You can type commands in this prompt. For example:

- To change to a directory, use `cd directory`
- To change the directory of a local machine, use `lcd`
- To create a directory use `mkdir`
- To list files in the current directory on the remote machine, use `ls`
- To download a file, use `get filename` as follows:

  ```
  lftp username@ftphost:~> get filename
  ```

- To upload a file from the current directory, use `put filename` as follows:

  ```
  lftp username@ftphost:~> put filename
  ```

- An `lftp` session can be terminated by using the `quit` command

Autocompletion is supported by in the `lftp` prompt.

There's more...

Let's go through additional techniques and commands used for file transfer through a network.

Automated FTP transfer

`ftp` is another command used for FTP-based file transfer. `lftp` is more flexible for usage. `lftp` and the `ftp` commands open an interactive session with the user (it prompts for user input by displaying messages). What if we want to automate file transfer instead of an interactive mode? We can automate FTP file transfers by writing a shell script as follows:

```
#!/bin/bash
#Filename: ftp.sh
#Automated FTP transfer
HOST='domain.com'
USER='foo'
PASSWD='password'
ftp -i -n $HOST <<EOF
user ${USER} ${PASSWD}
binary
cd /home/slynux
puttestfile.jpg
getserverfile.jpg
quit
EOF
```

The preceding script has the following structure:

```
<<EOF
DATA
EOF
```

This is used to send data through `stdin` to the `ftp` command. The *Playing with file descriptors and redirection* recipe of *Chapter 1*, *Shell Something Out*, explains various methods for redirection into stdin.

The `-i` option of `ftp` turns off the interactive session with user. `user ${USER} ${PASSWD}` sets the username and password. `binary` sets the file mode to binary. The `-n` option tells `ftp` to not attempt automatically logging in and use the username and password we supply it.

SFTP (Secure FTP)

SFTP is a FTP-like file transfer system that runs on top of an SSH connection and emulates as an FTP interface. It doesn't require an FTP server at the remote end to perform file transfer, but it requires an OpenSSH server to be installed and running. It is an interactive command, which offers an `sftp` prompt.

The following commands are used to perform the file transfer. All other commands remain the same for every automated FTP session with a specific HOST, USER, and PASSWORD:

```
cd /home/slynux
put testfile.jpg
get serverfile.jpg
```

In order to run `sftp`, use:

```
$ sftp user@domainname
```

Similar to `lftp`, an `sftp` session can be terminated by typing the `quit` command.

Sometimes, the SSH server will not be running at the default port 22. If it is running at a different port, we can specify the port along with `sftp` as `-oPort=PORTNO`. For example:

```
$ sftp -oPort=422 user@slynux.org
```

`-oPort` should be the first argument of the `sftp` command.

The rsync command

`rsync` is an important command-line utility that is widely used for copying files over networks and for taking backup snapshots. This is better explained in the *Backup snapshots with rsync* recipe of *Chapter 6*, *The Backup Plan*, that explains the usage of `rsync`.