



链滴

linux 命令 -curl

作者: [linyu](#)

原文链接: <https://ld246.com/article/1491448122524>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<td> --data-binary </td>	<td>以二进制的方式post数据</td>
<td> --negotiate</td>	<td>使用HTTP身份验证</td>
<td> --digest</td>	<td>使用数字身份验证</td>
<td> --disable-eprt</td>	<td>禁止使用EPRT或LPRT</td>
<td> --disable-epsv</td>	<td>禁止使用EPSV</td>
<td>-D/--dump-header </td>	<td>把header信息写入到该文件中</td>
<td> --egd-file </td>	<td>为随机数据(SSL)设置EGD socket路径</td>
<td> --tcp-nodelay</td>	<td>使用TCP_NODELAY选项</td>
<td>-e/--referer</td>	<td>来源网址</td>
<td>-E/--cert </td>	<td>客户端证书文件和密码 (SSL)</td>
<td> --cert-type </td>	<td>证书文件类型 (DER/PEM/ENG) (SSL)</td>
<td> --key </td>	<td>私钥文件名 (SSL)</td>
<td> --key-type </td>	<td>私钥文件类型 (DER/PEM/ENG) (SSL)</td>

 --pass	私钥密码 (SSL)
 --engine	加密引擎使用 (SSL). "--engine list" for list
 --cacert	CA证书 (SSL)
 --capath	CA目录 (made using c_rehash) to verify peer against (SSL)
 --ciphers	SSL密码
 --compressed	要求返回是压缩的形势 (using deflate or gzip)
 --connect-timeout	设置最大请求时间
 --create-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	使用PASV的时候,忽略该IP地址

<td>-k/--insecure</td>	
</tr>	
<td>-K/--config</td>	
<td>指定的配置文件读取</td>	
</tr>	
<td>-l/--list-only</td>	
<td>列出ftp目录下的文件名称</td>	
</tr>	
<td> --limit-rate </td>	
<td>设置传输速度</td>	
</tr>	
<td> --local-port</td>	
<td>强制使用本地端口号</td>	
</tr>	
<td>-m/--max-time </td>	
<td>设置最大传输时间</td>	
</tr>	
<td> --max-redirs </td>	
<td>设置最大读取的目录数</td>	
</tr>	
<td> --max-filesize </td>	
<td>设置最大下载的文件总量</td>	
</tr>	
<td>-M/--manual</td>	
<td>显示全手动</td>	
</tr>	
<td>-n/--netrc</td>	
<td>从netrc文件中读取用户名和密码</td>	
</tr>	
<td> --netrc-optional</td>	
<td>使用 .netrc 或者 URL来覆盖-n</td>	
</tr>	
<td> --ntlm</td>	
<td>使用 HTTP NTLM 身份验证</td>	
</tr>	
<td>-N/--no-buffer</td>	
<td>禁用缓冲输出</td>	
</tr>	
<td>-o/--output</td>	
<td>把输出写到该文件中</td>	

 --retry-delay	传输出现问题时，设置重试间隔时间
 --retry-max-time	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	用socks4代理给定主机和端口
 --socks5	用socks5代理给定主机和端口
 --stderr	
-t/--telnet-option	Telnet选项设置
 --trace	对指定文件进行debug
 --trace-ascii	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	上传文件
 --url	Spet URL to work with
-u/--user	设置服务器的用户和密码

</tr>
<tr>
<td>-U/--proxy-user </td>
<td>设置代理用户名和密码</td>
</tr>
<tr>
<td>-w/--write-out [format]</td>
<td>什么输出完成后</td>
</tr>
<tr>
<td>-x/--proxy </td>
<td>在给定的端口上使用HTTP代理</td>
</tr>
<tr>
<td>-X/--request </td>
<td>指定什么命令</td>
</tr>
<tr>
<td>-y/--speed-time</td>
<td>放弃限速所要的时间，默认为30</td>
</tr>
<tr>
<td>-Y/--speed-limit</td>
<td>停止传输速度的限制，速度时间</td>
</tr>

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 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.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.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
</code></pre>
<p>使用 curl 设置参照页字符串</p>
<p>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。</p>
<p>使用 <code>--referer</code> 选项指定参照页字符串：</p>
<pre><code class="highlight-chroma">curl --referer http://www.google.com http://man.linuxde.net
</code></pre>
<p>用 curl 设置 cookies</p>
<p>使用 <code>--cookie "COKKIES"</code> 选项来指定 cookie，多个 cookie 使用分号分隔：
</p>
<pre><code class="highlight-chroma">curl http://man.linuxde.net --cookie "user=root;pass=123456"
</code></pre>
<p>将 cookie 另存为一个文件，使用 <code>--cookie-jar</code> 选项：</p>
<pre><code class="highlight-chroma">curl URL --cookie-jar cookie_file
</code></pre>
<p>用 curl 设置用户代理字符串</p>
<p>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，可以用 curl 把用户代理设置为 IE，这样就可以访问了。使用 <code>--user-agent</code> 或者 <code>-A</code> 选项：</p>
<pre><code class="highlight-chroma">curl URL --user-agent "Mozilla/5.0"
curl URL -A "Mozilla/5.0"
</code></pre>
<p>其他 HTTP 头部信息也可以使用 curl 来发送，使用 <code>-H</code> "头部信息" 传递多个头信息，例如：</p>
<pre><code class="highlight-chroma">curl -H "Host:man.linuxde.net" -H "accept-language:zh-cn" URL
</code></pre>
<p>curl 的带宽控制和下载配额</p>
<p>使用 <code>--limit-rate</code> 限制 curl 的下载速度：</p>
<pre><code class="highlight-chroma">curl URL --limit-rate 50k
</code></pre>
<p>命令中用 k（千字节）和 m（兆字节）指定下载速度限制。</p>
<p>使用 <code>--max-filesize</code> 指定可下载的最大文件大小：</p>
<pre><code class="highlight-chroma">curl URL --max-filesize bytes
</code></pre>
<p>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。</p>
<p>用 curl 进行认证</p>
<p>使用 curl 选项 -u 可以完成 HTTP 或者 FTP 的认证，可以指定密码，也可以不指密码在后续操作中输入密码：</p>
<pre><code class="highlight-chroma">curl -u user:pwd http://man.linuxde.net
curl -u user http://man.linuxde.net
</code></pre>
<p>只打印响应头部信息</p>

<p>通过 <code>-I</code> 或者 <code>-head</code> 可以只打印出 HTTP 头部信息: </p>
<pre><code class="highlight-chroma">[root@localhost text]# curl -I http://man.linuxde.net
HTTP/1.1 200 OK
Server: nginx/1.2.5
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
</code></pre>