

lftpをWindows 10で使う

rsync が Windows 10 で無事動いたので、気を良くして lftp もチャレンジしてみた。

ロリポップはライトプランで契約しているのでSSHが使えない。だがFTPSは可能なので、Lftpを使ってFTPSでmirrorさせてみたい。

lftp

<http://lftp.yar.ru/>

参考サイト

lftp FTPサーバに接続してファイルを転送する

lftpで行う手軽なバックアップ

lftp + sftp(ssh) でファイルの同期をとる

CLIMマジック lftpで行う手軽なバックアップ

lftp for Windowsのダウンロード

lftpはWindowsをサポートしていないがcygwinのDLLを使ったWindows版がある。

LFTP for Windows

<https://nwgat.ninja/lftp-for-windows/>

chocolateyでも同じものが見つかる（現時点で4.8.0）。

<https://chocolatey.org/packages/lftp>

chocolatey用のインストールパッケージはこちら。

32bit - <https://f001.backblazeb2.com/file/nwgat-cdn/lftp/win32/lftp-4.8.0.win32-openssl.zip>

64bit - <https://f001.backblazeb2.com/file/nwgat-cdn/lftp/win64/lftp-4.8.0.win64-openssl.zip>

rsyncをWindow 10で使う」でcygwinの32bit版DLLを%PATH%にインストールした。そこにlftpの64bit版をインストールするとcygwinの64bit版DLLとのコンフリクトが起きてしまう。だからlftpも32bit版をインストールするchocoを使う場合には、

```
choco install --x86 lftp
```

自分でインストールする場合には、上記の32bit版ZIPファイルをダウンロードして解凍する。

```
2019/07/17 00:00 <DIR> bin
2019/07/17 00:00 <DIR> etc
2017/07/15 03:42 35,819 GPL.txt
2017/07/15 03:42 122 nwgat.ninja.url
```

binフォルダの中身は

```
2017/07/15 03:42 795,677 bash.exe
2017/07/15 03:42 12,829 cygcom_err-2.dll
2017/07/15 03:42 2,058,269 cygcrypto-1.0.0.dll
2017/07/15 03:42 110,109 cyggcc_s-1.dll
```

2017/07/15	03:42	285,213	cyggssapi_krb5-2.dll
2017/07/15	03:42	1,034,269	cygiconv-2.dll
2017/07/15	03:42	42,525	cygintl-8.dll
2017/07/15	03:42	193,565	cygk5crypto-3.dll
2017/07/15	03:42	762,397	cygkrb5-3.dll
2017/07/15	03:42	39,965	cygkrb5support-0.dll
2017/07/15	03:42	313,373	cygncursesw-10.dll
2017/07/15	03:42	227,357	cygreadline7.dll
2017/07/15	03:42	432,669	cygssl-1.0.0.dll
2017/07/15	03:42	12,829	cygssp-0.dll
2017/07/15	03:42	1,473,053	cygstdc++-6.dll
2017/07/15	03:42	3,454,230	cygwin1.dll
2017/07/15	03:42	84,519	cygz.dll
2017/07/15	03:42	1,357,312	lftp.exe
2015/06/23	14:51	676,903	sh.exe
2017/07/15	03:42	727,581	ssh.exe
20個 of ファイル		14,094,644	バイト

binフォルダの内容をPATHの通ったフォルダにコピーしておく。

lftpの実行とエラーへの対処

SFTP転送にチャレンジ

せっかくなのでSFTP転送にもチャレンジしてみたい。もちろん、ライトプランのロリポップではなく、別のサーバーを相手にしてのチャレンジだ。

[sshフォルダの準備](#)と[鍵の生成と転送](#)を行ってSFTPのためのSSHの準備をした。

```
lftp sftp://ftp.server.name
```

lftp中でlsコマンドを実行してみたら、エラーになってしまった。

```
`ls' at 0 [execl(/bin/sh) failed: No such file or directory]
```

デバッグモードで実行してみると、こんな出力行があった。

```
-- Running connect program (ssh -a -x -s -l user sftp.server.name sftp)
```

直接ssh -a -x -s -l user sftp.server.name sftp) を実行してみた。パスワード入力のおとで、

```
Permission denied, please try again.
```

と出力されたsshをつかって、リモートでsftpを起動させているが、パーミッションが許可されていないようだ(ちなみに、このサーバーはさくらのスタンダードプラン)。

ではSSHログインに使っているユーザー名を指定してみた。

```
lftp -u user,password sftp://sftp.server.name
```

残念ながら変化なし。

SFTP転送は諦めざるを得なかった(仕様のにはできるはずなんだけど)。

SFTPが使える環境ではrsyncを使えばいいので、ここは諦めることにした。

FTPS転送

```
lftp ftps://ftps.server.name
```

lftpのプロンプトでlsを実行してみると、
``ls' at 0 [Connecting...]`
と表示して固まってしまった。

デバッグモードで実行してみると、
-- Connecting to ftps.server.name (XX.XX.XX.XX) port 990
port 990 (FTPSのImplicitモード) に接続しようとしてタイムアウトしていることがわかった。

lftpは、プロトコルにFTPSを指定されるとImplicitモードでポート990に接続しようとする。ところが、サーバー側は(さくらもロリポップも)FTPSはExplicit(明示的)モード専用だったので、接続できない、というわけだ。

```
lftp -e "set ftp:ssl-force true;" ftp.server.name
```

プロトコル指定なしだとデフォルトのFTPになるが、サーバーがFTPSに対応していればExplicit(明示的)モードでFTPSが使われる。ssl-forceをtrueに設定しておく、サーバーがFTPSに対応していない場合には(パスワードの送信を拒否するため)エラーとなる。

lftpのプロンプトでlsを実行してみると、
``ls' at 0 [530 Login incorrect.]`

アノニマス・ユーザーでは当然だめということか、FTPのユーザー名とパスワードを指定する。

```
lftp -e "set ftp:ssl-force true;" -u user,password ftp.server.name
```

再びlsを実行すると。
ls: Fatal error: Certificate verification: unable to get local issuer certificate (XX:XX...
というエラーが発生した。

[lftp to webin freezes at `ls' at 0 \[Logging in...\] #1](#)

こちらによれば、SSLで接続している時に、証明書のサイト名とアドレスが合わないなどの場合にこのエラーになる。対策としては、set ssl:verify-certificate noの指定を行って証明書のチェックを省く。

```
lftp -e "set ftp:ssl-force true; set ssl:verify-certificate no" -u user,password ftp.server.name
```



FTPのほうがFTPSよりオーバーヘッドが少なく、転送が短時間で終わるので、敢えてFTPプロトコルで転送したい場合もありそうだ。そんな場合は、set ftp:ssl-allow offを指定すれば良い。

フォルダの同期やバックアップ

フォルダの同期やバックアップに使うのは mirror コマンドだ。

```
mirror [OPTS] [source [target]]
```

ダウンロードは

```
mirror remote local
```

アップロードは

```
mirror -R local remote
```

オプション

省略表記	フル表記	意味
-e	-delete	リモートに存在しないファイルを削除する
-n	-only-newer	新しいファイルのみダウンロードする
-v	-verbose[= <i>level</i>]	冗長な出力を行う
-x <i>RX</i>	-exclude <i>RX</i>	操作対象外のファイルを正規表現で指定する
-X <i>GP</i>	-exclude-glob <i>GP</i>	操作対象外のファイルをグロブパターン[*. <i>html</i> など)で指定する
-R	-reverse	逆の同期(ファイルをアップロードする)
-P	-parallel[= <i>N</i>]	<i>N</i> 個のファイルを並行してダウンロードする
	-script= <i>FILE</i>	実行されるコマンドを <i>FILE</i> に書き込むが、実際には実行しない
	-just-print, -dry-run	-script=-と同じ

```
mirror --delete --only-newer --verbose --parallel=2 source target  
mirror -env -P 2 source target
```

Lftp Mirror Command Exclude Matching Files [Regex]

実行例

```
lftp -e "set ftp:ssl-force true; set ssl:verify-certificate no; mirror -env  
-x data/cache /wiki /cygdrive/d/xampp/htdocs/wiki; bye" -u username,password  
ftps.server.lolipop.jp
```

ロリポップのFTPサーバ(SFTPではない)に接続し、リモートの/wikiフォルダを、ローカルのD:\xampp\htdocs\wikiフォルダにコピーするが、キャッシュフォルダ\data/cacheは除外する。また、リモートにないファイルはローカルから削除し(-e)新しいファイルだけダウンロードする(-n)



ローカルフォルダについてはC:\folderではなく、/cygdrive/c/folderというcygwinスタイルのフォルダ指定を行う必要がある。



LFTPは -parallel オプションを使って並列で転送できるところがナイス。rsyncにはこういう技はできないみたいで、GNU parallelを使うとかスクリプトを書くなどの手間がある。

TIPS

起動オプション

lftpの起動オプション。

オプション	意味
-d	デバッグモードをONにする。
-e <i>commands</i>	<i>commands</i> を実行する（終了しない）。
-u <i>user[,pass]</i>	ユーザー名とパスワードを指定する。
-norc	ホームディレクトリのrcファイルを実行しない。
-rcfile <i>file</i>	指定のrcファイルを実行する。
-f <i>script_file</i>	スクリプトファイルを実行して終了する。
-c <i>commands</i>	<i>commands</i> を実行して終了する。

パスワードをコマンドラインに記述するのは避けた方が良いので、[~/netrc](#) を活用すべき。

コマンド一覧

```

lftp :~> help
!<shell-command>                (commands)
alias [<name> [<value>]]        attach [PID]
bookmark [SUBCMD]
cache [SUBCMD]                  cat [-b] <files>                cd
<rdir>
chmod [OPTS] mode file...    close [-a]
[re]cls [opts] [path/][pattern] debug [OPTS] [<level>|off]        du
[options] <dirs>
edit [OPTS] <file>              exit [<code>|bg]
get [OPTS] <rfile> [-o <lfile>] glob [OPTS] <cmd> <args>
help [<cmd>]
history -w file|-r file|-c|-l [cnt] jobs [-v] [<job_no...>]
kill all|<job_no>
lcd <ldir>                        lftp [OPTS] <site>
ln [-s] <file1> <file2>        ls [<args>]
mget [OPTS] <files>
mirror [OPTS] [remote [local]] mkdir [OPTS] <dirs>
module name [args]
more <files>                       mput [OPTS] <files>
mrm <files>
mv <file1> <file2>               mmv [OPTS] <files> <target-dir>
[re]nlist [<args>]
open [OPTS] <site>               pget [OPTS] <rfile> [-o <lfile>]
put [OPTS] <lfile> [-o <rfile>]  pwd [-p]
queue [OPTS] [<cmd>]
quote <cmd>                        repeat [OPTS] [delay] [command]    rm
[-r] [-f] <files>

```

```
rmdir [-f] <dirs>
set [OPT] [<var> [<val>]]
source <file>
torrent [OPTS] <file|URL>...
wait [<jobno>]
zcat <files>

scache [<session_no>]
site <site-cmd>
user <user|URL> [<pass>]
zmore <files>
```

.lftprc

setコマンド類は、`~/.lftprc` [C:\home\ユーザー名\.lftprc] に記述しておけばLFTPの起動時に読み込まれて実行される。

環境変数

%SHELL%にcygwinのsh.exeのパスを指定しておく、`!command` が使えるようになる（さもないと、`execlp(/bin/sh) failed: No such file or directory` というエラーになる）。

include/excludeのファイル指定

1. `-include-rx-from=FILE`
2. `-exclude-rx-from=FILE`
3. `-include-glob-from=FILE`
4. `-exclude-glob-from=FILE` load include/exclude patterns from the file, one
 - ファイル名は `/cygdrive/` で始まるパスで指定する（`~/`は使えない）
 - ファイルの改行コードはLFのみにする（そうしないとパターン末尾にCR\rが含まれてしまう）

From:
<https://ragi.mokuren.ne.jp/> - らぎめも

Permanent link:
<https://ragi.mokuren.ne.jp/windows/lftp>

Last update: **2020/06/28**

