# Wget

Linux系统中的wget是一个下载文件的工具，它用在命令行下。对于Linux用户是必不可少的工具，我们经常要下载一些软件或从远程服务器恢复备份到本地服务器。wget支持HTTP，HTTPS和FTP协议，可以使用HTTP代理。所谓的自动下载是指，wget可以在用户退出系统的之后在后台执行。这意味这你可以登录系统，启动一个wget下载任务，然后退出系统，wget将在后台执行直到任务完成，相对于其它大部分浏览器在下载大量数据时需要用户一直的参与，这省去了极大的麻烦。

wget 可以跟踪HTML页面上的链接依次下载来创建远程服务器的本地版本，完全重建原始站点的目录结构。这又常被称作”递归下载”。在递归下载的时候，wget 遵循Robot Exclusion标准(/robots.txt). wget可以在下载的同时，将链接转换成指向本地文件，以方便离线浏览。

wget 非常稳定，它在带宽很窄的情况下和不稳定网络中有很强的适应性.如果是由于网络的原因下载失败，wget会不断的尝试，直到整个文件下载完毕。如果是服务器打断下载过程，它会再次联到服务器上从停止的地方继续下载。这对从那些限定了链接时间的服务器上下载大文件非常有用。

## 命令格式：

wget [参数] [URL地址]

## 命令功能：

用于从网络上下载资源，没有指定目录，下载资源回默认为当前目录。wget虽然功能强大，但是使用起来还是比较简单：

1）支持断点下传功能；这一点，也是网络蚂蚁和FlashGet当年最大的卖点，现在，Wget也可以使用此功能，那些网络不是太好的用户可以放心了；

2）同时支持FTP和HTTP下载方式；尽管现在大部分软件可以使用HTTP方式下载，但是，有些时候，仍然需要使用FTP方式下载软件；

3）支持代理服务器；对安全强度很高的系统而言，一般不会将自己的系统直接暴露在互联网上，所以，支持代理是下载软件必须有的功能；

4）设置方便简单；可能，习惯图形界面的用户已经不是太习惯命令行了，但是，命令行在设置上其实有更多的优点，最少，鼠标可以少点很多次，也不要担心是否错点鼠标；

5）程序小，完全免费；程序小可以考虑不计，因为现在的硬盘实在太大了；完全免费就不得不考虑了，即使网络上有很多所谓的免费软件，但是，这些软件的广告却不是我们喜欢的。

**3．命令参数：**

**启动参数：**

-V, –version 显示wget的版本后退出

-h, –help 打印语法帮助

-b, –background 启动后转入后台执行

-e, –execute=COMMAND 执行`.wgetrc’格式的命令，wgetrc格式参见/etc/wgetrc或~/.wgetrc

**记录和输入文件参数：**

-o, –output-file=FILE 把记录写到FILE文件中

-a, –append-output=FILE 把记录追加到FILE文件中

-d, –debug 打印调试输出

-q, –quiet 安静模式(没有输出)

-v, –verbose 冗长模式(这是缺省设置)

-nv, –non-verbose 关掉冗长模式，但不是安静模式

-i, –input-file=FILE 下载在FILE文件中出现的URLs

-F, –force-html 把输入文件当作HTML格式文件对待

-B, –base=URL 将URL作为在-F -i参数指定的文件中出现的相对链接的前缀

–sslcertfile=FILE 可选客户端证书

–sslcertkey=KEYFILE 可选客户端证书的KEYFILE

–egd-file=FILE 指定EGD socket的文件名

**下载参数：**

–bind-address=ADDRESS 指定本地使用地址(主机名或IP，当本地有多个IP或名字时使用)

-t, –tries=NUMBER 设定最大尝试链接次数(0 表示无限制).

-O –output-document=FILE 把文档写到FILE文件中

-nc, –no-clobber 不要覆盖存在的文件或使用.#前缀

-c, –continue 接着下载没下载完的文件

–progress=TYPE 设定进程条标记

-N, –timestamping 不要重新下载文件除非比本地文件新

-S, –server-response 打印服务器的回应

–spider 不下载任何东西

-T, –timeout=SECONDS 设定响应超时的秒数

-w, –wait=SECONDS 两次尝试之间间隔SECONDS秒

–waitretry=SECONDS 在重新链接之间等待1…SECONDS秒

–random-wait 在下载之间等待0…2\*WAIT秒

-Y, –proxy=on/off 打开或关闭代理

-Q, –quota=NUMBER 设置下载的容量限制

–limit-rate=RATE 限定下载输率

**目录参数：**

-nd –no-directories 不创建目录

-x, –force-directories 强制创建目录

-nH, –no-host-directories 不创建主机目录

-P, –directory-prefix=PREFIX 将文件保存到目录 PREFIX/…

–cut-dirs=NUMBER 忽略 NUMBER层远程目录

**HTTP 选项参数：**

–http-user=USER 设定HTTP用户名为 USER.

–http-passwd=PASS 设定http密码为 PASS

-C, –cache=on/off 允许/不允许服务器端的数据缓存 (一般情况下允许)

-E, –html-extension 将所有text/html文档以.html扩展名保存

–ignore-length 忽略 `Content-Length’头域

–header=STRING 在headers中插入字符串 STRING

–proxy-user=USER 设定代理的用户名为 USER

–proxy-passwd=PASS 设定代理的密码为 PASS

–referer=URL 在HTTP请求中包含 `Referer: URL’头

-s, –save-headers 保存HTTP头到文件

-U, –user-agent=AGENT 设定代理的名称为 AGENT而不是 Wget/VERSION

–no-http-keep-alive 关闭 HTTP活动链接 (永远链接)

–cookies=off 不使用 cookies

–load-cookies=FILE 在开始会话前从文件 FILE中加载cookie

–save-cookies=FILE 在会话结束后将 cookies保存到 FILE文件中

**FTP 选项参数：**

-nr, –dont-remove-listing 不移走 `.listing’文件

-g, –glob=on/off 打开或关闭文件名的 globbing机制

–passive-ftp 使用被动传输模式 (缺省值).

–active-ftp 使用主动传输模式

–retr-symlinks 在递归的时候，将链接指向文件(而不是目录)

**递归下载参数：**

-r, –recursive 递归下载－－慎用!

-l, –level=NUMBER 最大递归深度 (inf 或 0 代表无穷)

–delete-after 在现在完毕后局部删除文件

-k, –convert-links 转换非相对链接为相对链接

-K, –backup-converted 在转换文件X之前，将之备份为 X.orig

-m, –mirror 等价于 -r -N -l inf -nr

-p, –page-requisites 下载显示HTML文件的所有图片

**递归下载中的包含和不包含(accept/reject)：**

-A, –accept=LIST 分号分隔的被接受扩展名的列表

-R, –reject=LIST 分号分隔的不被接受的扩展名的列表

-D, –domains=LIST 分号分隔的被接受域的列表

–exclude-domains=LIST 分号分隔的不被接受的域的列表

–follow-ftp 跟踪HTML文档中的FTP链接

–follow-tags=LIST 分号分隔的被跟踪的HTML标签的列表

-G, –ignore-tags=LIST 分号分隔的被忽略的HTML标签的列表

-H, –span-hosts 当递归时转到外部主机

-L, –relative 仅仅跟踪相对链接

-I, –include-directories=LIST 允许目录的列表

-X, –exclude-directories=LIST 不被包含目录的列表

-np, –no-parent 不要追溯到父目录

wget -S –spider url 不下载只显示过程

## 使用实例：

**实例1：使用wget下载单个文件**

**命令：**

wget http://www.minjieren.com/wordpress-3.1-zh\_CN.zip

**说明：**

以下的例子是从网络下载一个文件并保存在当前目录，在下载的过程中会显示进度条，包含（下载完成百分比，已经下载的字节，当前下载速度，剩余下载时间）。

**实例2：使用wget -O下载并以不同的文件名保存**

**命令：**

: wget -O wordpress.zip http://www.minjieren.com/download.aspx?id=1080

**说明：**

wget默认会以最后一个符合”/”的后面的字符来命令，对于动态链接的下载通常文件名会不正确。

错误：下面的例子会下载一个文件并以名称download.aspx?id=1080保存

wget http://www.minjieren.com/download?id=1

即使下载的文件是zip格式，它仍然以download.php?id=1080命令。

正确：为了解决这个问题，我们可以使用参数-O来指定一个文件名：

wget -O wordpress.zip http://www.minjieren.com/download.aspx?id=1080

**实例3：使用wget –limit -rate限速下载**

**命令：**

wget --limit-rate=300k http://www.minjieren.com/wordpress-3.1-zh\_CN.zip

**说明：**

当你执行wget的时候，它默认会占用全部可能的宽带下载。但是当你准备下载一个大文件，而你还需要下载其它文件时就有必要限速了。

**实例4：使用wget -c断点续传**

**命令：**

wget -c http://www.minjieren.com/wordpress-3.1-zh\_CN.zip

**说明：**

使用wget -c重新启动下载中断的文件，对于我们下载大文件时突然由于网络等原因中断非常有帮助，我们可以继续接着下载而不是重新下载一个文件。需要继续中断的下载时可以使用-c参数。

**实例5：使用wget -b后台下载**

**命令：**

wget -b http://www.minjieren.com/wordpress-3.1-zh\_CN.zip

**说明：**

对于下载非常大的文件的时候，我们可以使用参数-b进行后台下载。

wget -b http://www.minjieren.com/wordpress-3.1-zh\_CN.zip

Continuing in background, pid 1840.

Output will be written to `wget-log'.

你可以使用以下命令来察看下载进度：

tail -f wget-log

**实例6：伪装代理名称下载**

**命令：**

wget --user-agent="Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US) AppleWebKit/534.16 (KHTML, like Gecko) Chrome/10.0.648.204 Safari/534.16" http://www.minjieren.com/wordpress-3.1-zh\_CN.zip

**说明：**

有些网站能通过根据判断代理名称不是浏览器而拒绝你的下载请求。不过你可以通过–user-agent参数伪装。

**实例7：使用wget –spider测试下载链接**

**命令：**

wget --spider URL

**说明：**

当你打算进行定时下载，你应该在预定时间测试下载链接是否有效。我们可以增加–spider参数进行检查。

wget --spider URL

如果下载链接正确，将会显示

wget --spider URL

Spider mode enabled. Check if remote file exists.

HTTP request sent, awaiting response... 200 OK

Length: unspecified [text/html]

Remote file exists and could contain further links,

but recursion is disabled -- not retrieving.

这保证了下载能在预定的时间进行，但当你给错了一个链接，将会显示如下错误

wget --spider url

Spider mode enabled. Check if remote file exists.

HTTP request sent, awaiting response... 404 Not Found

Remote file does not exist -- broken link!!!

你可以在以下几种情况下使用spider参数：

定时下载之前进行检查

间隔检测网站是否可用

检查网站页面的死链接

**实例8：使用wget –tries增加重试次数**

**命令：**

wget --tries=40 URL

**说明：**

如果网络有问题或下载一个大文件也有可能失败。wget默认重试20次连接下载文件。如果需要，你可以使用–tries增加重试次数。

**实例9：使用wget -i下载多个文件**

**命令：**

wget -i filelist.txt

**说明：**

首先，保存一份下载链接文件

cat > filelist.txt

url1

url2

url3

url4

接着使用这个文件和参数-i下载

**实例10：使用wget –mirror镜像网站**

**命令：**

wget --mirror -p --convert-links -P ./LOCAL URL

**说明：**

下载整个网站到本地。

–miror:开户镜像下载

-p:下载所有为了html页面显示正常的文件

–convert-links:下载后，转换成本地的链接

-P ./LOCAL：保存所有文件和目录到本地指定目录

**实例11：使用wget –reject过滤指定格式下载**

**命令：**wget --reject=gif ur

**说明：**

下载一个网站，但你不希望下载图片，可以使用以下命令。

**实例12：使用wget -o把下载信息存入日志文件**

**命令：**

wget -o download.log URL

**说明：**

不希望下载信息直接显示在终端而是在一个日志文件，可以使用

**实例13：使用wget -Q限制总下载文件大小**

**命令：**

wget -Q5m -i filelist.txt

**说明：**

当你想要下载的文件超过5M而退出下载，你可以使用。注意：这个参数对单个文件下载不起作用，只能递归下载时才有效。

**实例14：使用wget -r -A下载指定格式文件**

**命令：**

wget -r -A.pdf url

**说明：**

可以在以下情况使用该功能：

下载一个网站的所有图片

下载一个网站的所有视频

下载一个网站的所有PDF文件

**实例15：使用wget FTP下载**

**命令：**

wget ftp-url

wget --ftp-user=USERNAME --ftp-password=PASSWORD url

**说明：**

可以使用wget来完成ftp链接的下载。

使用wget匿名ftp下载：

wget ftp-url

使用wget用户名和密码认证的ftp下载

wget --ftp-user=USERNAME --ftp-password=PASSWORD url

**备注：编译安装**

使用如下命令编译安装：

# tar zxvf wget-1.9.1.tar.gz

# cd wget-1.9.1

# ./configure

# make

# make install

# curl命令

**curl命令**是一个利用URL规则在命令行下工作的文件传输工具。它支持文件的上传和下载，所以是综合传输工具，但按传统，习惯称curl为下载工具。作为一款强力工具，curl支持包括HTTP、HTTPS、[ftp](http://man.linuxde.net/ftp)等众多协议，还支持POST、cookies、认证、从指定偏移处下载部分文件、用户代理字符串、限速、文件大小、进度条等特征。做网页处理流程和数据检索自动化，curl可以祝一臂之力。

## 语法

curl(选项)(参数)

## 选项

|  |  |
| --- | --- |
| -a/--append | 上传文件时，附加到目标文件 |
| -A/--user-agent <string> | 设置用户代理发送给服务器 |
| -anyauth | 可以使用“任何”身份验证方法 |
| -b/--cookie <name=string/[file](http://man.linuxde.net/file)> | cookie字符串或文件读取位置 |
| --basic | 使用HTTP基本验证 |
| -B/--use-ascii | 使用ASCII /文本传输 |
| -c/--cookie-jar <file> | 操作结束后把cookie写入到这个文件中 |
| -C/--continue-[at](http://man.linuxde.net/at) <offset> | 断点续转 |
| -d/--data <data> | HTTP POST方式传送数据 |
| --data-ascii <data> | 以ascii的方式post数据 |
| --data-binary <data> | 以二进制的方式post数据 |
| --negotiate | 使用HTTP身份验证 |
| --digest | 使用数字身份验证 |
| --disable-eprt | 禁止使用EPRT或LPRT |
| --disable-epsv | 禁止使用EPSV |
| -D/--[dump](http://man.linuxde.net/dump)-header <file> | 把header信息写入到该文件中 |
| --egd-file <file> | 为随机数据(SSL)设置EGD socket路径 |
| --tcp-nodelay | 使用TCP\_NODELAY选项 |
| -e/--referer | 来源网址 |
| -E/--cert <cert[:[passwd](http://man.linuxde.net/passwd)]> | 客户端证书文件和密码 (SSL) |
| --cert-[type](http://man.linuxde.net/type) <type> | 证书文件类型 (DER/PEM/ENG) (SSL) |
| --key <key> | 私钥文件名 (SSL) |
| --key-type <type> | 私钥文件类型 (DER/PEM/ENG) (SSL) |
| --pass <pass> | 私钥密码 (SSL) |
| --engine <eng> | 加密引擎使用 (SSL). "--engine list" for list |
| --cacert <file> | CA证书 (SSL) |
| --capath <directory> | CA目录 (made using c\_rehash) to verify peer against (SSL) |
| --ciphers <list> | SSL密码 |
| --compressed | 要求返回是压缩的形势 (using deflate or [gzip](http://man.linuxde.net/gzip)) |
| --connect-timeout <seconds> | 设置最大请求时间 |
| --create-[dirs](http://man.linuxde.net/dirs) | 建立本地目录的目录层次结构 |
| --crlf | 上传是把LF转变成CRLF |
| -f/--fail | 连接失败时不显示http错误 |
| --ftp-create-dirs | 如果远程目录不存在，创建远程目录 |
| --ftp-method [multicwd/nocwd/singlecwd] | 控制CWD的使用 |
| --ftp-pasv | 使用 PASV/EPSV 代替端口 |
| --ftp-skip-pasv-[ip](http://man.linuxde.net/ip) | 使用PASV的时候,忽略该IP地址 |
| --ftp-ssl | 尝试用 SSL/TLS 来进行ftp数据传输 |
| --ftp-ssl-reqd | 要求用 SSL/TLS 来进行ftp数据传输 |
| -F/--form <name=content> | 模拟http表单提交数据 |
| --form-string <name=string> | 模拟http表单提交数据 |
| -g/--globoff | 禁用网址序列和范围使用{}和[] |
| -G/--get | 以get的方式来发送数据 |
| -H/--header <line> | 自定义头信息传递给服务器 |
| --ignore-content-length | 忽略的HTTP头信息的长度 |
| -i/--include | 输出时包括protocol头信息 |
| -I/--[head](http://man.linuxde.net/head) | 只显示请求头信息 |
| -j/--junk-session-cookies | 读取文件进忽略session cookie |
| --interface <interface> | 使用指定网络接口/地址 |
| --krb4 <level> | 使用指定安全级别的krb4 |
| -k/--insecure | 允许不使用证书到SSL站点 |
| -K/--config | 指定的配置文件读取 |
| -l/--list-only | 列出ftp目录下的文件名称 |
| --limit-rate <rate> | 设置传输速度 |
| --local-port<NUM> | 强制使用本地端口号 |
| -m/--max-[time](http://man.linuxde.net/time) <seconds> | 设置最大传输时间 |
| --max-redirs <num> | 设置最大读取的目录数 |
| --max-filesize <bytes> | 设置最大下载的文件总量 |
| -M/--manual | 显示全手动 |
| -n/--netrc | 从netrc文件中读取用户名和密码 |
| --netrc-optional | 使用 .netrc 或者 URL来覆盖-n |
| --ntlm | 使用 HTTP NTLM 身份验证 |
| -N/--no-buffer | 禁用缓冲输出 |
| -o/--output | 把输出写到该文件中 |
| -O/--remote-name | 把输出写到该文件中，保留远程文件的文件名 |
| -p/--proxytunnel | 使用HTTP代理 |
| --proxy-anyauth | 选择任一代理身份验证方法 |
| --proxy-basic | 在代理上使用基本身份验证 |
| --proxy-digest | 在代理上使用数字身份验证 |
| --proxy-ntlm | 在代理上使用ntlm身份验证 |
| -P/--ftp-port <address> | 使用端口地址，而不是使用PASV |
| -q | 作为第一个参数，关闭 .curlrc |
| -Q/--quote <cmd> | 文件传输前，发送命令到服务器 |
| -r/--range <range> | 检索来自HTTP/1.1或FTP服务器字节范围 |
| --range-file | 读取（SSL）的随机文件 |
| -R/--remote-time | 在本地生成文件时，保留远程文件时间 |
| --retry <num> | 传输出现问题时，重试的次数 |
| --retry-delay <seconds> | 传输出现问题时，设置重试间隔时间 |
| --retry-max-time <seconds> | 传输出现问题时，设置最大重试时间 |
| -s/--silent | 静默模式。不输出任何东西 |
| -S/--show-error | 显示错误 |
| --socks4 <[host](http://man.linuxde.net/host)[:port]> | 用socks4代理给定主机和端口 |
| --socks5 <host[:port]> | 用socks5代理给定主机和端口 |
| --stderr <file> |  |
| -t/--[telnet](http://man.linuxde.net/telnet)-option <OPT=val> | Telnet选项设置 |
| --trace <file> | 对指定文件进行debug |
| --trace-ascii <file> | Like --跟踪但没有hex输出 |
| --trace-time | 跟踪/详细输出时，添加时间戳 |
| -T/--upload-file <file> | 上传文件 |
| --url <URL> | Spet URL to work with |
| -u/--user <user[:password]> | 设置服务器的用户和密码 |
| -U/--proxy-user <user[:password]> | 设置代理用户名和密码 |
| -[w](http://man.linuxde.net/w)/--[write](http://man.linuxde.net/write)-out [format] | 什么输出完成后 |
| -x/--proxy <host[:port]> | 在给定的端口上使用HTTP代理 |
| -X/--request <[command](http://man.linuxde.net/command)> | 指定什么命令 |
| -y/--speed-time | 放弃限速所要的时间，默认为30 |
| -Y/--speed-limit | 停止传输速度的限制，速度时间 |

## 实例

**文件下载**

curl命令可以用来执行下载、发送各种HTTP请求，指定HTTP头部等操作。如果系统没有curl可以使用[yum](http://man.linuxde.net/yum) [install](http://man.linuxde.net/install) curl安装，也可以下载安装。curl是将下载文件输出到stdout，将进度信息输出到stderr，不显示进度信息使用--silent选项。

curl URL --silent

这条命令是将下载文件输出到终端，所有下载的数据都被写入到stdout。

使用选项-O将下载的数据写入到文件，必须使用文件的绝对地址：

curl http://[man](http://man.linuxde.net/man).linuxde.net/text.iso --silent -O

选项-o将下载数据写入到指定名称的文件中，并使用--progress显示进度条：

curl http://man.linuxde.net/[test](http://man.linuxde.net/test).iso -o filename.iso --progress

######################################### 100.0%

**断点续传**

curl能够从特定的文件偏移处继续下载，它可以通过指定一个便宜量来下载部分文件：

curl URL/File -C 偏移量

#偏移量是以字节为单位的整数，如果让curl自动推断出正确的续传位置使用-C -：

curl -C -URL

**使用curl设置参照页字符串**

参照页是位于HTTP头部中的一个字符串，用来表示用户是从哪个页面到达当前页面的，如果用户点击网页A中的某个连接，那么用户就会跳转到B网页，网页B头部的参照页字符串就包含网页A的URL。

使用--referer选项指定参照页字符串：

curl --referer http://www.google.com http://man.linuxde.net

**用curl设置cookies**

使用--cookie "COKKIES"选项来指定cookie，多个cookie使用分号分隔：

curl http://man.linuxde.net --cookie "user=root;pass=123456"

将cookie另存为一个文件，使用--cookie-jar选项：

curl URL --cookie-jar cookie\_file

**用curl设置用户代理字符串**

有些网站访问会提示只能使用IE浏览器来访问，这是因为这些网站设置了检查用户代理，可以使用curl把用户代理设置为IE，这样就可以访问了。使用--user-agent或者-A选项：

curl URL --user-agent "Mozilla/5.0"

curl URL -A "Mozilla/5.0"

其他HTTP头部信息也可以使用curl来发送，使用-H"头部信息" 传递多个头部信息，例如：

curl -H "Host:man.linuxde.net" -H "[accept](http://man.linuxde.net/accept)-language:zh-cn" URL

**curl的带宽控制和下载配额**

使用--limit-rate限制curl的下载速度：

curl URL --limit-rate 50k

命令中用k（千字节）和m（兆字节）指定下载速度限制。

使用--max-filesize指定可下载的最大文件大小：

curl URL --max-filesize bytes

如果文件大小超出限制，命令则返回一个非0退出码，如果命令正常则返回0。

**用curl进行认证**

使用curl选项 -u 可以完成HTTP或者FTP的认证，可以指定密码，也可以不指定密码在后续操作中输入密码：

curl -u user:[pwd](http://man.linuxde.net/pwd) http://man.linuxde.net

curl -u user http://man.linuxde.net

**只打印响应头部信息**

通过-I或者-head可以只打印出HTTP头部信息：

[root@localhost text]# curl -I http://man.linuxde.net

HTTP/1.1 200 OK

Server: nginx/1.2.5

[date](http://man.linuxde.net/date): Mon, 10 Dec 2012 09:24:34 GMT

Content-Type: text/html; charset=UTF-8

Connection: keep-alive

Vary: Accept-Encoding

X-Pingback: http://man.linuxde.net/xmlrpc.[php](http://man.linuxde.net/php)