



链滴

sed 入门详解

作者: [jianhongli](#)

原文链接: <https://ld246.com/article/1554039537192>

来源网站: [链滴](#)

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

<p> </p>

<blockquote>

<p>由于 <code>sed</code> 命令本身特别复杂,完全把此命令搞明白并记住不忘是非常困难的.因能够对于一些经常使用的场景总结出最佳实践一个不错的方式.</p>

</blockquote>

<h3 id="sed-工作模式">sed 工作模式</h3>

<p>其命令格式如下:</p>

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">sed -e [option] 'command' file</span></span></code></pre>
```

<h4 id="command-的格式构成">command 的格式构成</h4>

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">[addr]X[options]</span></span></code></pre>
```

<p>一个命令由地址空间,命令字母,以及相应的命令选项(可选),比如命令:<code>4,5p;q43</code>;先匹配地址空间 4 行与 5 行,然后执行打印命令 <code>p</code>,命令 <code>p</code> 没有带选参数,打印完成后执行 <code>q</code> 命令,命令 <code>q</code> 带了可选项参数:<code>4</code> 表明命令的退出 code 为:<code>42</code>;下面对此命令格式进行详细的介绍;</p>

<h5 id="-addr-">[addr]</h5>

<p>一个地址空间是后面跟的命令的执行匹配行的一个选择条件.后面的命令只对此地址空间匹配的行生作用;如下命令:</p>

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">sed '144s/hello/world/' input.txt &gt; output.txt</span></span></code></pre>
```

<p>Addresses determine on which line(s) the <code>sed</code> command will be executed The following command replaces the word 'hello' with 'world' only on line 144:</p>

<p>如果没写地址空间选项,后面的命令将会在所有的行上进行操作,下面的命令将替换所有行中的单词 <code>hello</code> 为 <code>world</code>(注:只会替换一行中出现的第一个 <code>hello</code> 单词):</p>

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">sed 's/hello/world/' input.txt &gt; output.txt</span></span></code></pre>
```

<p>除了使用行号,地址空间也可以包含一个正则表达式来匹配满足此表达式的行;下面的命令将会替换单词:<code>apple</code> 的行的 <code>hello</code> 为 <code>world</code>:</p>

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">sed '/apple/s/hello/world/' input.txt &gt; output.txt</span></span><span class="highlight-line"><span class="highlight-cl"></span></span></code></pre>
```

<p>一个地址范围通过两个地址指定,地址之间用 <code>,</code> 分隔;地址可以是数字,正则表达式或者是两者的混合;下面的命令仅仅在 4 到 17 行,替换单词 <code>hello</code> 为 <code>world</code>;</p>

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">sed '4,17s/hello/world/' input.txt &gt; output.txt</span></span><span class="highlight-line"><span class="highlight-cl"></span></span></code></pre>
```

<p>可以添加符号:<code>!</code> 来取地址空间的补集:<code>取反</code>(在命令字母前面加);下面的命令替换单词 <code>hello</code> 为 <code>world</code> 仅仅在那些不包含单词:<code>apple</code> 的行中:</p>

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">sed '/apple/!s/hello/world/' input.txt &gt; output.txt</span></span><span class="highlight-line"><span class="highlight-cl"></span></span></code></pre>
```

<p>The following command replaces the word 'hello' with 'world' only in lines 1 to 3 a

d 18 till the last line of the input file (i.e. excluding lines 4 to 17):</p>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> sed '4,17!s/hello/world/' input.txt &gt; output.txt
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> </code> </pre>
```

<h5 id="X">X</h5>

<p> <code>X</code> 是一个单字母命令</p>

<p>其常用的命令有:</p>

<table>

<thead>

<tr>

<th>命令</th>

<th>说明</th>

<th>示例</th>

</tr>

</thead>

<tbody>

<tr>

<td>#</td>

<td>注释</td>

<td> <code>'4,5p # 打印输出4到5行'</code> </td>

</tr>

<tr>

<td>p</td>

<td>打印匹配行</td>

<td> <code>'4,5p'</code> </td>

</tr>

<tr>

<td>s</td>

<td>对匹配行进行替换操作</td>

<td> <code>s/a/b/g</code> </td>

</tr>

<tr>

<td>=</td>

<td>显示文件行号</td>

<td> <code>sed -ne '=:p' aaa.txt</code> </td>

</tr>

<tr>

<td>d</td>

<td>删除定位行:Delete the pattern space; immediately start next cycle.</td>

<td> <code>sed -e '4,5d' aaa.txt</code> </td>

</tr>

<tr>

<td>q</td>

<td>第一个模式匹配完成后退出或立即退出</td>

<td></td>

</tr>

<tr>

<td>n</td>

<td>If auto-print is not disabled, print the pattern space, then, regardless, replace the pattern space with the next line of input. If there is no more input then <code>sed</code> exits without processing any more commands.</td>

<td>This command is useful to skip lines (e.g. process every Nth line).</td>

</tr>

```

<tr>
<td>{cmds}</td>
<td>一个命令组可以用{}包起来,这个会特别有用.当你想一组命令被一个单地址,或者是地址范围触发
</td>
<td></td>
</tr>
</tbody>
</table>
<h5 id="-options-">[options]</h5>
<p>options 选项针对不同的具体命令有具体的使用方法,也有可能不带选项;</p>
<p>一个脚本命令行或者是一个脚本文件里的命令可以使用分号进行分隔(<code>;</code>),或者是
用换行进行分隔('\n',ascii 码值为 10);多个脚本可以被同时用参数 <code>-e</code> 或者是 <code>
-f</code> 指定;</p>
<p>注:</p>
<blockquote>
<p>如上面的命令组 <code>{}</code> 的意义与区别在于,每一个命令是以[地址]X[选项]进行书写,
此这就是一个完整的命令.下一个命令与上一个命令之间由上面的知识可知是通过分号 <code>;</cod
e> 或者是换行 <code>\n</code> 进行分隔;因此多个命令看起来是这样的 <code>3,5p;6,7a//this is
a comment;</code> 这里有两个命令,如果这两个命令的地址范围都是 <code>3,5</code>,那可
合并为一个命令组:</p>
</blockquote>
<pre><code class="language-bash highlight-chroma"><span class="highlight-line"><span c
lass="highlight-cl">seq <span class="highlight-m">10</span> <span class="highlight-p">|<
span> sed -n -e <span class="highlight-s1">'3,5{a//this is a comment;
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-s1">p;
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-s1">}'</span>
</span></span></code></pre>
<h3 id="打印命令">打印命令 <code>p</code></h3>
<h4 id="打印单行">打印单行</h4>
<pre><code class="language-bash highlight-chroma"><span class="highlight-line"><span c
lass="highlight-cl"><span class="highlight-nb">echo</span> <span class="highlight-s2">"<
span><span class="highlight-nv">${var}</span><span class="highlight-s2">"</span> <span
class="highlight-p">|</span> sed -n -e <span class="highlight-s1">'5p'</span> <span clas
s="highlight-c1">#打印第五行</span>
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="high
light-nb">echo</span> <span class="highlight-s2">"</span><span class="highlight-nv">${v
r}</span><span class="highlight-s2">"</span> <span class="highlight-p">|</span> sed -n
-e <span class="highlight-s1">'5p;5q;'</span> <span class="highlight-c1">#打印第五行高效
式</span>
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="high
light-nb">echo</span> <span class="highlight-s2">"</span><span class="highlight-nv">${v
r}</span><span class="highlight-s2">"</span> <span class="highlight-p">|</span> sed -n
-e <span class="highlight-s1">'$p'</span> <span class="highlight-c1">#打印最后一行</span>
</span></span></code></pre>
<h4 id="打印多行">打印多行</h4>
<pre><code class="language-bash highlight-chroma"><span class="highlight-line"><span c
lass="highlight-cl"><span class="highlight-nb">echo</span> <span class="highlight-s2">"<
span><span class="highlight-nv">${var}</span><span class="highlight-s2">"</span> <span
class="highlight-p">|</span> sed -n -e <span class="highlight-s1">'5,8p'</span> <span cla
s="highlight-c1">#打印第5到8行</span>
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="high

```

```
ight-nb">echo</span> <span class="highlight-s2">"</span><span class="highlight-nv">$v  
r</span><span class="highlight-s2">"</span> <span class="highlight-p">|</span> sed -n  
e <span class="highlight-s1">'5,$p'</span> <span class="highlight-c1">#打印第5到最后一行  
</span>
```

```
</span></span></code></pre>
```

<h3 id="替换命令--">替换命令: <code>s</code></h3>

<blockquote>

<p><code>s</code> 命令(即替换命令)可能是 <code>sed</code> 最重要的命令,它包含非常多选项,此命令的语法为:<code>'s/regexp/replacement/flags'</code>

其基本概念特别简单:<code>s</code> 命令尝试使用命令中的 <code>正则表达式</code> 去匹配 <code>模式空间</code> 中的字符串;如果匹配成功,模式空间中的匹配项将会被 <code>replacem
nt</code> 替代</p>

</blockquote>

<p>标准命令如下, origin 是一个正则表达式, replacement 是替换后的目标字符串</p>

```
<pre><code class="language-bash highlight-chroma"><span class="highlight-line"><span c  
lass="highlight-cl"><span class="highlight-c1"># 标准命令</span>
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">sed -e <span clas  
s="highlight-s1">'s/regexp/replacement/flags'</span> filename.txt
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="high  
light-c1"># 简化命令</span>
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">sed <span class=  
highlight-s1">'s/origin/replacement/'</span> filename.txt
```

```
</span></span></code></pre>
```

<h4 id="-替换成---第一匹配组-"><code>aaa</code> 替换成 <code>bbb</code> (第一匹配组
</h4>

```
<pre><code class="language-bash highlight-chroma"><span class="highlight-line"><span c  
lass="highlight-cl"><span class="highlight-nb">echo</span> <span class="highlight-s2">"a  
a ccc bbb ddd eee"</span> <span class="highlight-p">|</span> sed -e <span class="highli  
ht-s1">'s/aaa/bbb/'</span>
```

```
</span></span></code></pre>
```

```
</span></span></code></pre>
```

<h4 id="-替换成---指定匹配组-"><code>aaa</code> 替换成 <code>bbb</code> (指定匹配组
</h4>

```
<pre><code class="language-bash highlight-chroma"><span class="highlight-line"><span c  
lass="highlight-cl"><span class="highlight-c1"># 替换第二个 'aaa' 为 'bbb'</span>
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="high  
light-nb">echo</span> <span class="highlight-s2">"aaa aaa bbb ddd eee"</span> <span cl  
ss="highlight-p">|</span> sed -e <span class="highlight-s1">'s/aaa/bbb/2'</span>
```

```
</span></span></code></pre>
```

<h4 id="-替换成---全部替换-"><code>aaa</code> 替换成 <code>bbb</code> (全部替换)</h
>

```
<pre><code class="language-bash highlight-chroma"><span class="highlight-line"><span c  
lass="highlight-cl"><span class="highlight-c1"># 替换所有 aaa 为 bbb</span>
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="high  
light-nb">echo</span> <span class="highlight-s2">"aaa aaa bbb ddd eee"</span> <span cl  
ss="highlight-p">|</span> sed -e <span class="highlight-s1">'s/aaa/bbb/g'</span>
```

```
</span></span></code></pre>
```

<h4 id="命令中的分隔符号-"><code>s</code> 命令中的分隔符号 <code>/</code></h4>

<blockquote>

<p>The <code>/</code> characters may be uniformly replaced by any other single characte
within any given <code>s</code> command. The <code>/</code> character (or whatever
other character is used in its stead) can appear in the regexp or replacement only if it is preced
d by a <code>\</code> character.</p>

</blockquote>

<p>在替换命令中(即 <code>s</code> 命令)的 <code>/</code> 是可以任意字符来分隔的:</p>

```
>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"># 最常见的形式
</span></span><span class="highlight-line"><span class="highlight-cl">echo "aaa bb cc" |
sed -e 's/a/b/g'
</span></span><span class="highlight-line"><span class="highlight-cl"># 这里用#分隔
</span></span><span class="highlight-line"><span class="highlight-cl">echo "aaa bb cc" |
sed -e 's#a#b#g'
</span></span><span class="highlight-line"><span class="highlight-cl"># 用空格分隔
</span></span><span class="highlight-line"><span class="highlight-cl">echo "aaa bb cc" |
sed -e 's a b g'
</span></span></code></pre>
<h4 id="目标串中的">目标串中的 <code>&amp;</code></h4>
<p>在字符串替换的时候,<code>&amp;</code> 代表匹配的字符串的值<br>
<strong>场景: 给电话号码添加双引号</strong></p>
<pre><code class="language-bash highlight-chroma"><span class="highlight-line"><span class="highlight-cl"><span class="highlight-nb">echo</span> <span class="highlight-s2">"p
oneNumber:13800138000"</span> <span class="highlight-p">|</span> sed -E <span class="
highlight-s1">'s/[0-9]{11}/"&amp;"/g'</span>
</span></span><span class="highlight-line"><span class="highlight-cl">phoneNumber:<s
an class="highlight-s2">"13800138000"</span>
</span></span></code></pre>
<h4 id="目标串中使用匹配组">目标串中使用匹配组</h4>
<blockquote>
<p>The replacement can contain <code>\n</code>(n being a number from 1 to 9, inclusive)
references, which refer to the portion of the match which is contained between the nth <cod
>\(</code> and its matching <code>\)</code>. Also, the replacement can contain unescap
d <code>&amp;</code> characters which reference the whole matched portion of the patte
n space.</p>
</blockquote>
<p><strong>场景: <code>phone:13800138000</code> 修改为:<code>{"phone": "13800138
00"}</code></strong></p>
<pre><code class="language-bash highlight-chroma"><span class="highlight-line"><span class="highlight-cl"><span class="highlight-cl"># 修改需要的部分</span>
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="high
ight-nb">echo</span> <span class="highlight-s2">"something*&amp;&amp;&amp;^^^ph
ne:13800138000sfjdsksfsf"</span> <span class="highlight-p">|</span> sed -E <span class="
highlight-s1">'s/(phone):([0-9]{11})/{"\1":"\2"}/g'</span>
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="high
ight-cl"># 只输出需要的部分</span>
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="high
ight-nb">echo</span> <span class="highlight-s2">"something*&amp;&amp;&amp;^^^ph
ne:13800138000sfjdsksfsf"</span> <span class="highlight-p">|</span> sed -E <span class="
highlight-s1">'s/.*(phone):([0-9]{11}).*/{"\1":"\2"}/g'</span>
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span></code></pre>
<h4 id="去除文件中的---r">去除文件中的 '\r'</h4>
<pre><code class="language-bash highlight-chroma"><span class="highlight-line"><span class="highlight-cl"><span class="highlight-nb">echo</span> -e <span class="highlight-s2">
"aaa\r\nbbb\r\nccc"</span> &gt; aaa.txt
</span></span><span class="highlight-line"><span class="highlight-cl">sed -e <span clas
="highlight-s1"> '$s/\r//g'</span> aaa.txt <span class="highlight-p">|</span> od -c
</span></span></code></pre>
<h3 id="行合并">行合并</h3>
```

<blockquote>

<p>This section uses <code>N</code>, <code>D</code> and <code>P</code> commands to process multiple lines, and the <code>b</code> and <code>t</code> commands for br nching</p>

</blockquote>

<h4 id="合并特定行">合并特定行</h4>

<pre> <code class="highlight-chroma"> \$ cat lines.txt

 hello

 hel

 lo

 hello

 \$ sed '2{N;s/\n//;}' ines.txt

 hello

 hello

 hello

 </code> </pre>

<p>命令解释:</p>

<blockquote>

<p><code>2{N;s/\n//;}</code> 匹配上第二行后,执行命令块:<code>{N;s/\n//;}</code>,命令块再读取下一行(命令 <code>N</code>),执行 <code>s</code> 命令;搜索 <code>\n</code> 替为空 <code></code>,这样 2-3 行合并在一行.其它行没有匹配上.直接原样输出;</p>

</blockquote>

<h4 id="合并换行的多行">合并 <code>'\n'</code> 换行的多行</h4>

<pre> <code class="language-bash highlight-chroma"> \$ cat 1.txt

 this \

 \

 \

 \

 \

 and another \

 \

 \$ sed -e ':x /\n/ { N; s/\n//g ; bx }' 1.txt

 this is a long line

 and another line

 #TODO: The above requires gnu sed.

 # non-gnu seds need newlines after ':' and 'b'

 </code> </pre>

<p>命令解释:</p>

```

<blockquote>
<p><code>x /\$/ { N; s/\n//g ; bx }</code> </p>
</blockquote>
<ol>
<li>首先定义一个标号 <code>x</code>;</li>
<li>匹配以 <code>\</code> 结尾的行,由于 <code>\</code> 是特殊字符.因此需要转义 <code>
\</code>,同时只匹配以 <code>\</code> 结尾的行,因此 <code>\</code> 后面加了一个 <cod
>$</code>;</li>
<li>搜索到行后,执行命令块:<code>{ N; s/\n//g ; bx }</code>;</li>
<li>以上命令块是一个循环,首先读取下一行.同上,替换 <code>\n</code> 为空;然后无条件跳转 <c
de>b</code> 到标签 <code>x</code>;<br>
5.此时再匹配当前行是否以 <code>\</code> 结尾;注意此时行已经发生变化,读取了下一行;</li>
</ol>
<h4 id="文本多行转一行-并且以分隔">文本多行转一行,并且以 <code>,</code> 分隔</h4>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">### 无法实现
</span></span><span class="highlight-line"><span class="highlight-cl">echo -e "aaa\nbb
\nccc" &gt; aaa.txt
</span></span><span class="highlight-line"><span class="highlight-cl">sed -e 's/\n/,/g' a
a.txt
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span></code></pre>
<p>以上代码无法执行的原因是可以参考:<a href="https://ld246.com/forward?goto=https%3A%
F%2Fblog.csdn.net%2F1qglqglqg%2Farticle%2Fdetails%2F44458625" target="_blank" rel="no
ollow ugc">sed 列转行无法返回是因为其处理机制的问题</a> </p>
<blockquote>
<p>说明: sed 默认只按行处理, N 可以让其读入下一行, 再对\n 进行替换, 这样就可以将两行并
一行。但是怎么将所有行并作一行呢? 可以采用 sed 的跳转功能。:a 在代码开始处设置一个标记 a,
代码执行到结尾处时利用跳转命令 t a 重新跳转到标号 a 处, 重新执行代码, 这样就可以递归的将所
行合并成一行。 </p>
</blockquote>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">sed ':a ; $!N; s/\n/,/ ; t a ; ' aaa.txt
</span></span></code></pre>
<p>命令解释:</p>
<table>
<thead>
<tr>
<th>命令片断</th>
<th>说明</th>
<th>作用</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>:a</code> </td>
<td>定义一个标签</td>
<td>用于控制后面的循环</td>
</tr>
<tr>
<td><code>$!N</code> </td>
<td><code>$</code> 表示最后一行,<code>$!</code> 表示非最后一行,<code>N</code>
示向 <code>模式空间中追加下一行</code> </td>
<td>非尾行时追加一行数据到 <code>模式空间</code> </td>

```



```

</tr>
<tr>
<td><code>s\n/,/ ;</code></td>
<td>执行 <code>s</code> 替换命令,把拼接的两行中 <code>\n</code> 替换为 <code>,</code></td>
<td>完成两行合并,去除换行</td>
</tr>
<tr>
<td><code>t a ;</code></td>
<td>如果最后一次输入的最后一个 s/// 子命令执行成功,跳转到标签 <code>a</code>,否则跳转到束</td>
<td>控制循环,直到读取到最后一行</td>
</tr>
</tbody>
</table>
<blockquote>
<p>b label ,无条件跳转到标签 label,如果 label 没有指定,跳转到命令的结尾<br>
t label ,如果最后一次输入的最后一个 s/// 子命令执行成功,跳转到标签 label,如果 label 没有指定,跳到命令的结尾</p>
</blockquote>
<ol>
<li><a href="https://ld246.com/forward?goto=https%3A%2F%2Fblog.csdn.net%2F1qglqglq%2Farticle%2Fdetails%2F44458625" target="_blank" rel="nofollow ugc">sed 命令换行符问题及跳转命令-对文件处理, 整个文件先转换为一行处理完后再把换行加上</a></li>
<li><a href="https://ld246.com/forward?goto=http%3A%2F%2Fbbs.chinaunix.net%2Fthread4155252-1-1.html" target="_blank" rel="nofollow ugc">sed 中跳转命令的用法</a></li>
<li><a href="https://ld246.com/forward?goto=http%3A%2F%2Fbbs.chinaunix.net%2Fforum.hp%3Fmod%3Dviewthread%26tid%3D1762006%26fromuid%3D26669282" target="_blank" rel="nofollow ugc">文本编辑的一点心得--sed 篇</a></li>
<li><a href="https://ld246.com/forward?goto=http%3A%2F%2Fbbs.chinaunix.net%2Fthread336126-1-1.html" target="_blank" rel="nofollow ugc">抛砖引玉----翻译加注 sed1line</a></li>

<li><a href="https://ld246.com/forward?goto=https%3A%2F%2Fsegmentfault.com%2Fa%2F190000015921775" target="_blank" rel="nofollow ugc">Linux 命令之 grep/sed/awk 等行转列</a></li>
<li><a href="https://ld246.com/forward?goto=http%3A%2F%2Fman.linuxde.net%2Fsed" target="_blank" rel="nofollow ugc">Linux 命令大全-sed</a></li>
<li><a href="https://ld246.com/forward?goto=https%3A%2F%2Fwww.yiibai.com%2Fsed%2Fed_loops.html" target="_blank" rel="nofollow ugc">sed 循环</a></li>
<li><a href="https://ld246.com/forward?goto=https%3A%2F%2Fmy.oschina.net%2Fshelllife%2Fblog%2F118337" target="_blank" rel="nofollow ugc">sed 命令替换换行符(同 1)</a></li>
<li><a href="https://ld246.com/forward?goto=https%3A%2F%2Fwww.cnblogs.com%2Fctaix%2Fp%2F5860221.html" target="_blank" rel="nofollow ugc">sed 命令详解</a></li>
<li><a href="https://ld246.com/forward?goto=https%3A%2F%2Fwww.cnblogs.com%2Frenpig%2Fp%2F9533507.html" target="_blank" rel="nofollow ugc">sed 常用命令</a></li>
<li><a href="https://ld246.com/forward?goto=https%3A%2F%2Fwww.cnblogs.com%2FBinBiStory%2Fp%2F7282789.html" target="_blank" rel="nofollow ugc">Sed 命令的使用详细解释</a></li>
</ol>

```