

linux shell 之 字符处理命令 二

作者: [c2y2](#)

原文链接: <https://ld246.com/article/1395939527327>

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>
  <strong>1, tr 命令</strong>
</p>
<p>
  哈哈, tr? ? ? table里边的tr吗, :-) 想多了, 不够就是处理行的哦
</p>
<p>
  <strong>tr:</strong> tr 可以用来删除一段讯息当中的文字, 或者是进行文字讯息的替换!
</p>
<p>
  使用方法:
</p>
<pre class="prettyprint lang-bsh">用法: tr [选项]... SET1 [SET2]
从标准输入中替换、缩减和/或删除字符, 并将结果写到标准输出。
<p>-c, -C, --complement    首先补足 SET1<br>
-d, --delete            删除匹配 SET1 的内容, 并不作替换<br>
-s, --squeeze-repeats  如果匹配于 SET1 的字符在输入序列中存在连续的<br>
重复, 在替换时会被统一缩为一个字符的长度<br>
-t, --truncate-set1    先将 SET1 的长度截为和 SET2 相等<br>
--help                显示此帮助信息并退出<br>
--version              显示版本信息并退出</p>
<p>SET 是一组字符串, 一般都可按照字面含义理解。解析序列如下: </p>
<p>\NNN 八进制值为 NNN 的字符(1 至 3 个数位)<br>
\    反斜杠<br>
\a    终端鸣响<br>
\b    退格<br>
\f    换页<br>
\n    换行<br>
\r    回车<br>
\t    水平制表符<br>
\v    垂直制表符<br>
字符 1-字符 2 从字符 1 到字符 2 的升序递增过程中经历的所有字符<br>
[字符*] 在 SET2 中适用, 指定字符会被连续复制直到吻合设置 1 的长度<br>
[字符*次数] 对字符执行指定次数的复制, 若次数以 0 开头则被视为八进制数<br>
[:alnum:] 所有的字母和数字<br>
[:alpha:] 所有的字母<br>
[:blank:] 所有呈水平排列的空白字符<br>
[:cntrl:] 所有的控制字符<br>
[:digit:] 所有的数字<br>
[:graph:] 所有的可打印字符, 不包括空格<br>
[:lower:] 所有的小写字母<br>
[:print:] 所有的可打印字符, 包括空格<br>
[:punct:] 所有的标点字符<br>
[:space:] 所有呈水平或垂直排列的空白字符<br>
[:upper:] 所有的大写字母<br>
[:xdigit:] 所有的十六进制数<br>
[=字符=] 所有和指定字符相等的字符</p>
<p>仅在 SET1 和 SET2 都给出, 同时没有-d 选项的时候才会进行替换。<br>
仅在替换时才可能用到-t 选项。如果需要 SET2 将被通过在末尾添加原来的末字符的方式<br>
补充到同 SET1 等长。SET2 中多余的字符将被省略。只有[:lower:] 和[:upper:]<br>
以升序展开字符; 在用于替换时的 SET2 中以成对表示大小写转换。-s 作用于 SET1, 既不<br>
替换也不删除, 否则在替换或展开后使用 SET2 缩减。<br>
</p></pre>
<p>例子: 将内容由小写转为大写</p>

```

```
<pre class="prettyprint lang-bsh">root@c2y2:/home/exp/vodxunlei/source# cat 1.txt |head -
```

```
1410cf3882936ef6717bef14db891d8f4e6a887e449
141f5a588b071d85924bbf82bcf1cc875f905eebd8e
root@c2y2:/home/exp/vodxunlei/source# cat 1.txt |head -2|tr "[:lower:]" "[:upper:]"
1410CF3882936EF6717BEF14DB891D8F4E6A887E449
141F5A588B071D85924BBF82BCF1CC875F905EEBD8E
root@c2y2:/home/exp/vodxunlei/source#&nbsp;&nbsp;&nbsp;</pre>
```

```
<pre class="prettyprint lang-bsh"><strong>2,join 命令</strong></pre>
```

join:针对每一对具有相同内容的输入行，整合为一行写到标准输出

</p>

<p>

用法:

</p>

```
<pre class="prettyprint lang-bsh">用法: join [选项]... 文件1 文件2
```

针对每一对具有相同内容的输入行，整合为一行写到标准输出，默认的内容连接区块是由第一个空白符代表的分界符号。当文件1或文件2 都被指定为 "-" 时，程序将从标准输入读取数据。

<p>-a 文件编号 文件编号的值可以是 1 或 2，分别对应文件 1 和 文件 2。

此选项用于根据指定文件编号输出不成对的行目。

-e 字符 将缺失的输入区块替换为指定字符

-i, --ignore-case 比较时忽略大小写

-j 域 等于"-1 域 -2 域"

-o 格式 按照指定格式构造输出行

-t 字符 使用指定字符作为输入和输出的分隔符

-v 文件编号 类似 -a 文件编号，但禁止组合输出行

-1 域 在文件 1 的此域组合

-2 域 在文件 2 的此域组合

--check-order 检查输入行是否正确排序，即使所有输入行均是成对的

--nocheck-order 不检查输入是否正确排序

--header 将首行视作域的头部，直接输出而不对其进行匹配

--help 显示此帮助信息并退出

--version 显示版本信息并退出</p>

<p>除非使用了"-t 字符串" 选项，否则前导空格分隔的域将被忽略，如果指定了字符串，
则使用指定字符串分隔任意的域并从 1 开始计数的域编号。可以指定的格式是由一个
或多个逗号活空格所分隔的描述，其形式为"文件编号.域"或者"0"。默认的
格式输出合并后的域、文件 1 和文件 2 剩下的域，均由该指定字符串分隔。</p>

<p>重要提示：文件 1 和文件 2 必须在合并域中排序。

例如，如果"join"后没有选项，使用"sort -k 1b,1"。

注意，所进行的比较遵从"LC_COLLATE"所指定的的规则。

如果输入没有被排序并导致某些行无法合并，将会显示警告信息。</p></pre><p></p>

<p>

例子:

</p>

```
<pre class="prettyprint lang-bsh">root@c2y2:/home/exp/vodxunlei/source# cat /etc/passwd
head -2 &gt; passwd
```

```
root@c2y2:/home/exp/vodxunlei/source# cat /etc/shadow|head -2 &gt; shadow
```

```
root@c2y2:/home/exp/vodxunlei/source# cat passwd
```

```
root:x:0:0:root:/root:/bin/bash
```

```
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
```

```
root@c2y2:/home/exp/vodxunlei/source# cat shadow
```

```
root:$6$CsRWrpV$b0Kmabhimd69oV46/MbsL64HNQMoZnh3g6lWu/Z5YKFsp20btmt4YGG/
EB4kVolt.mKmZzcX0LbEcljuo90.:15912:0:99999:7:::
```

```
daemon:*.15820:0:99999:7:::
root@c2y2:/home/exp/vodxunlei/source# join -t ":" passwd shadow
root:x:0:0:root:/root:/bin/bash: <span>$6$CsRWpeV$b0Kmabhimd69oV46/MbsL64HNQMoZ
h3g6lWu/Z5YKFsp20btmt4YGG/GEB4kVlt.mKmZzcX0LbEcljuo90.</span> <span>:15912:0:99
99:7:::</span> daemon:x:1:1:daemon:/usr/sbin:/bin/sh:*.15820:0:99999:7:::
```

</pre>

<p>

3, paste

</p>

<p>

paste:将每个文件的同行粘贴在一起

</p>

<p>

用法:

</p>

<p>

</p>

```
<pre class="prettyprint lang-bsh">用法: paste [选项]... [文件]...
```

将每个指定文件里的每一行整合到对应一行里写到标准输出，之间用制表符分隔。

如果没有指定文件，或指定文件为"-", 程序将从标准输入读取数据。

<p>长选项必须使用的参数对于短选项时也是必需使用的。

-d, --delimiters=列表 改用指定列表里的字符替代制表分隔符

-s, --serial 不使用平行的行目输出模式，而是每个文件占用一行

--help 显示此帮助信息并退出

--version 显示版本信息并退出

</p> </pre>

例子: <p> </p>

```
<pre class="prettyprint lang-bsh">root@c2y2:/home/exp/vodxunlei/source# paste passwd s
adow
```

```
root:x:0:0:root:/root:/bin/bash root:<span>$6$CsRWpeV$b0Kmabhimd69oV46/MbsL64H
QMoZnh3g6lWu/Z5YKFsp20btmt4YGG/GEB4kVlt.mKmZzcX0LbEcljuo90</span> <span>:15
12:0:99999:7:::</span> daemon:x:1:1:daemon:/usr/sbin:/bin/sh daemon:*.15820:0:99999:7:::
```

</pre>

<p>

4, split分割命令

</p>

<p>

split: 对文件进行切割，很使用，特别是有的文本特别大时，如大日字文件

</p>

<p>

用法:

</p>

<p>

</p>

```
<pre class="prettyprint lang-bsh">用法: split [选项]... [输入 [前缀]]
```

将输入内容拆分为固定大小的分片并输出到"前缀aa"、"前缀ab",...;

默认以 1000 行为拆分单位，默认前缀为"x"。如果不指定文件，或者文件为"-", 则从标准输入读取数据。

<p>长选项必须使用的参数对于短选项时也是必需使用的。

-a, --suffix-length=N use suffixes of length N (default 2)

-b, --bytes=SIZE put SIZE bytes per output file

-C, --line-bytes=SIZE put at most SIZE bytes of lines per output file

-d, --numeric-suffixes use numeric suffixes instead of alphabetic
 -e, --elide-empty-files do not generate empty output files with `-n' --filter=COMMA D write to shell COMMAND; file name is $FILE -l, --lines=NUMBER put NUMBER lines per output file -n, --number=CHUNKS generate CHUNKS output files. See below -u, --unbuffered immediately copy input to output with -n r/...'
`

--verbose 在每个输出文件打开前输出文件特征

--help 显示此帮助信息并退出

--version 显示版本信息并退出</p>

<p>SIZE 可以是一个可选的整数，后面跟着以下单位中的一个：

 KB 1000, K 1024, MB 10001000, M 10241024, 还有 G、T、P、E、Z、Y。 </p>

<p>块数可能为：

N 基于输入的大小分割为 N 个文件

K/N 将 N 个文件中的第 K 个输出到标准输出

l/N 分割为 N 个文件，但分割时不切断行

l/K/N 将 N 个文件中的第 K 个输出到标准输出，但分割时不切断行

r/N 类似 l 但使用轮询分配

r/K/N 类似 r/N 但尽将 N 个文件中的第 K 个输出到标准输出

</p></pre>

例子：拆分图片，并合并图片哈，合并后的完全能用哦。 <p></p>

<p>

</p>

<p>

</p><pre class="prettyprint lang-bsh">root@c2y2:/home/exp/Pictures/temp# ls -al

总用量 2336

drwxr-xr-x 2 root root 4096 3月 28 00:44 .

drwxr-xr-x 3 exp exp 4096 3月 28 00:44 ..

-rw----- 1 root root 2382720 3月 28 00:44 download.png

root@c2y2:/home/exp/Pictures/temp# split -b 1000k download.png download.png

root@c2y2:/home/exp/Pictures/temp# ls

download.png download.pngaa download.pngab download.pngac

root@c2y2:/home/exp/Pictures/temp# ls -al

总用量 4664

drwxr-xr-x 2 root root 4096 3月 28 00:44 .

drwxr-xr-x 3 exp exp 4096 3月 28 00:44 ..

-rw----- 1 root root 2382720 3月 28 00:44 download.png

-rw-r--r-- 1 root root 1024000 3月 28 00:44 download.pngaa

-rw-r--r-- 1 root root 1024000 3月 28 00:44 download.pngab

-rw-r--r-- 1 root root 334720 3月 28 00:44 download.pngac

root@c2y2:/home/exp/Pictures/temp# rm download.png -rf

root@c2y2:/home/exp/Pictures/temp# cat download.pngaa download.pngab download.pnga

> download.png

</pre>

5, -管线命令

<p></p>

<p>

-:将其作为前一个的标准输出，并作为后面一个命令的标准输入

</p>

<p>

例子：我将 /home 里面的档案给他打包，但打包的数据不是纪录到档案，而是传送到 std ut; 经过管线后，将 tar -cvf - /home 传送给后面的 tar -xvf -

</p>

<p>

</p><pre class="prettyprint lang-bsh">tar -cvf - /home | tar -xvf -</pre>

```
<p></p>
```

```
<p>
```

```
  后记: :-), 一天一天多一点。。。
```

```
</p>
```

```
<p>
```

```
  <span></span>
```

```
</p>
```