

[debian](#), [serveur](#), [téléchargement](#), [téléversement](#), [ftp](#), [synchro](#)

LFTP est réputé comme **LE** client ftp “simple” de Linux. Mais “simple” est ici à prendre au sens *linuxien* du terme : robuste, fiable et sans fioriture (comprenez “sans interface”). N'en déplaie aux puristes, je trouve qu'une interface web façon [Transmission](#) ne lui ferait pas de mal 😬.

Assez râlé, c'est parti :

```
# aptitude install lftp
```

Jusque là, tout va bien.

Le principe de base

LFTP est un *démon* : dès l'installation, on peut s'apercevoir qu'un processus “tourne” en mémoire.
<cli>root@muffin:~# ps aux | grep “lftp” | wc -l 1</cli>

Si l'on souhaite simplement récupérer un fichier sur un serveur ftp ou au contraire envoyer un fichier, on ajoute l'opération à la file d'attente du démon. Je n'ai pas encore essayé mais cela semble assez simple de revenir par la suite dans LFTP vérifier où en est la file d'attente, mettre en pause, etc.

Là où ça devient “velu”, c'est pour synchroniser toute une arborescence (dans un sens ou dans l'autre), surtout si on veut faire les choses bien et ne lancer cette synchro que de nuit (une fois encore, les développeurs devraient aller voir ce qui se fait ailleurs)...

Pour faire une synchronisation, on se connecte là encore au démon (avec la commande éponyme), on lance l'opération puis on quitte : <cli> root@muffin:~# lftp lftp :~> open [ftp.totos.fr](#) lftp :~> user <toto> <password_de_toto> lftp :~> mirror -v -c -use-pget-n=10 -parallel=2 /mnt/dossier/local lftp :~> exit </cli>

On s'est donc connecté au serveur [ftp.totos.fr](#) avec le login *toto* et son super mot de passe pour synchroniser tout le serveur distant (“/”) vers un chemin local en téléchargeant simultanément 2 fichiers chacun fractionné en 10 pour optimiser la bande passante.

Le problème c'est de retrouver l'opération en cours, de savoir où elle en est ou de l'arrêter (voire, avec certains scripts vus sur la toile, le risque de la lancer plusieurs fois simultanément **et de saturer la connexion Internet**).

Garder le contrôle

Heureusement, il est tout de même possible de faire un script efficace, en voici un excellent trouvé [sur ce site](#), mais sa mise en place est présentée d'une façon que j'ai eu du mal à appréhender.

Le fichier de commande

Il faut commencer par préparer un fichier dans lequel on place l'ensemble des commandes LFTP nécessaires à l'opération (vous pouvez choisir l'extension que vous voulez), pour reprendre l'exemple précédent :

[ftptoto.txt](#)

```
open ftp.totos.fr
user <toto> <password_de_toto>
mirror -v -c --use-pget-n=10 --parallel=2 / /mnt/dossier/local"
```

Le script

Viens ensuite le script qui se charge de lancer l'opération, de surveiller son état ou de l'arrêter :

[ftpsync.sh](#)

```
#!/bin/sh

# lftp synchronization script by Matthieu Bouthors
# http://www.bouthors.fr

set -e

# NAME can be anything you like as long as it is a valid filename
NAME=lftpsync
PIDFILE=/var/run/$NAME.pid
#Path to lftp daemon command
DAEMON=/usr/bin/lftp
#This is the command to be run, give the full pathname
DAEMON_OPTS="-f /chemin/vers/ftptoto.txt"

export PATH="${PATH:+$PATH:}/usr/sbin:/sbin"

# check if lftp is running

RUNNING=false
if [ -e $PIDFILE ]
then
    currentpid=`cat $PIDFILE`
    if `ps -p $currentpid > /dev/null`
    then
        RUNNING=true
    else
        rm -f $PIDFILE
    fi
fi
```

```
    fi
fi

case "$1" in
  start)
    date
    if $RUNNING
    then
        started=`date -r $PIDFILE`
        echo "daemon $NAME already started since $started, you
need to stop it first"
    else
        echo -n "Starting daemon: "$NAME
        start-stop-daemon --start --quiet --pidfile $PIDFILE --
make-pidfile --background --exec $DAEMON -- $DAEMON_OPTS
        echo "."
    fi
    ;;
  stop)
    date
    echo -n "Stopping daemon: "$NAME
    start-stop-daemon --stop --quiet --oknodo --pidfile $PIDFILE
    rm -f $PIDFILE
    echo "."
    ;;
  restart)
    date
        echo -n "Restarting daemon: "$NAME
    if $RUNNING
    then
        start-stop-daemon --stop --quiet --oknodo --retry 30 --
pidfile $PIDFILE
        rm -f $PIDFILE
    fi
    start-stop-daemon --start --quiet --pidfile $PIDFILE --make-
pidfile --background --exec $DAEMON -- $DAEMON_OPTS
    echo "."
    ;;
  check)
    if $RUNNING
    then
        echo "$NAME is running"
    else
        echo "$NAME is not running"
    fi
    ;;
*)
    echo "Usage: "$1" {start|stop|restart|check}"
    exit 1
esac
```

```
exit 0
```

Quand je disais que maîtriser LFTP n'est pas si simple, hein ?



Une fois ce script rendu exécutable, il ne reste qu'à le lancer avec l'un des paramètres possibles (start, stop, restart ou check).

Automatisation

Pour lancer automatiquement l'opération dans la soirée et la couper le matin, il faut évidemment ajouter 2 lignes au fichier `/etc/crontab` :

```
<cli>root@muffin:~# vi /etc/crontab 22 22 * * * root /mnt/muffin/scripts/lftpdaemon.sh start 07 07 * * * root /mnt/muffin/scripts/lftpdaemon.sh stop </cli>
```

From:

<https://wiki.geekitude.fr/> - **Geekitude**

Permanent link:

<https://wiki.geekitude.fr/archives/info/os/debian/lftp>

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

