

# Scripts bash

## Supprimer des fichiers en série

```
#!/bin/bash
cd /var/mail/virtual/map3.net/save/new
P=0
T=`ls|wc -l`
echo "$T fichiers total"
Q=0
for i in *;
do
    let "Q += 1"
    let "R = $Q % 1000"
    if [ "$R" -eq 0 ]
    then
        echo "$Q/$T scannes, $P supprime"
    fi

    if grep -F "htdocs/egroupware" $i > /dev/null;
    then
        rm $i;
        let "P += 1"
    fi
done
```

## Supprimer la pièce jointe d'un email et le réexpédier

```
#On regarde d'abord à quelles lignes se trouvent les limites de contenu de
lemail
#+ ces lignes commencent par --boundary

fichier="fax-email-test"

#on récupère le numéro de ligne du premier boundary
ligne1=`grep -m2 -n -e'--boundary' $fichier | sed -n '1 p' | awk --field-
separator=":" '{print $1}'`

#Ensuite on récupère le numéro de ligne du deuxième boundary
ligne2=`grep -m2 -n -e'--boundary' $fichier | sed -n '2 p' | awk --field-
separator=":" '{print $1}'`

#On affiche les lignes de lemail entre le premier et le deuxième boundary

let ligne1=ligne1+1
```

```
let ligne2=ligne2-1
sed -n ' '"$ligne1"' ,'"$ligne2"' p' <$fichier
```

## Traitement d'un fichier texte

```
#Deuxième partie traitement d'un fichier
#Préférer l'utilisation du code retour du script
#pour tester la fin de fichier, ça permet d'avoir des lignes
#vides dans le fichier

cat donnees | while true; do    # on dump le fichier
read ligne
    if [ "$ligne" == "" ] ;then
        exit 0                #On a atteint la fin de fichier
                                #Attention il ne faut pas de lignes vides
                                #dans le fichier
    fi

set -- $ligne                  #On split les valeurs de la ligne lue
                                #dans des variables $1, $2...
                                #voir man bash pour la commande set
                                #il ne reste plus qu'à afficher
    if [ $2 -ge 10 ]; then     #si supérieur ou égal on affiche
        echo -e "$1 \t $2"    #-e pour avoir le caractère de tabulation
    fi
done
```

## Chiffrer un fichier texte

```
#!/bin/bash
#Script d'ouverture du fichier de mots de passe

gpg motsdepasse.txt.gpg
nano motsdepasse.txt
gpg -er user@domaine motsdepasse.txt
rm motsdepasse.txt
```

## Sauvegardes incrémentales avec rsync

```
#!/bin/bash
#Script de sauvegardes incrémentales
#paramètres obligatoires : 1- fréquence (daily|weekly|monthly),
#                           2 - nombre de sauvegardes à conserver
##
#Auteur : Thierry Kauffmann
```

```
#
##
#   This program is free software; you can redistribute it and/or modify
#   it under the terms of the GNU General Public License as published by
#   the Free Software Foundation; either version 2 of the License, or
#   (at your option) any later version.
#
#   This program is distributed in the hope that it will be useful,
#   but WITHOUT ANY WARRANTY; without even the implied warranty of
#   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
#   GNU General Public License for more details.
#
#   You should have received a copy of the GNU General Public License
#   along with this program; if not, write to the Free Software
#   Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301
USA
##

case $1 in

    "daily" | "weekly" | "monthly" )
        freq=$1
        ;;

    *)
        echo "usage $0 daily|weekly|monthly nombre_de_sauvegardes"
        exit 1;
        ;;

esac

#Nombre de sauvegardes
glisse=$2

#Liste des répertoires à sauvegarder sur votre système séparés par des
espaces
source="/home /var"

# Vous devez créer des répertoires de sauvegarde du type
#> daily-backup, weekly-backup, monthly-backup

cible=/mnt/backup/${freq}-backup

cd $cible

#On calcule le nombre de répertoires de sauvegarde existants
nsauv=`ls | wc -w`

# S'il y a suffisamment de sauvegardes,
#> On supprime le répertoire le plus ancien
```

```
if [ "$nsauv" -ge "$glisse" ]; then

oldest=`ls -tr --time=ctime | head -1`
rm -rf $oldest

fi

#On note le répertoire de sauvegarde le plus récent
recent=`ls -t --time=ctime | head -1`

# On crée un nouveau répertoire de sauvegarde
new="$freq-backup-`date +%Y-%m-%d-%R:%S`"

#On copie en hard-links le plus récent vers le nouveau
cp -al $recent $new

#On fait la sauvegarde incrémentale
rsync -avP --delete $source $cible/$new/
```

## Compter le nombre de fichiers dans un dossier

```
#!/bin/bash
IFS=' '
ls -al | while read n; do echo `find $n | wc -l` $n; done | sort -n
```

## Synchroniser 2 fichiers de type "block-device"

C'est la seule chose que rsync ne sache pas faire ...

```
#!/bin/bash
#Incremental distant backup of block devices
#Adapted from : Stephane Chazelas stephane.chazelas at seebyte.com ;
https://lists.samba.org/archive/rsync/2010-June/025164.html

[[ -z "$1" ]] && { echo "Required first argument is source dev"; exit 1; }
[[ -z "$2" ]] && { echo "Required second argument is remote dev"; exit 1; }

dev1=$1
dev2=$2
remote=backup@backup

ssh $remote "perl -'MDigest::MD5 md5' -ne 'BEGIN{\\$/=\\1024};print md5(\\$_)'
$dev2 | lzop -c" | \
    lzop -dc | perl -'MDigest::MD5 md5' -ne 'BEGIN{\\$/=\\1024};$b=md5(\\$_);
read STDIN,$a,16;if ($a eq $b) {print "s"} else {print "c" . \\$_}' $dev1 |
```

```

lzop -c | \
  pv | \
    ssh $remote "lzop -dc | perl -ne 'BEGIN{$/=\1} if (\\$_ eq\\\"s\\\") {\\$s++}
else {if (\\$s) { seek STDOUT,\\$s*1024,1; \\$s=0}; read ARGV,\\$buf,1024; print
\\$buf}' 1<> $dev2"

#Local version
#perl -'MDigest::MD5 md5' -ne 'BEGIN{$/=\1024};print md5(\\$_)' $dev2 | perl -
'MDigest::MD5 md5' -ne 'BEGIN{$/=\1024};$b=md5(\\$_); read STDIN,$a,16;if ($a
eq $b) {print "s"} else {print "c" . \\$_}' $dev1 | perl -ne 'BEGIN{$/=\1} if
(\\$_ eq"s") {\\$s++} else {if (\\$s) { seek STDOUT,\\$s*1024,1; \\$s=0}; read
ARGV,$buf,1024; print $buf}' 1<> $dev2

```

## Modifier la valeur d'une variable dans un fichier exécutable (ELF)

Installer les paquets nécessaires

```
apt install binutils hexedit
```

Optionnel : lister les sections et les variables

```
readelf -a my_executable_elf_file | less
```

Décompiler le fichier et rechercher la commande d'affectation de la variable

```
objdump -M intel -d my_executable_elf_file | less
#/ma_variable
```

Noter la chaîne hexadécimale correspondante

Éditer le fichier en recherchant la chaîne

```
hexedit my_executable_elf_file
#CTRL+S hexstring
#CTRL+X save and exit
```

## Décaler uid et gid d'un fichier (shift uid and gid)

```
#!/bin/bash

#dir=/var/lib/lxc/gestion/rootfs/
dir=$1
range=$2

declare -i uid
```

```

declare -i gid
declare -i newuid
declare -i newgid

while IFS= read -r -d $'\0' f
do
    #echo $f
    read uid gid << (ls -dn "$f" | awk '{print $3,$4}')
    #echo "$uid:$gid"
    newuid=$(( uid + range ))
    newgid=$(( gid + range ))
    #echo "$newuid:$newgid"
    (( $uid < $range )) && { echo "$f" $newuid:$newgid; chown -h
$newuid:$newgid "$f"; }
    #ls -dn $f
done << (find $dir -print0)

```

## Réencoder en utf8 un fichier texte mal encodé

Lien : <http://www.il8nqa.com/debug/utf8-debug.html>

```

sed -e 's/Ã¢/â/g' -e 's/Ã/à/g' -e 's/Ãª/ê/g' -e 's/Ã«/ë/g' -e 's/Ã”/è/g'
-e 's/Ã/é/g' -e 's/Ã/ï/g' -e 's/Ã´/ô/g' -e 's/Ã»/û/g' -e 's/Ã¼/ü/g' -e
's/Ã§/ç/g' mon_fichier

```

## Activer un certificat SSL pour l'interface d'administration de ISPconfig3

cf :

<https://www.howtoforge.com/community/threads/get-ssl-certificate-for-ispconfig-admin-from-letsencrypt.73097/>

```

/root/.local/share/letsencrypt/bin/letsencrypt auth --text --agree-tos --
authenticator webroot --server https://acme-
v01.api.letsencrypt.org/directory --rsa-key-size 4096 --email
postmaster@`hostname` -d `--domains` `hostname` -f `--webroot-path
/usr/local/ispconfig/interface/acme

dt=`date '+%Y%m%d%H%M%S'`
cd /usr/local/ispconfig/interface/ssl/
for ext in csr key.secure key crt; do if [ -f ispserver.$ext ]; then mv
ispserver.$ext ispserver.$ext.old.$dt; fi; done

ln -s /etc/letsencrypt/live/`hostname` -f /privkey.pem ispserver.key
ln -s /etc/letsencrypt/live/`hostname` -f /fullchain.pem ispserver.crt

```

```
service apache2 restart
```

## Ajouter le support de php 5.6 dans une debian

Lien

- <https://deb.sury.org/#debian-dpa>
- <https://packages.sury.org/php/>

```
apt install apt-transport-https lsb-release ca-certificates
wget -O /etc/apt/trusted.gpg.d/php.gpg https://packages.sury.org/php/apt.gpg
echo "deb https://packages.sury.org/php/ $(lsb_release -sc) main" >
/etc/apt/sources.list.d/php.list
apt update
apt install php5.6 php5.6-curl php5.6-xml php5.6-mysql php5.6-mbstring
php5.6-mcrypt php5.6-gd php5.6-json php5.6-intl php5.6-fpm php5.6-cli
```

## Redimensionner un disque virtuel et le faire rescanner par le système

<https://geekpeek.net/rescan-scsi-bus-on-linux-system/>

```
echo "1" > /sys/class/block/sdX/device/rescan
resize2fs /dev/sdX
```

Avec LVM, en plus

```
pvresize /dev/sdX
vgresize <volume>
lvresize -l +100%FREE <logical-volume>
```

From:

<https://wiki.pielo.net/> - **Pielo.net - Wiki**

Permanent link:

<https://wiki.pielo.net/scripts-bash?rev=1537352150>

Last update: **2018/09/19 12:15**

