

## 第12章 tr 用法

### 12.1 关于tr

tr用来从标准输入中通过替换或删除操作进行字符转换。tr主要用于删除文件中控制字符或进行字符转换。使用tr时要转换两个字符串：字符串1用于查询，字符串2用于处理各种转换。tr刚执行时，字符串1中的字符被映射到字符串2中的字符，然后转换操作开始。

本章内容有：

- 大小写转换。
- 去除控制字符。
- 删除空行。

带有最常用选项的tr命令格式为：

```
tr -c-d-s ["string1_to_translate_from"] ["string2_to_translate_to"]  
file
```

这里：

- c 用字符串1中字符集的补集替换此字符集，要求字符集为 ASCII。
- d 删除字符串1中所有输入字符。
- s 删除所有重复出现字符序列，只保留第一个；即将重复出现字符串压缩为一个字符串。

Input-file是转换文件名。虽然可以使用其他格式输入，但这种格式最常用。

#### 12.1.1 字符范围

使用tr时，可以指定字符串列表或范围作为形成字符串的模式。这看起来很像正则表达式，但实际上不是。指定字符串1或字符串2的内容时，只能使用单字符或字符串范围或列表。

[a-z] a-z内的字符组成的字符串。

[A-Z] A-Z内的字符组成的字符串。

[0-9] 数字串。

/octal 一个三位的八进制数，对应有效的 ASCII 字符。

[O\*n] 表示字符O重复出现指定次数n。因此[O\*2]匹配OO的字符串。

大部分tr变种支持字符类和速记控制字符。字符类格式为[:class]，包含数字、希腊字母、空行、小写、大写、cntrl键、空格、点记符、图形等等。表 12-1包括最常用的控制字符的速记方式及三位八进制引用方式。

当用一个单字符替换一个字符串或字符范围时，注意字符并不放在方括号里（[]）。一些系统也可以使用方括号，例如可以写成["\012"]或"\012"，tr也允许不加引号，因此命令中看到单引号而不是双引号时也不要感到奇怪。

像大多数系统工具一样，tr也受特定字符的影响。因此如果要匹配这些字符，需使用反斜

线屏蔽其特殊含义。例如，用 `\{` 指定花括号左边可以屏蔽其特殊含义。

表12-1 tr中特定控制字符的不同表达方式

速记符	含义	八进制方式
<code>\a</code>	Ctrl-G 铃声	<code>\007</code>
<code>\b</code>	Ctrl-H 退格符	<code>\010</code>
<code>\f</code>	Ctrl-L 走行换页	<code>\014</code>
<code>\n</code>	Ctrl-J 新行	<code>\012</code>
<code>\r</code>	Ctrl-M 回车	<code>\015</code>
<code>\t</code>	Ctrl-I tab键	<code>\011</code>
<code>\v</code>	Ctrl-X	<code>\030</code>

### 12.1.2 保存输出

要保存输出结果，需将之重定向到一个文件。下面的例子重定向输出到文件 `results.txt`。输入文件是 `cops.txt`。

```
$ tr -s "[a-z]"< cops.txt >results.txt
```

现在看一些例子。

### 12.1.3 去除重复出现的字符

下面文件包含了一些打印错误。这种情况时常发生，例如在 `vi`编辑器中，偶尔按住一个键不放。

```
$ pg oops.txt
And the cowwwwws went homeeeeeeee
Or did theyyyy
```

如果要去掉重复字母或将其压缩在一起，使用 `-s`选项。因为都是字母，故使用 `[a-z]`。输入文件重定向到 `tr`命令。

```
$ tr -s "[a-z]"< oops.txt
And the cows went home
Or did they
```

所有重复字符被压缩成一个。如果使用 `cat`命令，再将结果管道输出至 `tr`，结果是一样的。

```
$ cat oops.txt | tr -s "[a-z]"
And the cows went home
Or did they
```

### 12.1.4 删除空行

要删除空行，可将之剔出文件。下面是一个文件 `plane.txt`。文本间有许多空行。

```
$ pg plane.txt
987932 Spitfire
```

```
190992 Lancaster
```

```
238991 Typhoon
```

使用-s来做这项工作。换行的八进制表示为\012，命令为：

```
$ tr -s "[\012]" < plane.txt
987932 Spitfire
190992 Lancaster
238991 Typhoon
```

也可以使用换行速记方式\n，这里用单引号（通常用双引号）。

```
$ tr -s ["\n"] < plane.txt
987932 Spitfire
190992 Lancaster
238991 Typhoon
```

### 12.1.5 大写到小写

除了删除控制字符，转换大小写是 tr最常用的功能。为此需指定即将转换的小写字符 [a-z] 和转换结果 [A-Z]。

第一个例子，tr从一个包含大小写字母的字符串中接受输入。

```
$ echo "May Day, May Day, Going Down.." | tr "[a-z]" "[A-Z]"
MAY DAY, MAY DAY, GOING DOWN..
```

同样，也可以使用字符类[:lower:]和[:upper:]。

```
$ echo "May Day, May Day, Going Down.." | tr "[:lower:]" "[:upper:]"
MAY DAY, MAY DAY, GOING DOWN..
```

将文本文件大写转换为小写并输出至一个新文件，格式为：

```
cat file-to-translate | tr "[A-Z]" "[a-z]" > new-file-name
```

这里file-to-translate保存即将转换的文件，new-file-name为保存结果的新文件名。例如：

```
cat myfile | tr "[A-Z]" "[a-z]" > lower_myfile
```

### 12.1.6 小写到大写

转换小写到大写与上一节大写到小写过程刚好相反。以下有两个例子：

```
$ echo " Look for the route, or make the route" | tr "[a-z]" "[A-Z]"
LOOK FOR THE ROUTE, OR MAKE THE ROUTE
```

```
$ echo "May Day, May Day, Going Down.." | tr "[:upper:]" "[:lower:]"
may day, may day, going down..
```

将文本文件从小写转换为大写并将结果存入一个新文件，格式为：

```
cat file-to-translate | tr "[a-z]" "[A-Z]" > new-file-name
```

file-to-translate保存即将转换的文件，new-file-name保存结果文件，例如：

```
cat myfile | tr "[a-z]" "[A-Z]" > upper_myfile
```

### 12.1.7 删除指定字符

偶尔会从下载文件中删除只包含字母或数字的列。需要结合使用 -c和-s选项完成此功能。

下面的文件包含一个星期的日程表。任务是从其中删除所有数字，只保留日期。日期有大写，也有小写格式。因此需指定两个字符范围 [a-z]和[A-Z]，命令tr -cs "[a-z][A-Z]" "\012\*"将

文件每行所有不包含在 [a-z] 或 [A-Z] (所有希腊字母) 的字符串放在字符串 1 中并转换为一新行。-s 选项表明压缩所有新行，-c 表明保留所有字母不动。原文件如下，后跟 tr 命令：

```
$ pg diary.txt
monday 10:50
Tuesday 15:30
wednesday 15:30
thursday 10:30
Friday 09.20
$ tr -cs "[a-z][A-Z]" "[\012*]" < diary.txt
monday
Tuesday
wednesday
thursday
Friday
```

### 12.1.8 转换控制字符

tr 的第一个功能就是转换控制字符，特别是从 dos 向 UNIX 下载文件时，忘记设置 ftp 关于回车换行转换的选项时更是如此。

下面是故意没有设置转换开关的一个文本文件，是关于文具需求的一部分内容。使用 cat -v 显示控制字符。

```
$ cat -v stat.tr
Boxes paper^^^^^12^M
Clips metal^^^^^50^M
Pencils-medium^^^^^10^M
^Z
```

猜想 ‘^^^^^’ 是 tab 键。每一行以 Ctrl-M 结尾，文件结尾 Ctrl-Z，以下是改动方法。

使用 -s 选项，查看 ASCII 表。^ 的八进制代码是 136，^M 是 015，tab 键是 011，^Z 是 032，下面将按步骤完成最终功能。

用 tab 键替换 ^^^^^，命令为 "\136" "[\011\*]"。将结果重定向到临时工作文件 stat.tmp。

```
$ tr -s "[\136]" "[\011*]" < stat.tr >stat.tmp
Boxes paper 12^M
Clips metal 50^M
Pencils-medium 10^M
^Z
```

用新行替换每行末尾的 ^M，并用 \n 去除 ^Z，输入要来自于临时工作文件 stat.tmp。

```
$ tr -s "[\015\032]" "\n" < stat.tmp
Boxes paper    12
Clips metal    50
Pencils-medium 10
```

最后去除所有的控制字符，文件就可以使用了。

### 12.1.9 快速转换

如果需要删除文件中 ^M，并代之以换行。使用命令：

```
$ tr -s "[\015]" "\n" < input_file
```

或者用下述命令得同样结果。

```
$ tr -s "[\r]" "[\n]" < input_file
```

也可以用下述命令：

```
$ tr -s "\r" "\n" < input_file
```

另一个一般的Dos到UNIX转换是命令：

```
$ tr -s "[\015\032]" "[\012*]" < input_file
```

将删除所有^M和^Z，代之以换行。

要删除所有的tab键，代之以空格，使用命令：

```
$ tr -s "[\011]" "[\040*]" < input_file
```

替换passwd文件中所有冒号，代之以tab键，可以增加可读性。将冒号引起来，指定替换字符串中tab键八进制值011，下面是passwd文件，后跟tr命令结果：

```
$ pg passwd
halt:*:7:0:halt:/sbin:/sbin/halt
mail:*:8:12:mail:/var/spool/mail:
news:*:9:13:news:/var/spool/news:
uucp:*:10:14:uucp:/var/spool/uucp:
```

```
$ tr -s "[:]" "[\011]" < passwd
halt * 7 0 halt /sbin /sbin/halt
mail * 8 12 mail /var/spool/mail
news * 9 13 news /var/spool/news
uucp * 10 14 uucp /var/spool/uucp
```

或者用下述命令得同样结果。这里使用tab键的速记符。

```
$ tr "[:]" "[\t]" < passwd
```

### 12.1.10 匹配多于一个字符

可以使用[character\*n]格式匹配多于一个字符。下述文件列出系统硬盘信息，其中包含了系统已经注册的和未识别的。第一列是数字，如果不全是0，表明第二列相应硬盘已经注册。

有时全部为0看起来很烦人，找个吸引人注意力的符号来代替它，以便一眼就能看出哪个硬盘已注册，哪个不可识别。原文件如下：

```
$ pg hdisk.txt
1293 hdisk3
4512 hdisk12
0000 hdisk5
4993 hdisk12
2994 hdisk7
```

从文件列表中知道，有一个硬盘未注册，因此用星号代替所有的0。模式为[0\*4]，意即匹配至少4个0，替换字符串为星号，过滤命令及结果如下：

```
$ tr "[0*4]" "*" < hdisk.txt
1293 hdisk3
4512 hdisk12
**** hdisk5
4993 hdisk12
2994 hdisk7
```

现在从文件中可以直接看出哪个未注册。

## 12.2 小结

tr主要用于字符转换或者抽取控制字符。本章所有功能都可以用sed来完成，但有些人宁愿使用tr，因为tr更加快捷、容易。