

LFTP

Command line based FTP client

LFTP is a fully featured FTP client with a CLI interface:

- Multiple segmented Download
- Automation via Bash script
- Multiple simultaneous downloads
- Directory Navigation

<https://lftp.yar.ru>

Multiple Segmented Downloads

Clients that have an issue with SFTP download speeds, can take advantage of LFTP's feature to download data in multiple segments. It basically breaks down a file into smaller pieces and downloads them separately to give a boost to your overall download speed. Each piece is downloaded using it's own connection which means that on paper your speed will be multiplied by the number of segments that are there. However, in the real world that is not the case and you should experiment a little to figure out the number of segments that give you the best performance.

- Login to your Ultra.cc service via SSH. [How to connect to your Ultra.cc slot via SSH](#)
- The above is necessary so that your local machine is able to verify the host key of your Ultra.cc service when you use LFTP in the future.
- Exit your service's SSH with the following command: `exit`
- Use LFTP to access your service's SFTP server.

```
lftp sftp: //username@servername.usbx.me
```

Example:

```
lftp sftp: //ultradocs@oblivion.usbx.me
```

- Navigate to the directory on your service that contains the data that you wish to download. By default, data is saved in `~/downloads/<torrent_client_name>`. Given below are example commands to change the directory to `~/downloads/rtorrent` and then view the data that it contains.

```
cd ~/downloads/rtorrent
ls
```

- Navigate to the directory on your local machine where you wish to save the downloaded data. Given below is an example to change the local directoy to the Desktop folder and view its contents.

```
lcd C:\Users\ultradocs\Desktop
!ls
```

- Use 5 segments to download a single file.

```
pget -n 5 <filename>
```

Example:

```
pget -n 5 ubuntu.iso
```

The above will download `~/downloads/rtorrent/ubuntu.iso` to the Desktop folder on your local machine.

- Use 5 segments to download a directory.

```
mirror --use-pget-n=5 <directory_name>
```

Example:

```
mirror --use-pget-n=5 MyLinuxISOs
```

- Use segmented downloads along with parallel downloading for best download speeds. In the example given below, we are using 5 parallel downloads along with 5 segments.

```
mirror --parallel=5 --use-pget-n=5 MyLinuxISOs
```

The above will download `~/downloads/rtorrent/MyLinuxISOs` to the Desktop folder on your local machine.

As mentioned previously, you can experiment with the number of segments and parallel downloads to get maximum download speeds. We would not recommend setting a number higher than 10.

Pushing files from your slot using LFTP

This is for those who would like to automate downloading or "pushing" files off your Ultra.cc Slot to another FTP server. For example, you may run a Network Attached Storage device at home which is able to run a simple SFTP/FTP server. This command below can be set up to run automatically and send anything in one folder to your Server at your residence while you sleep. Great for grabbing the latest Nightlight Linux distro's.

```
lftp -u destinationusername,destinationpassword -e 'mirror --reverse --parallel=3 --continue --only-missing /home/usbdocs/files/LFTP /home/destinationusername/LFTP; quit'
sftp: //destinationip: port
```

The Path `/home/usbdocs/files/LFTP` is used purely as an example. You could change this to any path on your Ultra.cc Slot and LFTP would push that path to the target. You may also add multiple copies of the above line when creating your bash script (details below) to sync multiple folders to multiple target servers and/or folders.

Argument	Effect
----------	--------

use-pget-n=2	Number of Segments per file, This is best matched to the number of Parallel downloads. For example, if your parallel=2, then pget-n=2 too. Don't aim for too high a number for example 4 and 4 would be 4 downloads each in 4 segments for 16 threads even this may be higher than required. Experiment to see what works best for your target connection
continue	Create .part files in case of accidental disconnection. This will allow the next run of lftp to pick up where it left off. This may need disabling if you find the end transfer to be corrupted (just delete --continue)
mirror	specify a source and target directory
parallel=number	Number of files to Push to target at once
reverse	reverse mirror i.e. put all files from an Ultra.cc slot to a Target Server
only-missing	Push only files not already present on the target server
sftp://destinationip:port	This will need to be changed to reflect the type of Server your connecting to SFTP/FTP the example here connects to an SFTP server. destinationip and port will need filling based on your target connection details.

Please be aware that LFTP does not support use-pget-n during uploads only during downloads.

Uploading files to your slot using LFTP

Uploading to your slot is just as simple as downloading from your slot "pushing". Using the same command above will work as long as it is run from the location you want your files to arrive at. `lftp -u usbslotusername,usbslotpassword -e 'mirror --reverse --parallel=3 --continue --only-missing /home/destinationusername/LFTP /home/usbdocs/files/LFTP; quit' sftp://destinationip:22`

Notice that the main changes are the order of the directory and the address at the end. The function of the command is exactly the same; It is just the direction that has changed.

Automating the Process With Bash and Crontab

Your first step is to find the full path of your home directory `pwd` will output your full home path.

Something like this will be outputted. Be sure to make a note of it :

```
/home/usbdocs
```

then create two new folders; one may already exist. This is fine

```
mkdir ~/lock
```

```
mkdir ~/scripts
```

Then you need to enter the new folder called scripts

```
cd ~/scripts
```

And create the script file

```
nano LFTP.sh
```

Paste the following lines into it :

```
#!/bin/bash
exec {lock_fd}>/home1/usbdocs/lock/lftpLock || exit 1
flock -n "$lock_fd" || { echo "ERROR: flock() failed." >&2; exit 1; }
if [ -z "$STY" ]; then exec screen -dm -S lftp /bin/bash "$0"; fi

lftp -u destinationusername,destinationpassword 'mirror --reverse --parallel=3 --continue --
use-pget-n=5 --only-missing /home/usbdocs/files/LFTP /home/destinationusername/LFTP; quit'
sftp://destinationip:port

flock -u "$lock_fd"
```

Save it by pressing Ctrl+X, then Y, then Enter. You will need to change the paths "home/usbdocs" to match your own home and username.

Now we need to set a few things that will be passed to LFTP automatically ready for automation.

Create a folder to house the configuration file.

```
mkdir ~/.lftp
```

now create the rc file to place the options inside

```
nano ~/.lftp/rc
```

Paste these lines in and save

```
set net:limit-total-rate 0:41943040
set ssl:verify-certificate no
```

Testing the script before Automating

So to test, first we navigate to ~/scripts folder we made earlier

```
cd ~/scripts
```

Then we need to allow the LFTP.sh permissions to run

```
chmod +x LFTP.sh
```

And finally, run it

```
./LFTP.sh
```

If the script is running and you were to rerun it, you may see an error message "Flock Failed" this is a file lock to stop multiple backups running and is normal. If you are sure it isn't running, you can delete the lock file from `~/lock`. You can also check the progress of the backup script, which is running on a screen with the command

```
screen -rd lftp
```

Setting the automation and time

If all is well after the test, we can automate the check via crontab

Open crontab with

```
crontab -e
```

You may have a choice of editors. We recommend Nano

Inside the crontab add a single line under everything else in the file that looks like this

```
0 */4 * * * /home/usbdocs/scripts/LFTP.sh
```

Save it by pressing Ctrl+X then Y, then Enter.

Once again, we want to remind you that the paths will need changing to reflect the path shown by the `pwd` command

The script will now run every 4 hours, checking for files that have changed and sync them to the target server. Please be aware that anything removed from Target will be synced again. Remove the item from the source or move it to a non synced folder.

Revision #17

Created 3 February 2021 21:43:33 by Joe

Updated 14 July 2023 09:06:03 by varg