente plus de travail, est que ça permet de trouver facilement ce qui doit être sauvegardé et réinstallé en cas de réinstallation complète depuis le média de distribution (habituellement un CD à l'heure actuelle). En utilisant un répertoire /usr/local/src/.from, vous gardez aussi une trace de la provenance de vos sources, ce qui est utile pour trouver les mises à jour et qui peut s'avérer critique pour suivre les listes d'annonces de sécurité.

Un de mes systèmes personnels (celui que j'utilise) a été monté avant que je n'applique moi-même cette politique. Je ne "sais" toujours pas en quoi il diffère du système de base "tel qu'installé". Et cela bien que je n'ai changé que très peu de choses quant à sa configuration et que je suis le *seul* à l'utiliser.

A contrario, tous les systèmes que j'ai mis en place au travail (où j'ai été bombardé administrateur système) ont été configurés de cette façon. Ils ont été administrés par plusieurs personnes extérieures et autres membres du département informatique, et ils ont subi de nombreuses mises à jour et installations de logiciels. Pourtant, j'ai une idée très précise de ce qui a été rajouté *après* l'installation et la configuration initiales.

## 3.5 Conversion de tous les fichiers d'un répertoire en minuscules. *Justin Dossey,* dossey@ou.edu.

J'ai remarqué quelques procédures difficiles ou superflues recommandées dans les trucs et astuces du numéro 12 [2]. Comme il y en a plusieurs, je vous adresse ce message.

```
#!/bin/sh
         # lowerit
         # convertit les noms de tous les fichiers du repertoire
         # courant en minuscules
         # n'affecte que les fichiers, pas les sous-repertoires
         # demande confirmation avant d'ecraser un fichier existant
         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
```

Voilà un long script. Je n'écrirais pas un script pour ça ; j'utiliserais plutôt la commande suivante :

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

[2]NdT : Apparemment, cette section est tirée de la Linux Gazette

Ce contributeur dit qu'il a écrit le script de cette façon pour des raisons de lisibilité (voir plus bas).

Pour l'astuce suivante, qui traite de l'ajout et de la suppression d'utilisateurs, Geoff s'en sort bien jusqu'à la dernière étape. Rebooter ? J'espère qu'il ne reboote pas à chaque fois qu'il supprime un utilisateur. Les deux premières étapes suffisent. De toutes façons, quels processus cet utilisateur pourrait-il laisser tourner ? Un bot IRC ? Tuez simplement les processus avec :

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

Par exemple, pour l'utilisateur foo:

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

Cette question étant classée, passons au mot de passe de root oublié.

La solution donnée dans la Gazette est la plus universelle, mais pas la plus facile. Aussi bien avec LILO qu'avec Loadlin, le paramètre "single" permet de lancer directement le shell par défaut au démarrage, sans entrer de login ni de mot de passe. À partir de là, il suffit de changer ou d'enlever le mot de passe problématique, avant de taper "`init 3`" pour passer en mode multi-utilisateurs. De cette façon, un seul reboot ; de l'autre, deux reboots.

Justin Dossey.

## 3.6 Mise à jour de Sendmail. *Paul Anderson,* paul@geeky1.ebtech.net

Nous partons d'une source propre. Commencez par vous procurer le code source de sendmail. J'ai téléchargé la version 8.9.0, qui est comme vous pouvez le voir à la pointe du progrès. Je l'ai récupérée depuis ftp.sendmail.org:/pub/sendmail/sendmail-8.9.0.tar.gz

Il pèse à peu près un méga-octet, et sachant que j'utilise la version 8.7.6, je crois que ça vaut le coût. Si ça marche, vous en entendrez sûrement parler ; sinon, je n'aurai plus de courrier et je ne pourrai pas distribuer la nouvelle version de ce HOWTO :)

Maintenant que vous avez téléchargé le source, décompactez-le. Cela va créer un sous-répertoire `sendmail-8.9.0` dans le répertoire courant. Placez-vous dans ce sous-répertoire et lisez les fichiers `README` et `RELEASE_NOTES` (et soyez époustouflé par toutes les améliorations qu'ils ont apportées). Maintenant, placez-vous dans `src`. C'est là que vous allez faire le plus gros du travail.

*Une remarque au passage : Sendmail est un programme petit, puissant et bien écrit. Le binaire `sendmail` lui-même a mis moins de 5 minutes à compiler sur mon 5x86 133 avec 32 Mo de RAM ! La totalité de la compilation et de l'installation (sans compter la configuration) ont pris moins de 15 minutes !*

Je n'utilise pas BIND sur mon système, donc j'ai cherché les lignes suivantes :

```
# ifndef NAMED_BIND
#  define NAMED_BIND    1       /* use Berkeley Internet Domain Server */
# endif
```

et j'ai remplacé le 1 par un 0:

```
# ifndef NAMED_BIND
#  define NAMED_BIND    0       /* use Berkeley Internet Domain Server */
# endif
```

Sur la Debian 1.3, `db.h` est installé par défaut dans `/usr/include/db`, au lieu de `/usr/include` où sendmail espère le trouver. Placez-vous successivement dans les sous-répertoires `src`, `mailstats`, `makemap`, `praliases`, `rmail` et `smrsh` et éxecutez la commande suivante :

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

Ensuite, `cd ..` et tapez `make install`. Voilà ! La version 8.9.0 de Sendmail doit maintenant être installée ! Bien sûr, ça suppose que vous avez déjà votre configuration d'origine. Pour que tout marche bien sur mon système, comme j'héberge des listes de diffusion gratuites utilisant majordomo, j'ai ajouté la ligne suivante au début de mon `/etc/sendmail.cf` :

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

Sendmail 8.9.0 est à l'heure actuelle plutôt bavard à propos des permissions des répertoires et des fichiers, et il va se plaindre à propos des répertoires et des fichiers qui autorisent l'accès en écriture pour le groupe ou pour tout le monde parmi les fichiers d'alias ou `.forward`. Bien qu'il ne soit pas recommandé d'inhiber ces avertissements, je suis toujours seul à la console et j'ai trouvé que ce trou de sécurité mineur n'était en fait pas gênant. C'est vous qui voyez.

## 3.7 Quelques astuces pour les administrateurs système débutants. *Jim Dennis,* `jadestar@rahul.net`

Créez et tenez à jour un fichier `/README.‘hostname‘` ou `/etc/README.‘hostname‘` *[ ou éventuellement /usr/local/etc/README.‘hostname‘ - le rédacteur ]*

Absolument, à compter du *premier jour* de l'administration d'un système, prenez des notes dans un fichier journal. Vous pouvez mettre "vi /README.$(hostname)" sur une ligne du fichier `.bash_logout` de root. Une autre façon de faire est d'écrire un script `su` ou `sudo` qui fait quelque chose comme ça :

```
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 -
```

(utilise la commande tapée pour créer une trace de la session et crée une fonction pour automatiser la mise à jour du fichier journal).

J'admets que je n'ai pas implanté cette automatisation - jusqu'à maintenant je me suis reposé sur ma discipline. Cependant j'ai envisagé l'idée (au point d'écrire les scripts et les fonctions que vous avez sous les yeux). Une chose qui me retient est la commande "script" elle-même. Je pense qu'il va falloir que je me procure les sources et que je rajoute une paire de paramètres (pour arrêter l'enregistrement du script depuis la ligne de commandes) avant de me mettre à utiliser ça.

Ma dernière suggestion (pour cette fois) :

La variable PATH de root devrait contenir `PATH=~/bin`.

C'est tout. Rien d'autre dans le PATH de root. Tout ce que root peut faire est fourni par un lien symbolique dans `~/bin`, un alias, une fonction shell, un script ou un binaire situé dans `~/bin`, ou bien la commande est tapée avec un chemin d'accès explicite.

De cette façon, toute personne utilisant le compte root se rend compte (parfois douloureusement) à quel point elle fait confiance aux binaires. L'administrateur avisé d'un système multi-utilisateurs va en plus parcourir régulièrement son répertoire `~/bin` et ses fichiers `~/.*history` pour y chercher des répétitions et des moyens de les contourner.

L'administrateur vraiment motivé va repérer les enchaînements qui peuvent être automatisés, les endroits où des vérifications peuvent être ajoutées, et les tâches pour lesquelles les privilèges de root peuvent être abandonnées (comme lancer un éditeur, un agent de transport de courrier électronique ou autre gros programme pouvant exécuter des scripts qui *pourraient* être inclus dans des fichiers de données - comme vi (`./.exrc`) ou emacs (`./.emacs`) ou même, plus insidieux, $EXINIT et les macros contenues au début ou à la fin des documents). Bien sûr, les commandes de ce genre peuvent être lancées avec quelque chose comme ça :

```
cp $donnees $repertoire_utilisateur/tmp
su -c $commande_d_origine $parametres
cp $repertoire_utilisateur/tmp $donnees
```

(... où les détails dépendent de la commande).

Ces dernières précautions sont pour la plupart superflues pour la machine personnelle ou la station "mono-utilisateur". Mais elles représentent une très bonne manière d'administrer un gros système multi-utilisateurs, particulièrement dans le cas d'un accès public (comme les machines de netcom).

## 3.8 Comment configurer `xdm` pour qu'il permette de choisir le système hôte ? *Arrigo Triulzi,* `a.triulzi@ic.ac.uk`.

1. Modifier le fichier lançant xdm lors du démarrage (probablement nommé /etc/rc/rc.6 ou /etc/rc.local) de façon que la section de xdm contienne :

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

2. Modifier le fichier /usr/lib/X11/xdm/Xservers et commenter la ligne invoquant le serveur sur la machine locale (commence par "0:")

3. Relancer le système... tout doit fonctionner !

J'ajoute cette section après avoir sué une semaine durant sur ce problème !

Attention : certaines anciennes versions de la distribution SLS (1.1.1) exigent qu'un paramètre "-nodaemon" accompagne l'invocation d'xdm. Les version ultérieures ne présentent **PAS** cette caractéristique.