



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 \(CC BY-SA 4.0\)](#)

```
<h2 id="curl">curl</h2>
<p>利用 URL 规则在命令行下工作的文件传输工具</p>
<h2 id="补充说明">补充说明</h2>
<p><strong>curl 命令</strong> 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 等众多协议，还支持 POST、cookies、认证、从指定偏移处下载部分件、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 可祝一臂之力。</p>
<h3 id="语法">语法</h3>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">curl(选项)(参数)</span></span></code></pre>
<h3 id="选项">选项</h3>
<table>
<tbody>
<tr>
<td>-a/--append</td>
<td>上传文件时，附加到目标文件</td>
</tr>
<tr>
<td>-A/--user-agent </td>
<td>设置用户代理发送给服务器</td>
</tr>
<tr>
<td>-anyauth</td>
<td>可以使用“任何”身份验证方法</td>
</tr>
<tr>
<td>-b/--cookie </td>
<td>cookie字符串或文件读取位置</td>
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--basic</td>
<td>使用HTTP基本验证</td>
</tr>
<tr>
<td>-B/--use-ascii</td>
<td>使用ASCII /文本传输</td>
</tr>
<tr>
<td>-c/--cookie-jar </td>
<td>操作结束后把cookie写入到这个文件中</td>
</tr>
<tr>
<td>-C/--continue-at </td>
<td>断点续转</td>
</tr>
<tr>
<td>-d/--data </td>
<td>HTTP POST方式传送数据</td>
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--data-ascii </td>
<td>以ascii的方式post数据</td>
```

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

```
<td>&nbsp; &nbsp; &nbsp;--pass </td>
<td>私钥密码 (SSL)</td>
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--engine </td>
<td>加密引擎使用 (SSL). "--engine list" for list</td>
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--cacert </td>
<td>CA证书 (SSL)</td>
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--capath </td>
<td>CA目录 (made using c_rehash) to verify peer against (SSL)</td>
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--ciphers </td>
<td>SSL密码</td>
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--compressed </td>
<td>要求返回是压缩的形势 (using deflate or gzip)</td>
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--connect-timeout </td>
<td>设置最大请求时间</td>
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--create-dirs</td>
<td>建立本地目录的目录层次结构</td>
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--crlf</td>
<td>上传是把LF转变成CRLF</td>
</tr>
<tr>
<td>-f/--fail</td>
<td>连接失败时不显示http错误</td>
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--ftp-create-dirs</td>
<td>如果远程目录不存在, 创建远程目录</td>
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--ftp-method [multicwd/nocwd/single cwd]</td>
<td>控制CWD的使用</td>
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--ftp-pasv</td>
<td>使用 PASV/EPSV 代替端口</td>
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--ftp-skip-pasv-ip</td>
<td>使用PASV的时候,忽略该IP地址</td>
```

```
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--ftp-ssl</td>
<td>尝试用 SSL/TLS 来进行ftp数据传输</td>
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--ftp-ssl-reqd</td>
<td>要求用 SSL/TLS 来进行ftp数据传输</td>
</tr>
<tr>
<td>-F/--form </td>
<td>模拟http表单提交数据</td>
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--form-string </td>
<td>模拟http表单提交数据</td>
</tr>
<tr>
<td>-g/--globoff</td>
<td>禁用网址序列和范围使用{}和[]</td>
</tr>
<tr>
<td>-G/--get</td>
<td>以get的方式来发送数据</td>
</tr>
<tr>
<td>-H/--header </td>
<td>自定义头信息传递给服务器</td>
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--ignore-content-length</td>
<td>忽略的HTTP头信息的长度</td>
</tr>
<tr>
<td>-j/--include</td>
<td>输出时包括protocol头信息</td>
</tr>
<tr>
<td>-l/--head</td>
<td>只显示请求头信息</td>
</tr>
<tr>
<td>-j/--junk-session-cookies</td>
<td>读取文件进忽略session cookie</td>
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--interface </td>
<td>使用指定网络接口/地址</td>
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--krb4 </td>
<td>使用指定安全级别的krb4</td>
</tr>
<tr>
```

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

```
</tr>
<tr>
<td>-O/--remote-name</td>
<td>把输出写到该文件中，保留远程文件的文件名</td>
</tr>
<tr>
<td>-p/--proxytunnel</td>
<td>使用HTTP代理</td>
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--proxy-anyauth</td>
<td>选择任一代理身份验证方法</td>
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--proxy-basic</td>
<td>在代理上使用基本身份验证</td>
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--proxy-digest</td>
<td>在代理上使用数字身份验证</td>
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--proxy-ntlm</td>
<td>在代理上使用ntlm身份验证</td>
</tr>
<tr>
<td>-P/--ftp-port </td>
<td>使用端口地址，而不是使用PASV</td>
</tr>
<tr>
<td>-q</td>
<td>作为第一个参数，关闭 .curlrc</td>
</tr>
<tr>
<td>-Q/--quote </td>
<td>文件传输前，发送命令到服务器</td>
</tr>
<tr>
<td>-r/--range </td>
<td>检索来自HTTP/1.1或FTP服务器字节范围</td>
</tr>
<tr>
<td>--range-file</td>
<td>读取 (SSL) 的随机文件</td>
</tr>
<tr>
<td>-R/--remote-time</td>
<td>在本地生成文件时，保留远程文件时间</td>
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--retry </td>
<td>传输出现问题时，重试的次数</td>
</tr>
<tr>
```

```
<td>&nbsp; &nbsp; &nbsp;--retry-delay </td>
<td>传输出现问题时，设置重试间隔时间</td>
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--retry-max-time </td>
<td>传输出现问题时，设置最大重试时间</td>
</tr>
<tr>
<td>-s/--silent</td>
<td>静默模式。不输出任何东西</td>
</tr>
<tr>
<td>-S/--show-error</td>
<td>显示错误</td>
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--socks4 </td>
<td>用socks4代理给定主机和端口</td>
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--socks5 </td>
<td>用socks5代理给定主机和端口</td>
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--stderr </td>
<td>&nbsp;</td>
</tr>
<tr>
<td>-t/--telnet-option </td>
<td>Telnet选项设置</td>
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--trace </td>
<td>对指定文件进行debug</td>
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--trace-ascii </td>
<td>Like --跟踪但没有hex输出</td>
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--trace-time </td>
<td>跟踪/详细输出时，添加时间戳</td>
</tr>
<tr>
<td>-T/--upload-file </td>
<td>上传文件</td>
</tr>
<tr>
<td>&nbsp; &nbsp; &nbsp;--url </td>
<td>Spes URL to work with</td>
</tr>
<tr>
<td>-u/--user </td>
<td>设置服务器的用户和密码</td>
```

```
</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>
</tbody>
</table>
<h3 id="实例">实例</h3>
<p><strong>文件下载</strong></p>
<p>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 <code>yum install curl</code> 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 <code>--silent</code> 选项。</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">curl URL --silent</span></span></code></pre>
<p>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。</p>
<p>使用选项 <code>-O</code> 将下载的数据写入到文件，必须使用文件的绝对地址：</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">curl http://man.linuxde.net/text.iso --silent -O</span></span></code></pre>
<p>选项 <code>-o</code> 将下载数据写入到指定名称的文件中，并使用 <code>--progress</code> 显示进度条：</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">curl http://man.linuxde.net/test.iso -o filename.iso --progress</span></span><span class="highlight-line"><span class="highlight-cl">##### 100.0%</span></span></code></pre>
<p><strong>断点续传</strong></p>
<p>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：</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">curl URL/File -C 偏移量</span></span><span class="highlight-line"><span class="highlight-cl">#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 -:</span></span></code></pre>
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">curl -C -URL</span></span></code></pre>
<p><strong>使用 curl 设置参照页字符串</strong></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，如果用点击网页 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"><span class="highlight-line"><span class="highlight-cl">curl --referer http://www.google.com http://man.linuxde.net</span></span></code></pre>
<p><strong>用 curl 设置 cookies</strong></p>
<p>使用 <code>--cookie "COOKIES"</code> 选项来指定 cookie，多个 cookie 使用分号分隔：</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">curl http://man.linuxde.net --cookie "user=root;pass=123456"</span></span></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"><span class="highlight-line"><span class="highlight-cl">curl URL --cookie-jar cookie_file</span></span></code></pre>
<p><strong>用 curl 设置用户代理字符串</strong></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"><span class="highlight-line"><span class="highlight-cl">curl URL --user-agent "Mozilla/5.0"</span></span></code></pre>
<span><span class="highlight-line"><span class="highlight-cl">curl URL -A "Mozilla/5.0"</span></span></span></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"><span class="highlight-line"><span class="highlight-cl">curl -H "Host:man.linuxde.net" -H "accept-language:zh-cn" URL</span></span></code></pre>
<p><strong>curl 的带宽控制和下载配额</strong></p>
<p>使用 <code>--limit-rate</code> 限制 curl 的下载速度：</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">curl URL --limit-rate 50k</span></span></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"><span class="highlight-line"><span class="highlight-cl">curl URL --max-filesize bytes</span></span></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><strong>用 curl 进行认证</strong></p>
<p>使用 curl 选项 &nbsp;-u &nbsp;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"><span class="highlight-line"><span class="highlight-cl">curl -u user:pwd http://man.linuxde.net</span></span></code></pre>
<span><span class="highlight-line"><span class="highlight-cl">curl -u user http://man.linuxde.net</span></span></span></code></pre>
<p><strong>只打印响应头部信息</strong></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.netHTTP/1.1 200 OKServer: nginx/1.2.5date: Mon, 10 Dec 2012 09:24:34 GMTContent-Type: text/html; charset=UTF-8Connection: keep aliveVary: Accept-EncodingX-Pingback: http://man.linuxde.net/xmlrpc.php</code></pre>