

## 在 Shell 中下载 ftp 文件的几种方法

FTP 是 File Transfer Protocol (文件传输协议) 的英文简称, 而中文简称为“文传协议”。用于 Internet 上的控制文件的双向传输。同时, 它也是一个应用程序 (Application)。基于不同的操作系统有不同的 FTP 应用程序, 而所有这些应用程序都遵守同一种协议以传输文件。在 FTP 的使用当中, 用户经常遇到两个概念: 下载 (Download) 和上传 (Upload)。下载文件就是从远程主机拷贝文件至自己的计算机上; 上传文件就是将文件从自己的计算机中拷贝至远程主机上。用 Internet 语言来说, 用户可通过客户机程序向 (从) 远程主机上传 (下载) 文件。



如果想及时了解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog\_hadoop

本文仅介绍几种通过 Shell 下载存储在 FTP 服务器上文件的几种方法。关于 FTP 的其他使用 (比如上传等) 请参见其他人写的文章。下面的操作全部在 CentOS 进行的, 如果你的服务器上找不到 ftp 命令, 请通过下面命令安装:

```
[iteblog@www.iteblog.com /user/iteblog]$ ftp
-bash: ftp: command not found
[iteblog@www.iteblog.com /user/iteblog]$ yum install ftp
```

在运行下面命令前, 请设置好相关变量的值:

```
HOST=www.iteblog.com
PORT=21
USER=iteblog
```

PASSWD=iteblog\_hadoop

请根据自己的情况设置。

## 使用 ftp 命令下载

### 下载单个文件

下载单个文件的时候可以通过下面命令实现

```
ftp -n $HOST $PORT <<ITEBLOG
user $USER $PASSWD
binary
get iteblog/2018-05-22/test.txt
quit
ITEBLOG
```

上面命令将 test.txt 文件下载到当前目录下的 iteblog/2018-05-22/ 文件夹中，我们需要事先在当前工作目录下创建好 iteblog/2018-05-22/ 目录，否则会报错。

如果你 FTP 服务器的端口为 21 ，那么我们可以不设置端口：

```
ftp -n $HOST <<ITEBLOG
user $USER $PASSWD
binary
get iteblog/2018-05-22/test.txt
quit
ITEBLOG
```

这个同样适用于下面的 wget 和 curl 方式的下载。

当然，我们也可以将文件存放到自定义的目录下面：

```
ftp -n $HOST $PORT <<ITEBLOG
user $USER $PASSWD
```

```
binary
get iteblog/2018-05-22/test.txt /home/iteblog/
quit
ITEBLOG
```

上面命令从 FTP 服务器上下载 iteblog/2018-05-22/test.txt 文件，并存放在 /home/iteblog/ 目录下。

## 下载多个文件

如果需要下载多个文件，可以使用 mget 实现：

```
ftp -n $HOST $PORT <<ITEBLOG
user $USER $PASSWD
binary
prompt
mget iteblog/2018-05-22/18987409588.txt iteblog/2018-05-22/18987931313.txt
quit
ITEBLOG
```

同上面 get 一样，这个命令将 iteblog/2018-05-22/test.txt 和 iteblog/2018-05-22/test1.txt 文件下载到当前工作目录下的 iteblog/2018-05-22/ 文件夹下

当然，我们也可以通过通配符来一次性下载所有符合条件的文件：

```
ftp -n $HOST $PORT <<ITEBLOG
user $USER $PASSWD
binary
prompt
mget iteblog/2018-05-22/*.txt
quit
ITEBLOG
```

## 通过 wget 下载

我们也可以使用 wget 下载 FTP 服务器上的文件：

```
wget -r -nH -P/tmp ftp://$HOST:$PORT/iteblog/2018-05-22/* --ftp-user=$USER --ftp-  
password=$PASSWD
```

```
--2018-05-23 17:23:29-- ftp://iteblog:*password*@www.iteblog.com/iteblog/2018-05-22/test.t  
xt
```

```
    => "test.txt"
```

```
Resolving www.iteblog.com... 10.94.76.234
```

```
Connecting to www.iteblog.com|10.94.76.234|:21... connected.
```

```
Logging in as iteblog ... Logged in!
```

```
==> SYST ... done.  ==> PWD ... done.
```

```
==> TYPE I ... done.  ==> CWD (1) /iteblog/2018-05-22 ... done.
```

```
==> SIZE test.txt ... 804763
```

```
==> PASV ... done.  ==> RETR test.txt ... done.
```

```
Length: 804763 (786K) (unauthoritative)
```

```
100%[=====>] 804,763  1.09M/s  in 0.7s
```

```
2018-05-23 17:23:31 (2.14 MB/s) - "test.txt" saved [804763]
```

-P 参数指定的是下载的文件存放目录。  
我们还可以这么写：

```
[iteblog@www.iteblog.com ~]$ wget ftp://$USER:$PASSWD@$HOST:$PORT/iteblog/2018-05-22  
/test.txt
```

```
--2018-05-23 17:35:31-- ftp://iteblog:*password*@www.iteblog.com/iteblog/2018-05-22/test.t  
xt
```

```
    => "test.txt"
```

```
Resolving www.iteblog.com... 10.94.76.234
```

```
Connecting to www.iteblog.com|10.94.76.234|:21... connected.
```

```
Logging in as iteblog ... Logged in!
```

```
==> SYST ... done.  ==> PWD ... done.
```

```
==> TYPE I ... done.  ==> CWD (1) /iteblog/2018-05-22 ... done.
```

```
==> SIZE test.txt ... 804763
```

```
==> PASV ... done.  ==> RETR test.txt ... done.
```

```
Length: 804763 (786K) (unauthoritative)
```

```
100%[=====>] 804,763  1.09M/s  in 0.7s
```

```
2018-05-23 17:35:32 (1.09 MB/s) - "test.txt" saved [804763]
```

## 通过 curl 命令下载

```
[iteblog@www.iteblog.com ~]$ curl ftp://$HOST:$PORT/iteblog/2018-05-22/test.txt -u $USER:$PASSWD -o /tmp/test.txt
```

```
% Total % Received % Xferd Average Speed Time Time Time Current
          Dload Upload Total Spent Left Speed
100 785k 100 785k 0 0 1066k 0 --:--:-- --:--:-- --:--:-- 1084k
```

还可以通过下面两种方法下载

```
curl ftp://$HOST:$PORT/iteblog/2018-05-22/test.txt --user $USER:$PASSWD -o /tmp/test.txt
```

```
curl ftp://$USER:$PASSWD@$HOST:$PORT/iteblog/2018-05-22/test.txt -o /tmp/test.txt
```

本博客文章除特别声明，全部都是原创！

原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。

本文链接: [【】](#) ( )