

# 15 文本处理基础命令

## 文本处理基础命令

Linux文本处理命令是Shell编程中的常用命令，文本处理包含对文件记录的排序、文件的合并和分割等。常用命令如含 `cut` 命令、`sort` 命令、`uniq` 命令、`wc` 命令、`tr` 命令、`tee` 命令、`paste` 命令等，这些命令能实现对文件记录提取、排序、统计、合并、粘贴、转换等功能，它们与 `sed` 和 `awk` 构成了Linux文本处理的工具集。

### 一、使用cut截取文本

顾名思义，`cut` 就是截取的意思，它处理的对象是**一行文本**，可从中选取出用户所需要的部分。在有特定的分隔符时，可以指定**分隔符**，然后打印出以分隔符隔开的具体某**一列（字符或字段）**或某几列。

`cut` 命令的语法格式为：`cut [选项] [文件]`

`cut` 命令的常用选项如下。

选项	含义
<code>-c 范围</code>	以字符为单位进行分割，仅显示行中指定范围的字符
<code>-d 分隔符</code>	自定义分隔符，默认为制表符 <code>TAB</code>
<code>-f 范围</code>	显示指定字段的内容，与 <code>-d</code> 一起使用
<code>--complement</code>	提取指定字符（字段）之外的字符（字段）
<code>--output-delimiter=新分隔符</code>	指定输出内容的分隔符

`-c`、`-f` 选项使用相同的**范围表达式**。

范围表达式	含义
-------	----

<code>n</code>	第 <code>n</code> 个字符、字段。
<code>m,n</code>	第 <code>m</code> 和第 <code>n</code> 个字符、字段。
<code>m-n</code>	从第 <code>m</code> 个字符、字段到第 <code>n</code> 个（包括 <code>n</code> 在内）字符、段。
<code>-n</code>	从第 <code>1</code> 个字符、字段到第 <code>n</code> 个（包括 <code>n</code> 在内）字符、字段。
<code>n-</code>	从第 <code>n</code> 个字符、字段到结尾。

▼

Shell | 复制代码

```

1 [root@shell ~]# cat /etc/hosts
2 127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdoma
   in4
3 ::1         localhost localhost.localdomain localhost6 localhost6.localdoma
   in6

```

案例： `-c` 选项——以字符为单位截取文本

```
1  # 截取每行第1个字符
2  [root@shell ~]# cut -c1 /etc/hosts
3  1
4  :
5  # 截取每行第1和第3个字符
6  [root@shell ~]# cut -c1,3 /etc/hosts
7  17
8  :1
9  # 截取每行第2-3个字符
10 [root@shell ~]# cut -c2-3 /etc/hosts
11 27
12 :1
13 # 截取每行第2至最后一个字符
14 [root@shell ~]# cut -c2- /etc/hosts
15 27.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdoma
    in4
16 :1          localhost localhost.localdomain localhost6 localhost6.localdoma
    in6
17 # 截取每行第1个字符至第2个字符
18 [root@shell ~]# cut -c-2 /etc/hosts
19 12
20 ::
```

案例： **-d** 和 **-f** 选项——指定分隔符截取字段

```
1  # 以空格为分隔符分割字段，输出第1个字段
2  [root@shell ~]# cut -d' ' -f1 /etc/hosts
3  127.0.0.1
4  ::1
5  # 以:为分隔符分割字段，输出第1个字段
6  [root@shell ~]# cut -d':' -f1,3 /etc/passwd
7  root:0
8  bin:1
9  daemon:2
10 adm:3
11 lp:4
12 sync:5
13 shutdown:6
14 halt:7
15 mail:8
16 operator:11
17 games:12
18 ftp:14
19 nobody:99
20 systemd-network:192
21 dbus:81
22 polkitd:999
23 sshd:74
24 postfix:89
```

案例： `--complement` 选项——提取指定字符（字段）之外的字符（字段）

```
1  # 以:为分隔符分割字段, 输出第1个字段
2  [root@shell ~]# cut -d':' -f2- /etc/passwd --complement
3  root
4  bin
5  daemon
6  adm
7  lp
8  sync
9  shutdown
10 halt
11 mail
12 operator
13 games
14 ftp
15 nobody
16 systemd-network
17 dbus
18 polkitd
19 sshd
20 postfix
21 # 输出第1个字符
22 [root@shell ~]# cut -c2- /etc/passwd --complement
23 r
24 b
25 d
26 a
27 l
28 s
29 s
30 h
31 m
32 o
33 g
34 f
35 n
36 s
37 d
38 p
39 s
40 p
```

案例: `--output-delimiter` 选项——指定输出的分隔符

```

1 [root@shell ~]# cut -d':' -f1-2 /etc/passwd --output-delimiter=*
2 root*x
3 bin*x
4 daemon*x
5 adm*x
6 lp*x
7 sync*x
8 shutdown*x
9 halt*x
10 mail*x
11 operator*x
12 games*x
13 ftp*x
14 nobody*x
15 systemd-network*x
16 dbus*x
17 polkitd*x
18 sshd*x
19 postfix*x

```

`cut` 命令可以灵活地提取文本文件中的内容，它默认将提取的内容放到标准输出上，如果要将提取的内容保存到文件，可以使用文件重定向来实现。

## 二、使用sort排序

`sort` 命令对文件进行**排序**，并将排序结果标准输出。

`sort` 命令既可以从特定的文件，也可以从标准输入中获取输入。

`sort` 将文件/文本的**每一行**作为一个单位，相互比较，比较原则是**从首字符向后，依次按ASCII码值进行比较**，最后将他们按**升序**输出。

`sort` 命令的语法格式为： `sort [选项] [文件]`

`sort` 命令的常用选项如下。

选项	含义
<code>-n</code>	依照数值的大小排序。
<code>-o &lt;输出文件&gt;</code>	将排序后的结果存入制定的文件。
<code>-t &lt;分隔字符&gt;</code>	指定排序时所用的栏位分隔字符。

<code>-k</code>	指定需要排序的栏位。
<code>-r</code>	以相反的顺序来排序。
<code>-u</code>	去重重复行。
<code>-b</code>	忽略每行前面开始出的空格字符。
<code>-c</code>	检查文件是否已经按照顺序排序。
<code>-d</code>	排序时，处理英文字母、数字及空格字符外，忽略其他的字符。
<code>-f</code>	排序时，将小写字母视为大写字母。
<code>-i</code>	排序时，除了040至176之间的ASCII字符外，忽略其他的字符。
<code>-m</code>	将几个排序号的文件进行合并。
<code>-M</code>	将前面3个字母依照月份的缩写进行排序。

▼

Shell | 复制代码

```

1 [root@shell ~]# cat /etc/passwd
2 root:x:0:0:root:/root:/bin/bash
3 bin:x:1:1:bin:/bin:/sbin/nologin
4 daemon:x:2:2:daemon:/sbin:/sbin/nologin
5 adm:x:3:4:adm:/var/adm:/sbin/nologin
6 lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
7 sync:x:5:0:sync:/sbin:/bin/sync
8 shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
9 halt:x:7:0:halt:/sbin:/sbin/halt
10 mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
11 operator:x:11:0:operator:/root:/sbin/nologin
12 games:x:12:100:games:/usr/games:/sbin/nologin
13 ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
14 nobody:x:99:99:Nobody:/:/sbin/nologin
15 systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
16 dbus:x:81:81:System message bus:/:/sbin/nologin
17 polkitd:x:999:998:User for polkitd:/:/sbin/nologin
18 sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
19 postfix:x:89:89:/:/var/spool/postfix:/sbin/nologin

```

案例： `sort` 命令默认按每行第1个字符升序排序

```
1 # 默认按每行第1个字符升序排序
2 [root@shell ~]# sort /etc/passwd
3 adm:x:3:4:adm:/var/adm:/sbin/nologin
4 bin:x:1:1:bin:/bin:/sbin/nologin
5 daemon:x:2:2:daemon:/sbin:/sbin/nologin
6 dbus:x:81:81:System message bus:/:/sbin/nologin
7 ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
8 games:x:12:100:games:/usr/games:/sbin/nologin
9 halt:x:7:0:halt:/sbin:/sbin/halt
10 lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
11 mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
12 nobody:x:99:99:Nobody:/:/sbin/nologin
13 operator:x:11:0:operator:/root:/sbin/nologin
14 polkitd:x:999:998:User for polkitd:/:/sbin/nologin
15 postfix:x:89:89:/:/var/spool/postfix:/sbin/nologin
16 root:x:0:0:root:/root:/bin/bash
17 shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
18 sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
19 sync:x:5:0:sync:/sbin:/bin/sync
20 systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
```

案例： `-t -k` 配合——按分隔符分割字段，并指定字段排序



```
1 # 以:为分隔符分割字段, 按第5字段进行排序
2 [root@shell ~]# sort -t":" -k5 /etc/passwd
3 adm:x:3:4:adm:/var/adm:/sbin/nologin
4 bin:x:1:1:bin:/bin:/sbin/nologin
5 daemon:x:2:2:daemon:/sbin:/sbin/nologin
6 ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
7 games:x:12:100:games:/usr/games:/sbin/nologin
8 halt:x:7:0:halt:/sbin:/sbin/halt
9 lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
10 mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
11 nobody:x:99:99:Nobody:/:/sbin/nologin
12 operator:x:11:0:operator:/root:/sbin/nologin
13 sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
14 root:x:0:0:root:/root:/bin/bash
15 shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
16 sync:x:5:0:sync:/sbin:/bin/sync
17 systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
18 dbus:x:81:81:System message bus:/:/sbin/nologin
19 polkitd:x:999:998:User for polkitd:/:/sbin/nologin
20 postfix:x:89:89:/:/var/spool/postfix:/sbin/nologin
21 # 以:为分隔符分割字段, 按第1,2字段进行排序
22 [root@shell ~]# sort -t":" -k1,2 /etc/passwd
23 adm:x:3:4:adm:/var/adm:/sbin/nologin
24 bin:x:1:1:bin:/bin:/sbin/nologin
25 daemon:x:2:2:daemon:/sbin:/sbin/nologin
26 dbus:x:81:81:System message bus:/:/sbin/nologin
27 ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
28 games:x:12:100:games:/usr/games:/sbin/nologin
29 halt:x:7:0:halt:/sbin:/sbin/halt
30 lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
31 mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
32 nobody:x:99:99:Nobody:/:/sbin/nologin
33 operator:x:11:0:operator:/root:/sbin/nologin
34 polkitd:x:999:998:User for polkitd:/:/sbin/nologin
35 postfix:x:89:89:/:/var/spool/postfix:/sbin/nologin
36 root:x:0:0:root:/root:/bin/bash
37 shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
38 sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
39 sync:x:5:0:sync:/sbin:/bin/sync
40 systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
```

案例: **-r** 选项——根据指定字段逆序排序

```
1 # 以:为分隔符分割字段, 按第1字段进行排序, 逆序
2 [root@shell ~]# sort -t":" -k1 -r /etc/passwd
3 systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
4 sync:x:5:0:sync:/sbin:/bin/sync
5 sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
6 shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
7 root:x:0:0:root:/root:/bin/bash
8 postfix:x:89:89::/var/spool/postfix:/sbin/nologin
9 polkitd:x:999:998:User for polkitd:/:/sbin/nologin
10 operator:x:11:0:operator:/root:/sbin/nologin
11 nobody:x:99:99:Nobody:/:/sbin/nologin
12 mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
13 lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
14 halt:x:7:0:halt:/sbin:/sbin/halt
15 games:x:12:100:games:/usr/games:/sbin/nologin
16 ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
17 dbus:x:81:81:System message bus:/:/sbin/nologin
18 daemon:x:2:2:daemon:/sbin:/sbin/nologin
19 bin:x:1:1:bin:/bin:/sbin/nologin
20 adm:x:3:4:adm:/var/adm:/sbin/nologin
```

案例: **-n** 选项——按数值大小排序

```
1 # 按字母和数字排序的差异
2 [root@shell ~]# seq 1 11 | sort
3 1
4 10
5 11
6 2
7 3
8 4
9 5
10 6
11 7
12 8
13 9
14 [root@shell ~]# seq 1 11 | sort -n
15 1
16 2
17 3
18 4
19 5
20 6
21 7
22 8
23 9
24 10
25 11
26 # 以:为分隔符分割字段, 按第5字段进行排序, 按数值大小排序
27 [root@shell ~]# sort -t":" -k3 -n /etc/passwd
28 root:x:0:0:root:/root:/bin/bash
29 bin:x:1:1:bin:/bin:/sbin/nologin
30 daemon:x:2:2:daemon:/sbin:/sbin/nologin
31 adm:x:3:4:adm:/var/adm:/sbin/nologin
32 lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
33 sync:x:5:0:sync:/sbin:/bin/sync
34 shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
35 halt:x:7:0:halt:/sbin:/sbin/halt
36 mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
37 operator:x:11:0:operator:/root:/sbin/nologin
38 games:x:12:100:games:/usr/games:/sbin/nologin
39 ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
40 sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
41 dbus:x:81:81:System message bus:/:/sbin/nologin
42 postfix:x:89:89:/:/var/spool/postfix:/sbin/nologin
43 nobody:x:99:99:Nobody:/:/sbin/nologin
44 systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
45 polkitd:x:999:998:User for polkitd:/:/sbin/nologin
```

## 案例：-u 选项——去除重复行后进行排序

Shell | 复制代码

```
1 [root@shell ~]# cut -d ":" -f4 /etc/passwd | sort
2 0
3 0
4 0
5 0
6 0
7 1
8 100
9 12
10 192
11 2
12 4
13 50
14 7
15 74
16 81
17 89
18 99
19 998
20 [root@shell ~]# cut -d ":" -f4 /etc/passwd | sort -u
21 0
22 1
23 100
24 12
25 192
26 2
27 4
28 50
29 7
30 74
31 81
32 89
33 99
34 998
```

## 三、使用uniq去除重复内容

`uniq` 命令全称是unique，中文释义是“独特的，唯一的”。该命令的作用是用来去除文本文件中**连续的重复行**，**中间不能夹杂其他文本行**。去除了重复的，保留

的都是唯一的。

⚠️ **sort** 只要有重复行就去掉，而 **uniq** 重复行必须连续。

**uniq** 命令的语法格式为： **uniq** [选项] [文件]

**uniq** 命令的常用选项如下。

选项	含义
<b>-c</b>	统计每行在文本中重复出现的次数
<b>-d</b>	只显示有重复的纪录，每个重复纪录只出现一次
<b>-u</b>	只显示没有重复的纪录

▼ Shell 复制代码

```
1  #/etc/passwd文件中第4字段，多个行重复 (0)
2  [root@shell ~]# cut -d ":" -f4 /etc/passwd
3  0
4  1
5  2
6  4
7  7
8  0
9  0
10 0
11 12
12 0
13 100
14 50
15 99
16 192
17 81
18 998
19 74
20 89
```

案例： **uniq** 命令只能去除连续的重复行

```
1  # uniq命令只能去除/etc/passwd连续的重复行
2  [root@shell ~]# cut -d ":" -f4 /etc/passwd | uniq
3  0
4  1
5  2
6  4
7  7
8  0
9  12
10 0
11 100
12 50
13 99
14 192
15 81
16 998
17 74
18 89
19 # sort命令排序后再用uniq命令可去除所有重复行
20 [root@shell ~]# cut -d ":" -f4 /etc/passwd | sort | uniq
21 0
22 1
23 100
24 12
25 192
26 2
27 4
28 50
29 7
30 74
31 81
32 89
33 99
34 998
```

案例： **-c** 选项——统计每行在文本中重复出现的次数

```
1 # -c选项统计每行在文本中重复出现的次数
2 [root@shell ~]# cut -d ":" -f4 /etc/passwd | sort | uniq -c
3      5 0
4      1 1
5      1 100
6      1 12
7      1 192
8      1 2
9      1 4
10     1 50
11     1 7
12     1 74
13     1 81
14     1 89
15     1 99
16     1 998
```

案例： **-d** 选项——只保留重复的行

```
1 # -d选项只保留重复的行
2 [root@shell ~]# cut -d ":" -f4 /etc/passwd | sort | uniq -d
3 0
```

案例： **-u** 选项——只保留不重复的行

```

1  # -u选项只保留不重复的行
2  [root@shell ~]# cut -d ":" -f4 /etc/passwd | sort | uniq -u
3  1
4  100
5  12
6  192
7  2
8  4
9  50
10 7
11 74
12 81
13 89
14 99
15 998

```

## 四、使用wc统计文件的字节数、字数、行数

`wc` 命令统计指定文件中的**字节数**、**字数**、**行数**，并将统计结果显示输出。

若不指定文件名称，或者文件名为 `-`，则 `wc` 命令会从标准输入设备读取数据。

`wc` 命令的语法格式为：`wc [选项] [文件]`

`wc` 命令的常用选项如下。

选项	含义
<code>-w</code>	统计字数，或 <code>--words</code> ：只显示字数。一个字被定义为由空白、跳格或换行字符分隔的字符串。
<code>-c</code>	统计字节数，或 <code>--bytes</code> 或 <code>--chars</code> ：只显示字节数。
<code>-l</code>	<b>统计行数，或 <code>--lines</code>：只显示列数。</b>
<code>-m</code>	统计字符数。
<code>-L</code>	打印最长行的长度。



```

1  # 默认显示文件的行数、字数、字符数
2  [root@shell ~]# wc /etc/hosts
3      2  10 158 /etc/hosts
4  # 统计文件的行数
5  [root@shell ~]# wc -l /etc/hosts
6      2 /etc/hosts
7  # 统计文件的词数
8  [root@shell ~]# wc -w /etc/hosts
9      10 /etc/hosts
10 # 统计文件的字节数
11 [root@shell ~]# wc -c /etc/hosts
12     158 /etc/hosts
13 # 统计文件的字符数
14 [root@shell ~]# wc -m /etc/hosts
15     158 /etc/hosts
16 # 输出最长行的长度
17 [root@shell ~]# wc -L /etc/hosts
18      78 /etc/hosts
19 # 统计用户数
20 [root@shell ~]# cat /etc/passwd | wc -l
21      18
22 # 统计用户组数
23 [root@shell ~]# cut -d ":" -f4 /etc/passwd | sort | uniq -u | wc -l
24      13

```

## 五、使用tr进行文本转换

`tr` 的英文全称是 `translate`，即转换的意思。该命令的作用是一种可将字符进行替换、压缩、删除，他可以将**一组字符**转换成**另一组字符**。

`tr` 命令从标准输入中替换、缩减或删除字符，并将结果写到标准输出。

`tr` 命令只能从标准输入中读取数据，因此，`tr` 要么将输入文件重定向到标准输入，要么从管道读入数据。

注意：`tr` 类似于 `sed` 命令，但是比 `sed` 简单，所以 `tr` 能实现的功能，`sed` 都能实现。

`tr` 命令的语法格式为：`tr [选项] [字符集1] [字符集2]`

`tr` 命令的常用选项如下。

选项	含义	语法格式
----	----	------

无选项	<b>按位依次</b> 将字符集1中的字符替换为字符集2中的字符	<code>tr 字符集1 字符集2</code>
<code>-d</code>	删除字符集1中出现的所有字符	<code>tr -d 字符集1</code>
<code>-s</code>	删除所有重复出现的字符序列，只保留一个	<code>tr -s 字符集1</code>
<code>-c</code>	选定字符集1的补集，即反选字符集1的补集	<code>tr -c 字符集1 字符集2</code>

`tr` 命令的支持的常见字符集如下。

字符集	含义
<code>\\</code>	反斜杠
<code>\n</code>	新行
<code>\r</code>	回车
<code>\t</code>	tab键、制表符
<code>\v</code>	水平制表符
字符1-字符2	字符范围从 字符1 到 字符2 的指定，范围的指定以 ASCII 码的次序为基础， <b>只能由小到大，不能由大到小</b> 。
<code>[字符*]</code>	字符集2 专用的设定，功能是重复指定的字符到与 字符集1 相同长度为止
<code>[字符*REPEAT]</code>	字符集2 专用的设定，功能是重复指定的字符到设定的 REPEAT 次数为止 (REPEAT 的数字采 8 进位制计算，以 0 为开始)
<code>[:alnum:]</code>	所有字母字符与数字
<code>[:alpha:]</code>	所有字母字符
<code>[:blank:]</code>	所有水平空格
<code>[:cntrl:]</code>	所有控制字符
<code>[:digit:]</code>	所有数字
<code>[:graph:]</code>	所有可打印的字符(不包含空格符)
<code>[:lower:]</code>	所有小写字母
<code>[:print:]</code>	所有可打印的字符(包含空格符)
<code>[:punct:]</code>	所有标点字符
<code>[:space:]</code>	所有水平与垂直空格符
<code>[:upper:]</code>	所有大写字母

`[:xdigit:]`

所有 16 进位制的数字

▼

Shell

复制代码

```
1 [root@shell ~]# cat /etc/hosts
2 127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdoma
   in4
3 ::1         localhost localhost.localdomain localhost6 localhost6.localdoma
   in6
```

## 案例： `tr` 命令的默认行为

▼

Shell

复制代码

```
1 #把所有l替换为x
2 [root@shell ~]# cat /etc/hosts | tr l x
3 127.0.0.1    xocaxhost xocaxhost.xocaxdomain xocaxhost4 xocaxhost4.xocaxdoma
   in4
4 ::1         xocaxhost xocaxhost.xocaxdomain xocaxhost6 xocaxhost6.xocaxdoma
   in6
5 #把所有l替换为x，把所有o替换为y。
6 #注意!不是将lo替换为xy
7 [root@shell ~]# cat /etc/hosts | tr lo xy
8 127.0.0.1    xycaxhyst xycaxhyst.xycaxdymain xycaxhyst4 xycaxhyst4.xycaxdyma
   in4
9 ::1         xycaxhyst xycaxhyst.xycaxdymain xycaxhyst6 xycaxhyst6.xycaxdyma
   in6
```

```

1 # 将文件中的小写字母替换为大写
2 [root@shell ~]# cat /etc/hosts | tr a-z A-Z
3 127.0.0.1    LOCALHOST LOCALHOST.LOCALDOMAIN LOCALHOST4 LOCALHOST4.LOCALDOM
  AIN4
4 :::1        LOCALHOST LOCALHOST.LOCALDOMAIN LOCALHOST6 LOCALHOST6.LOCALDOM
  AIN6
5 [root@shell ~]# cat /etc/hosts | tr "[a-z]" "[A-Z]"
6 127.0.0.1    LOCALHOST LOCALHOST.LOCALDOMAIN LOCALHOST4 LOCALHOST4.LOCALDOM
  AIN4
7 :::1        LOCALHOST LOCALHOST.LOCALDOMAIN LOCALHOST6 LOCALHOST6.LOCALDOM
  AIN6
8 # 将文件中的数字替换为*
9 [root@shell ~]# cat /etc/hosts | tr [0-9] "*"
10 ***.*.*.*   localhost localhost.localdomain localhost* localhost*.localdom
  ain*
11 :::*        localhost localhost.localdomain localhost* localhost*.localdom
  ain*

```

### 案例： `-d` 选项根据字符集删除字符

```

1 # 删除换行符
2 [root@shell ~]# cat /etc/hosts | tr -d "[\n]"
3 127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdom
  ain4:::1    localhost localhost.localdomain localhost6 localhost6.loca
  ldomain6[root@shell ~]#
4 # 删除所有空格
5 [root@shell ~]# cat /etc/hosts | tr -d "[:space:]"
6 127.0.0.1localhostlocalhost.localdomainlocalhost4localhost4.localdomain4::
  1localhostlocalhost.localdomainlocalhost6localhost6.
7 # 删除文件中的ltoa字符
8 [root@shell ~]# cat /etc/hosts | tr -d "ltoa"

```

### 案例： `-s` 选项根据字符集去除连续的重复字符（只保留1个）

```

1 # 删除文本中匹配字符集od的重复字符，d无重复，o有重复，因此只剩下1个o
2 [root@shell ~]# echo 'ffoooood' | tr -s 'od'
3 ffod
4 # 只合并的重复的3个o
5 [root@shell ~]# echo 'ffooodod' | tr -s 'od'
6 ffodod

```

### 案例：-c 选项选择字符集之外的其他字符

```

1 # 将od之外的字符全部替换为-
2 [root@shell ~]# echo 'ffoooood' | tr -c 'od' '-'
3 --ooooo-[root@shell ~]#

```

## 六、使用tee实现多重定向

`tee` 命令的作用是从标准输入设备读取数据，将其内容输出到标准输出设备，同时保存成文件。

`tee` 命令和 `>` 重定向很相似，区别在于 `>` 重定向只会将内容重定向到文件，而不会在终端输出，而 `tee` 命令会在输出到终端的同时，将内容重定向到文件。

`tee` 命令可以用到Shell脚本的管道及输入/输出重定向的调试上。如果由管道连接起来的一系列命令的执行结果并非如预期的那样，就需要逐步检查各条命令的执行结果来定位错误，但因为使用了管道，这些中间结果并不会显示在屏幕上，这给调试带来了困难，此时，使用 `tee` 命令可以解决该问题。

`tee` 命令的语法格式为：`tee [选项] [文件1] [文件2]...`

`tee` 命令的常用选项如下。

选项	含义
<code>-a</code>	追加至既有文件的后面，而非覆盖它。相当于 <code>&gt;&gt;</code> 。
<code>-i</code>	忽略中断信号。 <code>Ctrl+C</code> 不能终止重定向，只能等待重定向完成。

### 案例：tee 命令和重定向的区别

```
1 # tee命令, 既有标准输出也将标准输出保存至文件
2 [root@shell ~]# echo 123 | tee aa.txt
3 123
4 # 重定向将标准输出保存至文件
5 [root@shell ~]# echo 123 > aaa.txt
6 [root@shell ~]# cat aa.txt
7 123
8 [root@shell ~]# cat aaa.txt
9 123
```

案例： `-a` 选项将输入追加至文件

```
1 [root@shell ~]# echo 123 | tee aa.txt
2 123
3 [root@shell ~]# cat aa.txt
4 123
5 [root@shell ~]# echo 123 | tee -a aa.txt
6 123
7 [root@shell ~]# cat aa.txt
8 123
9 123
```

案例：文件为 `-`，则标准输出输出两次。

```
1 [root@shell ~]# echo -e '123\n456' | tee -
2 123
3 456
4 123
5 456
```

## 七、使用paste合并文本

`paste` 命令会把每个文件以列对列的方式，一列列地加以合并，相当于把两个不同的文件内容粘贴在一起，形成新的文件。

`paste` 命令默认粘贴方式以列的方式粘贴，但是加上 `-s` 选项就可以行方式粘贴。

若不指定文件名称，或者文件名为 `-`，则 `paste` 命令会从标准输入设备读取数据。

`paste` 命令的语法格式为：`paste [参数] [文件1] [文件2]`

`paste` 命令的常用选项如下。

选项	含义
<code>-d</code>	默认域的分隔符是空格或tab键，设置新的域分隔符
<code>-s</code>	将每个文件粘贴成一行（按行粘贴）

▼ Shell 复制代码

```
1 [root@shell ~]# vi file1
2 a
3 b
4 c
5 d
6 e
7 [root@shell ~]# vi file2
8 A
9 B
10 C
11 D
12 E
```

### 案例：将file1和file2粘贴成一个新的文件

▼ Shell 复制代码

```
1 [root@shell ~]# paste file1 file2
2 a      A
3 b      B
4 c      C
5 d      D
6 e      E
```

### 案例：将file1和file2粘贴成一个新的文件，分隔符为 `:`

```
1 [root@shell ~]# paste -d: file1 file2
2 a:A
3 b:B
4 c:C
5 d:D
6 e:E
```

案例：将file1和file2粘贴成一个新的文件，将每个文件粘贴成一行

```
1 [root@shell ~]# paste -d: -s file1 file2
2 a:b:c:d:e
3 A:B:C:D:E
```

案例：从标准输入读取数据，并对数据进行分隔

```
1 # 从标准输入中读取数据，每行显示3个文件名
2 [root@shell ~]# ls | paste -d" " - - -
3 aaa.txt aa.txt anaconda-ks.cfg
4 moniter.sh netstat.sh no_break.sh
5 sed_output.txt sed_redirect.txt sed_result.txt
6 sex.txt temp.txt test_sed.txt
7 test_sed.txt.bak user1.txt yum.sh
8 [root@shell ~]# seq 1 4 | paste -d"-" - -
9 1-2
10 3-4
11 [root@shell ~]# seq 1 4 | paste -d"-" - - -
12 1-2-3
13 4--
14 [root@shell ~]# seq 1 6 | paste -d"." - - -
15 1.2.3
16 4.5.6
```

## 小结

- `cut` 命令：截取文本 `cut -c` `cut -d -f`
- `sort` 命令：排序 `sort -t -k`



- `uniq` 命令：去除重复记录
- `wc` 命令：统计文件字数、行数 `wc -l`
- `tr` 命令：文本转换 `tr 字符集1 字符集2` `tr -d 字符集`
- `tee` 命令：多重定向
- `paste` 命令：合并文本 `paste -d`

## 课程目标

- 知识目标：熟练掌握cut、sort、uniq、wc、tr、tee、paste命令的基本语法。
- 技能目标：能够利用文本处理基础命令完成实际需求。

## 课外拓展

- 进一步了解文本处理命令的应用场景。

## 参考资料

- 《Linux系统命令及Shell脚本实践指南》，王军，机械工业出版社
- 《跟老男孩学Linux运维：Shell编程实战》，老男孩，机械工业出版社