

ftp & sftp : transférer des fichiers

Le FTP (*File Transfer Protocol*) est un protocole permettant d'échanger des fichiers sur le réseau. Il est assez ancien (1985) et toujours utilisé à l'heure actuelle pour transférer des fichiers.

On l'utilise en général dans deux cas.

- Pour télécharger un fichier depuis un serveur FTP public. En général, les navigateurs web font cela de manière automatique et transparente lorsque vous cliquez sur un lien de téléchargement.

La connexion se fait alors en **mode anonyme**.

- Pour transférer des fichiers vers un serveur FTP privé (et éventuellement en télécharger aussi). Lorsque l'on prend un hébergement pour son site web, l'hébergeur nous donne en général des accès FTP pour aller y déposer les fichiers du site.

La connexion se fait donc en **mode authentifié**.

Tout le monde n'a pas forcément accès à un serveur FTP privé, aussi je vous propose pour les exemples suivants de vous connecter à un serveur FTP public (rassurez-vous, si vous voulez vous connecter à un FTP privé, la méthode reste la même).

Nous nous intéressons ici au fonctionnement du FTP en ligne de commande. Bien sûr, il existe des logiciels graphiques qui font la même chose, comme par exemple FileZilla.

Connexion à un serveur FTP

Essayons de nous connecter au serveur FTP de Debian, accessible à l'adresse suivante :
`ftp://ftp.debian.org`.

```
$ ftp ftp.debian.org
```

Le serveur FTP devrait répondre en vous demandant un login et un mot de passe. Pour les serveurs FTP publics, le login à utiliser est toujours `anonymous` (anonyme).

```
$ ftp ftp.debian.org
```

```
Connected to ftp.debian.org.
```

```
220 saens.debian.org FTP server (vsftpd)
```

```
Name (ftp.debian.org:mateo21): anonymous
```

```
331 Please specify the password.
```

Password:

Pour le mot de passe, peu importe ce que vous mettez, vous serez acceptés. :-)

Vous devriez alors voir s'afficher un message de bienvenue se terminant par :

```
230 Login successful.  
Remote system type is UNIX.  
Using binary mode to transfer files.  
ftp>
```

Vous avez maintenant un **prompt** (il s'agit du terme anglais pour *invite de commandes*) ftp> qui vous permet de rentrer des commandes FTP.

Se déplacer au sein du serveur FTP

Vous savez quoi ? Bonne nouvelle : les commandes que vous pouvez utiliser sont pour la plupart les mêmes que celles que vous connaissez. :-)

Citons dans le lot :

- `ls` : affiche le contenu du répertoire actuel ;
- `pwd` : affiche le chemin du répertoire actuel ;
- `cd` : change de répertoire.

Avec ces commandes, vous devriez déjà pouvoir vous balader sur le serveur FTP.

Faites un `ls` pour voir :

```
ftp> ls  
200 PORT command successful. Consider using PASV.  
150 Here comes the directory listing.  
drwxrwsr-x    7 1176    1176          4096 Dec 05 09:10 debian  
226 Directory send OK.
```

Les lignes commençant par un numéro sont des messages envoyés par le serveur FTP. Vous noterez que les fichiers s'affichent comme si l'on avait écrit `ls -l`.

Il y a seulement un répertoire, rendez-vous donc dans `debian` :

```
ftp> cd debian  
250 Directory successfully changed.
```

Affichez à nouveau le contenu :

```
ftp> ls
```

```
200 PORT command successful. Consider using PASV.
```

```
150 Here comes the directory listing.
```

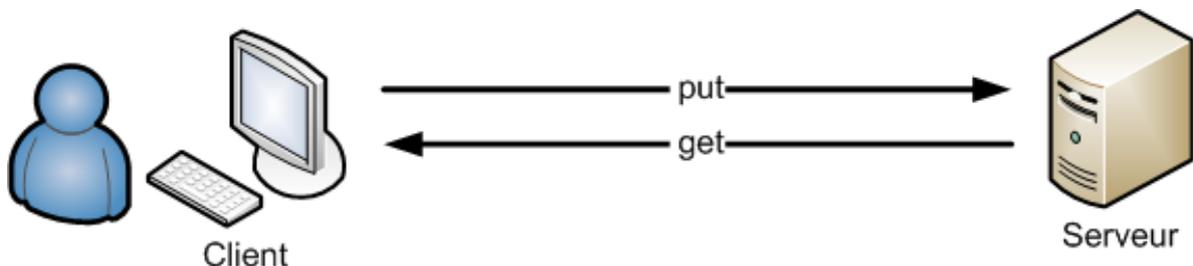
```
-rw-rw-r--    1 1176    1176          940 Oct 27 20:29 README
-rw-rw-r-
-    1 1176    1176          1290 Dec 04 2000 README.CD-
manufacture
-rw-rw-r-
-    1 1176    1176          2426 Oct 27 20:29 README.html
-rw-r--r-
-    1 1176    1176       124286 Dec 03 19:52 README.mirrors.html

-rw-r--r-
-    1 1176    1176       62059 Dec 03 19:52 README.mirrors.txt
drwxr-sr-x    9 1176    1176          4096 Nov 16 18:56 dists
drwxr-sr-x    3 1176    1176          4096 Nov 11 22:16 doc
drwxr-sr-x    3 1176    1176          4096 Dec 05 09:08 indices
-rw-rw-r--    1 1176    1176     4557196 Dec 05 08:49 ls-lR.gz
-rw-r--r--    1 1176    1176     154934 Dec 05 08:49 ls-
lR.patch.gz
drwxr-sr-x    5 1176    1176          4096 Nov 11 22:16 pool
drwxr-sr-x    4 1176    1176          4096 Nov 18 09:04 project
226 Directory send OK.
```

Le transfert de fichiers

Si vous souhaitez récupérer un fichier ou en envoyer un, il y a deux commandes à connaître (figure suivante) :

- `put` : envoie un fichier vers le serveur ;
- `get` : télécharge un fichier depuis le serveur.



Notez qu'il est impossible d'utiliser `put` sur les serveurs FTP publics comme celui auquel nous sommes connectés. Seul le téléchargement de fichiers est autorisé.

D'autres commandes, comme celle qui permet de changer les `chmod` des fichiers, ne sont pas activées non plus.

Vous pouvez par exemple récupérer le fichier `README` en écrivant `get README` :

```
ftp> get README
local: README remote: README
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for README (940 bytes).
226 File send OK.
940 bytes received in 0.00 secs (918.9 kB/s)
```

Et voilà ! Le fichier se trouve maintenant sur votre ordinateur.

Il a été téléchargé dans le dossier dans lequel vous vous trouviez sur votre ordinateur. Pour savoir dans quel dossier vous êtes **chez vous**, tapez `!pwd` :

```
ftp> !pwd
/home/mateo21
```

Si vous voulez changer de dossier **chez vous**, utilisez `!cd`. Pour lister les fichiers **chez vous**, utilisez `!ls`. Bref, vous m'avez compris, il suffit de faire précéder les commandes d'un point d'exclamation pour qu'elles s'exécutent chez vous et non sur le serveur FTP.

Les autres commandes

Il existe de nombreuses autres commandes FTP, nous n'allons pas toutes les voir.

Tapez `man ftp` pour obtenir un aperçu des commandes disponibles. Vous noterez que toutes ne sont pas identiques à celles que vous connaissez. Par exemple, pour supprimer un fichier, ce n'est pas `rm` mais... `delete` ! D'autres vous seront familières : `mkdir` permet de créer un dossier, par exemple.

Pour quitter le serveur et vous déconnecter, vous avez le choix entre la bonne vieille combinaison de touches `Ctrl + D` qui commande la fermeture de la session ou encore les commandes `bye`, `exit` et `quit`, qui sont analogues.

sftp : un FTP sécurisé

Le protocole FTP a un défaut : il n'est pas sécurisé ; les données ne sont pas cryptées.

Quelqu'un ayant accès au réseau pourrait alors intercepter le contenu des fichiers que vous échangez ou encore votre mot de passe lors de la connexion.

Pour remédier à cela, on a inventé `sftp`, qui repose sur SSH pour sécuriser la connexion :

```
sftp login@ip
```

Par exemple :

```
sftp mateo21@lisa.simple-it.fr
```

On vous demandera alors votre mot de passe (bien entendu, la clé publique sera utilisée, si elle est présente).

Une fois que vous serez connectés, les commandes sont presque les mêmes que pour le FTP. Vous retrouverez notamment `get` et `put` pour échanger des fichiers. Sachez que les commandes sont globalement plus puissantes et pratiques en SFTP qu'en FTP. Méfiez-vous toutefois, certaines commandes changent ! Par exemple, pour supprimer un fichier, ce n'est plus `delete` mais à nouveau `rm` ! Lisez le manuel pour plus d'informations : `man sftp`.

Pour se connecter en SFTP, on utilise le même port que SSH (soit 22 par défaut). Si votre serveur SSH fonctionne sur un autre port, vous devrez le préciser comme ceci : `sftp -oPort=27401 mateo21@serveur`.

À l'heure actuelle, le SFTP reste assez peu utilisé. Les hébergeurs web utilisent toujours le FTP classique alors que la plupart des logiciels graphiques comme FileZilla sont pourtant capables de se connecter en SFTP.