debian, linux, scripts, todo

# Audio

## Jouer un fichier audio en fin de boot

```
# vi /etc/rc.local
```

Ajouter ce qui suit avant la ligne 'exit 0':

```
# Jouer un fichier wav en fin de boot
aplay /chemin/vers/le/fichier/fichier.wav
```

## Jouer un fichier audio à l'arrêt du système

```
# vi /chemin/des/scripts/chord.sh
```

chord.sh

```
#! /bin/sh
### BEGIN INIT INFO
# Provides:          chord
# Required-Start:
# Required-Stop:
# Default-Start:
# Default-Stop:      0 6
# Short-Description: Jouer Chord.wav à l'arrêt.
# Description:
### END INIT INFO

aplay /chemin/Chord.wav
```

Pour utiliser les procédés normaux de lancement automatique d'un script, il doit impérativement être dans `/etc/init.d` donc soit on le place là, soit on crée un lien symbolique :

```
# ln -s /chemin/vers/script/chord.sh /etc/init.d/chord.sh
```

Puis on indique à Debian de gérer ce script en fonction des paramètres qu'il contient (section `INIT INFO`) :

```
# update-rc.d chord.sh defaults
```

# Mises à jour

```
# vi /chemin/des/scripts/maj.sh
```

maj.sh

```
#!/bin/bash
#

if [ "$(id -u)" != "0" ]; then
echo "Faut être 'root' pour lancer ce script, patate!" 1>&2
exit 1
fi

echo "***** Mise à jour de la liste des paquets dispos *****"
aptitude update
sleep 3
echo "***** Mise à jour de la liste des fichiers dans les paquets
dispos *****"
apt-file update
sleep 3
echo "***** Mise à jour des paquets installés ****"
aptitude safe-upgrade
sleep 3
echo "***** Mise à jour complète (kernel, paquets installés et
désinstallation des dépendances obsolètes) ****"
aptitude full-upgrade
sleep 3
date "+%Y-%m-%d %H:%M : 0(màj faite)" >> /var/log/update_aptitude.log
echo "***** MISE A JOUR TERMINÉE *****"
echo ""
echo "***** Suppression des paquets .deb dans le cache *****"
aptitude clean
sleep 3
#echo "***** Purge des dépendances obsolètes *****"
#aptitude purge --purge-unused
echo "***** Liste des paquets orphelins *****"
deborphan
#echo "***** Suppression des paquets orphelins *****"
#aptitude remove --purge `deborphan`
echo "***** Suppression des résidus de logiciels supprimés *****"
aptitude purge `dpkg -l | grep "^rc" | tr -s ' ' | cut -d ' ' -f 2`
sleep 3
echo "***** Liste des noyaux détectés et leurs en-têtes *****"
dpkg --list | grep linux-image
dpkg --list | grep linux-headers
echo "** Pour supprimer des éléments obsolètes, utiliser par exemple:"
echo "aptitude remove linux-headers-2.6.20-15 linux-headers-2.6.20-15-
generic linux-image-2.6.20-15-generic --purge"
```

```
echo "** IL EST RECOMMANDÉ DE CONSERVER 2 VERSIONS DU NOYAU **"
echo "***** NETTOYAGE TERMINÉ *****"
```

# Sauvegardes

## MySQL / MariaDB



**Excellente nouvelle** : ce script fonctionne aussi bien avec `MySQL` qu'avec `MariaDB` (en tout cas sur un système mis à jour de Jessie à Stretch) à part qu'il a fallu augmenter les privilèges de l'utilisateur *sqldumper* pour lui ajouter SUPER.

Le but de ce script est de pouvoir automatiser via *crontab* des sauvegardes SQL compressées quotidiennes, hebdomadaires, mensuelles et annuelles.

- Les sauvegardes quotidiennes sont purgées au delà de 5 jours
- Les sauvegardes hebdomadaires sont purgées au delà de 31 jours
- Les sauvegardes mensuelles sont purgées au delà de 365 jours
- Les sauvegardes annuelles ne sont pas purgées automatiquement

La syntaxe est visible dans le script et expliquée par un message d'aide en cas d'erreur mais voici un exemple : <cli>root@muffin:~# /mnt/muffin/scripts/sqlbackup.sh quotidien

## 2018-01-23 06:43

Sauvegarde de l'ensemble des bases dans un fichier unique...

Sauvegarde des bases Kodi les plus récentes : kodi_music60... kodi_video107... Sauvegarde de la base mysql... Sauvegarde de la base mediatomb... Sauvegarde de la base httpauth... Sauvegarde de la base piwigo... Sauvegarde de la base www...

Suppression des sauvegardes quotidiennes vieilles de plus de 5 jours... Suppression des sauvegardes hebdomadaires vieilles de plus de 31 jours... Suppression des sauvegardes mensuelles vieilles de plus de 365 jours...</cli>



Que l'on utilise des bases de données ~~*MySQL*~~*MariaDB* ou *SQLite*, la sauvegarde des base *Kodi* est très spécifique : au fil des versions du logiciel, de nouvelles bases sont parfois créées avec un numéro d'identification qui augmente au fur



et à mesure mais pas toujours par incrément de 1 (par exemple *kodi_music56* puis *kodi_music60*) et il n'est donc pas totalement simple de créer un script de sauvegarde qui ne soit pas sensible à ces changements.

```
# vi /chemin/des/scripts/sqlbackup.sh
```

sqlbackup.sh

```
#!/bin/bash
#

if [ "$(id -u)" != "0" ]; then
echo "Faut être 'root' pour lancer ce script, patate!" 1>&2
exit 1
fi

## Vérif du paramètre fournit
case $1 in
  "quotidien" | "hebdomadaire" | "mensuel" | "annuel" )
    REP="/mnt/muffin/backups/MUFFIN/sqldumps/"$1;;
  * )
    echo
    echo "SYNTAXE: sqlbackup.sh [periode]"
    echo "    [periode] doit être "quotidien", "hebdomadaire",
"mensuel" ou "annuel"."
    echo
    exit 100;;
esac

UTIL="utilisateur_sqldump"
MDP="mdp_utilisateur_sqldump"

## Sauvegarde compressée des bases de donnees en fichiers .gz
echo ===== `date +%Y-%m-%d` `date +%H:%M` =====
echo "Sauvegarde de l'ensemble des bases dans un fichier unique..."
mysqldump -u$UTIL -p$MDP --lock-all-tables --all-databases --events |
gzip > $REP/all_databases_`date +%Y-%m-%d`.gz
echo ========================================
#echo "Sauvegarde de la base information_schema..."
mysqldump -u$UTIL -p$MDP --lock-all-tables information_schema | gzip >
$REP/information_schema_`date +%Y-%m-%d`.gz
echo "Sauvegarde des bases Kodi les plus récentes :"
# Déterminer la base kodi_music la plus récente
MUSICDB=$(mysql -u$UTIL -p$MDP -e "Show databases;" | grep
"kodi_music");
echo "$MUSICDB..."
mysqldump -u$UTIL -p$MDP --lock-all-tables $MUSICDB | gzip >
```

```
$REP/${MUSICDB}_`date +%Y-%m-%d`.gz
# Déterminer la base kodi_video la plus récente
VIDEODB=$(mysql -u$UTIL -p$MDP -e "Show databases;" | grep
"kodi_video")
echo "$VIDEODB..."
mysqldump -u$UTIL -p$MDP --lock-all-tables $VIDEODB | gzip >
$REP/${VIDEODB}_`date +%Y-%m-%d`.gz
echo "Sauvegarde de la base mysql..."
mysqldump -u$UTIL -p$MDP --lock-all-tables --events  mysql | gzip >
$REP/mysql_`date +%Y-%m-%d`.gz
echo "Sauvegarde de la base piwigo..."
mysqldump -u$UTIL -p$MDP --lock-all-tables piwigo | gzip >
$REP/piwigo_`date +%Y-%m-%d`.gz
echo ========================================
#echo "===== Suppression des sauvegardes obsolètes ====="
echo "Suppression des sauvegardes quotidiennes vieilles de plus de 5
jours..."
find /mnt/muffin/backups/MUFFIN/sqldumps/quotidien/ -type f -mtime +4 |
xargs -r rm
echo "Suppression des sauvegardes hebdomadaires vieilles de plus de 31
jours..."
find /mnt/muffin/backups/MUFFIN/sqldumps/hebdomadaire/ -type f -mtime
+30 | xargs -r rm
echo "Suppression des sauvegardes mensuelles vieilles de plus de 365
jours..."
find /mnt/muffin/backups/MUFFIN/sqldumps/mensuel/ -type f -mtime +364 |
xargs -r rm
## Pas de purge des sauvegardes annuelles
```

From:
https://wiki.geekitude.fr/ - **Geekitude**

Permanent link:
**https://wiki.geekitude.fr/archives/info/os/debian/scripts?rev=1715926985**

Last update: **2024/05/17 08:23**