过滤器

cat、tac和rev

- 多个文件合并： cat FILE1 FILE2 > output
- 加上行号

- cat [OPTION]... [FILE] 将一个或者多个文件的内容打印到标准输出

不指定文件时从标准输入读。FILE为-时表示从标准输入读

-n, --number	最前面添加行号
-b, --number-nonblank	添加行号，但是对于空行不添加
-s, --squeeze-blank	多个连续空行仅仅保留1个空行
-v, --show-nonprinting	非打印字符以^X等形式显示
-A, --show-all	显示非打印字符、Tab(^I)和换行($)

- tac [FILE]将一个或者多个文件的内容打印到标准输出，只是每个文件的内容逆序打印，即最后1行首先打印，然后是倒数第2行…最后是第1行。不指定文件时从标准输入读
- rev [FILE] 将一个或多个文件中的内容打印到标准输出，只是每行的字符反转输出，即每行最后一个字符、倒数第二个...第一个字符。不指定文件时从标准输入读

```
$ tac /etc/{networks,shells}
link-local 169.254.0.0
# symbolic names for networks, ...
/usr/bin/tmux
/bin/dash
/bin/rbash
/bin/bash
/bin/sh
# /etc/shells: valid login shells
dlmao@mars:~$ tac
line 1
line 2
line 2
line 1
$ echo -e 'hello world!\nLinux'|rev
!dlrow olleh
xunil
```

复旦大学版权所有

查看文本文件的开始多行：head

head [OPTION]... [FILE]...

- 输出文件中的前面多行（缺省前面10行）。参数为多个文件时，每个文件内容前包含一行 ==>FILENAME<==

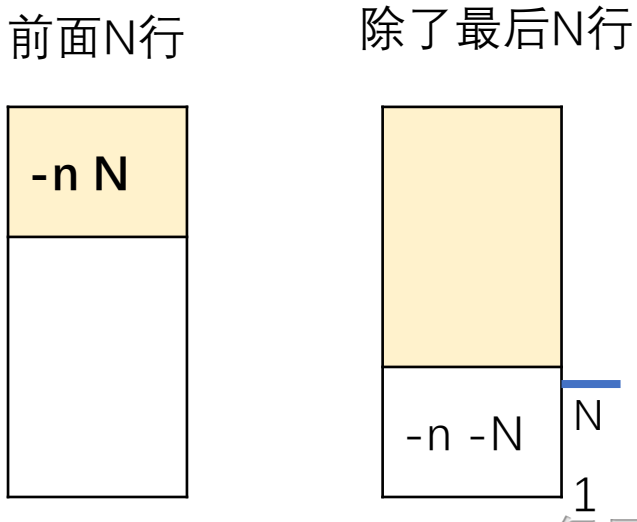
-n, --lines=[-]NUM	输出每个文件中前面N行，缺省为10行。如果为**-n -N**，表示从第一行直到（不包括）倒数第N行，即**除了最后N行外其他行都输出**
-N	等同于-n N，即输出前面N行
-c, --bytes=[-]NUM	输出前面N字节的内容。如果为-N，表示除了最后N字节外其他都输出
-q, --quiet	多个文件时，文件之间不输出包含文件名的行
-z, --zero-terminated	行之间不是以\n而是以NUL隔开

```
$ head -n 4 /etc/hosts | cat -n
     1  127.0.0.1       localhost
     2  127.0.1.1       mars
     3
     4  # The following .。。
$ head -n -8 /etc/hosts
127.0.0.1       localhost
```

取前面4行，并且加上行号

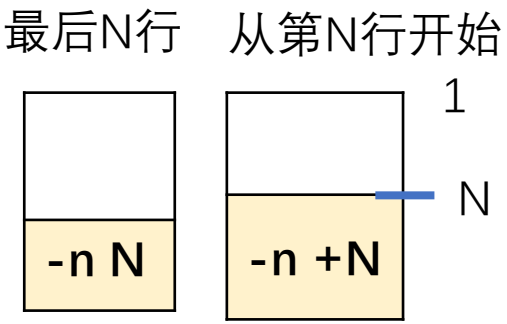
除了最后8行外的其他行

```
$ head -c 16 /bin/ls | xxd
00000000: 7f45 4c46 0201 0100 0000 0000 0000 0000  .ELF............
```



复旦大学版权所有

查看文本文件的最后多行：tail



tail [OPTION]... [FILE]...

输出每个文件中的最后多行（缺省为10行）。参数为多个文件时，每个文件内容前包含一行 ==>FILENAME<==

-n, --lines=[+]N	输出最后多少行。如果为**-n +N**，表示**从第N行（包括）开始的所有行**
-N	等同于-n N，即输出最后N行
+N	等同于-n +N，即从第N行开始的所有行
-c, --bytes=[+]N	输出最后N字节。如果为-c +N，表示从第N字节开始的所有内容
-q, --quiet	多个文件时，文件之间不输出包含文件名的行
-z, --zero-terminated	行之间不是以\n而是以NUL隔开
-f, --follow	一直等待FILE中的内容变动。当FILE中有新的内容附加到该文件中，新增的内容会输出

```
$ tail -n 2 /etc/hosts
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

```
$ ls -l / | tail -n +2   #从第2行开始，跳过第1行
drwxr-xr-x   2 root root      4096 5月  16 06:55 bin
drwxr-xr-x   3 root root      4096 5月  24 07:00 boot
...
```

ping www.fudan.edu.cn > output&	检查主机是否连通，输出重定向到output，放在后台运行
tail -f output	一直跟踪output文件，输出output文件新增的内容
kill $!	将刚才在后台运行的进程结束

复旦大学版权所有

统计文件的行/单词和字符数量：wc

wc [OPTION]... [FILE]...

缺省输出行数、单词、字节数和文件名。如果没有参数，从标准输入读，没有文件名信息

-l, --lines 输出行数，后面包含文件名

-w, --words, 输出单词数

-c, --bytes 输出字节数

-m, --chars 输出字符数

-L, --max-line-length 输出最长行的显示宽度

/etc目录下有多少个文件和目录呢?
/etc目录下有多少个普通文件?

```
$ wc /etc/{passwd,hosts}
  46   75 2633 /etc/passwd
   9   25  219 /etc/hosts
  55  100 2852 total
$ wc </etc/hosts
  9  25 219
$ wc -w /etc/hosts
25 /etc/hosts
$ ls /etc | wc -l
222
$ ls -l /etc | head -2
total 1112
drwxr-xr-x  2 root root    4096 Feb  4 02:25 ImageMagick-6
$ ls -l /etc | grep '^-' | wc -l
88
```

复旦大学版权所有

抽取数据列cut

cut OPTION... [FILE]...

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

```
$ cut -d: -f 1,6- /etc/passwd
root:/root:/bin/bash
```

```
$ echo 1:::4:5 | cut -d: -f 3-5
:4:5
```

- 许多文本文件的各行：以某个delimiter（Tab、冒号、逗号等)隔开的多列数据，注意多个连续的分隔符会隔开成多列，而不是看成一个分隔符
- cut命令从文件的每一行中抽取相应内容并且输出。在抽取时可以按照字节(-b)、字符(-c)或字段(-f, 分隔符缺省为制表符)为单位抽取指定范围的内容

LIST的格式	包含一个或多个range，range之间以,分割。range可以为：N表示第N列， N-M表示第N到第M列，-M表示第1到第M列，而N-表示第N到最后一列
-b, --bytes=LIST	选择LIST指定的字节
-c, --characters=LIST	选择LIST指定的字符，中文处理时应该采用
-f, --fields=LIST	选择LIST指定的字段，分隔符缺省为制表符\t。如果某行中没有分隔符，则该行也会输出，除非指定了-s选项
-d, --delimiter=DELIM	字段分隔符采用DELIM而不是缺省的\t
-s, --only-delimited	在采用字段来抽取时，如果某行没有分隔符，该行不输出
--complement	取反，输出那些不在-c –f -b选项中指出的列表里面的字节、字符或字段
--output-delimiter=STRING	选择字段时，输出的字段之间以STRING隔开。如果不指定，缺省为分割字段时采用的分隔符DELIM
-z, --zero-terminated	行之间采用NUL而不是缺省的换行(\n)分隔

复旦大学版权所有

抽取数据列cut

```
root:x:0:0:root:/root:/bin/bash
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
```

- 以某个delimiter（Tab/冒号/逗号等)隔开的多列数据
- cut命令分割列的时候，不会将多个连续的分隔符看成一个整体，经常要在前期使用grep/tr/sed等命令进行预处理

```
John      99
Anne      75
Andrew    50
Tim       95
Arun      33
Sowmya    76
```

```
$ ls -l | tail +2 | cut -c 2-10
rw-rw-r--
rwxr-xr-x
...
$ grep bash /etc/passwd | cut -d: -f 1,3-4,6 --output-delimiter=$'\t'
root    0       0       /root
dlmao   1000    1000    /home/dlmao
$ cut -d: -f 2,5 --complement /etc/passwd
root:0:0:/root:/bin/bash
daemon:1:1:/usr/sbin:/usr/sbin/nologin
...
$ ps ax | grep [s]sh
  910 ?        Ss     0:00 /usr/sbin/sshd -D
 1723 ?        Ss     0:00 sshd: dlmao [priv]
 1845 ?        S      0:02 sshd: dlmao@pts/0
$ $ ps ax | grep [s]sh | grep -o -E '\S+.*$' | cut -d' ' -f 1 | xargs
910 1723 1845
```

列出当前目录下所有文件的权限字段，前面的管道命令去掉ls长列表的第一行

输出passwd文件中除了shadow和desc外的其他字段

复旦大学版权所有

删除数据列colrm和格式化列column

colrm [start [end]]

cut -c start-end --complement

- 从标准输入读取数据，每行删除指定的字符（从start到end)后输出。colrm start表示删除从start到最后的字符，即仅保留start之前的字符

```
$ ls -ld .  | colrm 11
drwxr-xr-x
$ ls -ld . | cut -c 11- --complement
drwxr-xr-x
```

column [-et] [-s sep] [file ...]
以多列方式格式化。一般使用-t选项产生一个表格形式的输出

```
$ echo -e '1\t      2      \t\t3\n4 56789 0'
1               2                       3
4 56789 0
$ echo -e '1\t      2      \t\t3\n4 56789 0'
| column -t
1  2       3
4  56789   0
```

-t	根据分隔符(缺省为空格和制表符)来决定文件中每行内容的各列。注意多个连续的分隔符看成一个分隔符。输出的结果为表格形式，添加足够的空格保证各列对齐
-s sep	指定sep作为列之间的分隔符，缺省为$' \t'
-e	空行保留。缺省空行被删除

```
$ column -t -s : < /etc/passwd
root                x  0    0    root                /root                /bin/bash
daemon              x  1    1    daemon              /usr/sbin            /usr/sbin/nologin
bin                 x  2    2    bin                 /bin                 /usr/sbin/nologin
```

复旦大学版权所有

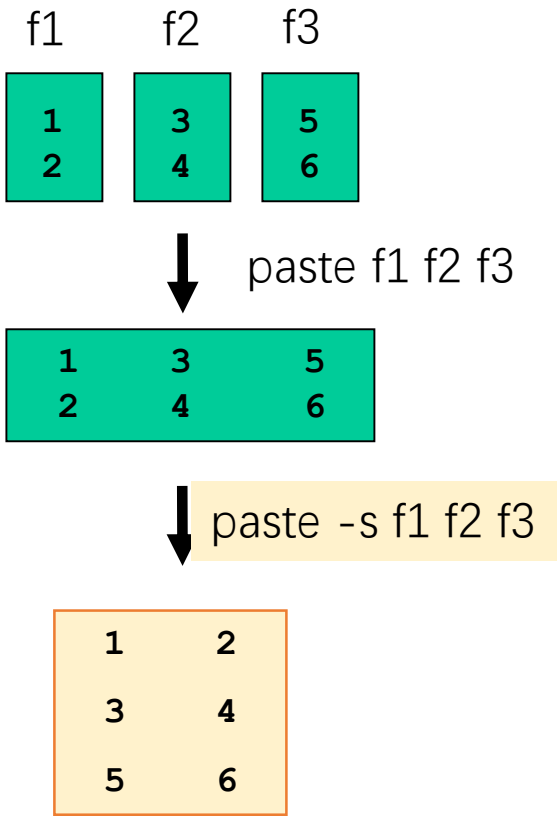
组合数据列 paste

paste [-d DELIM –s]... [FILE]...

- 将多个文件的数据组合在一起输出。每个文件看成包含了某一列的内容，将多个文件的数据组合在一起输出，列之间**缺省以\t分隔**。如果某个文件没有足够的行，则对应的列为空字符串

-d, --delimiters=LIST	指定**组合时列之间的分隔符**。如果LIST包括多个字符，则第一个间隔使用LIST中的第一个字符，第二个间隔使用第二个字符，直到最后的字符用完后又从头开始轮流使用
-s, --serial	把**文件中的各行合并成一行**，这些行的内容之间以分隔符(缺省\t)隔开。即第一行为第一个文件中所有行合并后的结果，第二行为第二个文件中所有行合并后的结果...

```
$ paste -d $'\t,' <(echo -e '1. a\n2. b\n3. c') <(echo -e 'tony\ntom')
<(echo -e '12345\n456\n789')
1. a    tony,12345
2. b    tom,456
3. c    ,789
$ paste -s <(echo -e '1\n2\n3') <(echo -e 'a\nb')
1       2       3
a       b
```



复旦大学版权所有

排序sort

```
sort -t : -k 7 -k 1,1 < /etc/passwd   基于shell排序，相同时基于用户名排序
sort -t: -k 3,3rn /etc/passwd         按照第3个字段即UID的数值逆序排列
```

sort [-bfnr -cmu –o output –k field –t sep] … [FILE]…　将一个或多个文件的内容进行排序后输出。也可合并已经排序好的多个文件。也可以检查某个文件是否已排序好

```
$ echo -e '32.2\n4.3'|sort
32.2
4.3
$ echo -e '32.2\n4.3'|sort -n
4.3
32.2
```

基于哪些字符来排序：缺省基于整行的内容

- -k KEYDEF选项：基于某些字段或字段的某些字符进行排序
  - 如何确定字段？通过分隔符来确定字段，字段编号从1开始
    - 缺省的分隔符为空格以及制表符，而且多个连续的分隔符作为一个整体
    - 可通过选项-t DELIM指定其他单字符分隔符，但注意此时多个连续的分隔符会分割成多个字段（这些字段为空字符串）
  - -k选项有两种格式
    - -k field 表示排序基于的字符为从field描述所对应的字符开始到行结束为止
    - -k field1,field2 表示从field1描述的字符开始，到field2描述的字符结束
    - field的格式为F[.C][OPTS] F表示第几个字段，C表示该字段的第几个字符开始(如果描述开始，缺省为字段的第1个字符，否则缺省为0，表示字段的最后一个字符），OPTS给出对该字段排序时适用的选项，取值可为bfnr等，分别表示忽略开头空格、大小写无关、数值顺序和逆序
  - -k选项可出现多次，首先按照某些字符排序，如果相同时基于其他字符排序…

排序的基准：缺省按照**字符先后顺序**进行排序。环境变量LC_COLLATE决定字符先后顺序, LC_ALL=C sort …

- 可通过-n选项(numeric)表示采用**数值顺序**进行排序，可通过-r(reverse)选项表示**逆序排序**，可通过-f选项(fold)表示**大小写无关**，可通过-b选项(blank)表示忽略字段前面的空格

复旦大学版权所有

对文本行排序sort　sort [-bfnr -cmu –o output –k field –t sep] … [FILE]…

-o, --output=FILE	排序后的结果写到FILE文件中。比如sort -o file file
-b, --ignore-leading-blanks	首先将字段中开头的空格去掉再排序
-f, --ignore-case	比较时忽略大小写
-n, --numeric-sort	按照数值顺序，而不是缺省的字符顺序
-k field1[, field2]	不是基于整行，而是基于字段进行排序。如果只有field1，则表示排序用到的字符从field1描述的字符开始到行结束，否则表示从field1描述的字符开始，到field2描述的字符结束
-t, --field-separator=SEP	定义字段分隔符，缺省为空格或制表符，而且多个连续分隔符看成一个分隔符，但注意-t选项严格按照分隔符分割
-r, --reverse	逆序排序
-u, --unique	排序规则认为相同的行仅仅留下一行，可用来排序去除重复的行
-c, --check	检查文件是否已排序好，如果已排序，状态码为成功(echo $?为0)，否则错误输出中给出没有排序的第一行
-m, --merge	要求文件已经排序好，将多个已经排序好的文件合并，速度比用排序模式更快

```
$ sort -c /etc/passwd
sort: /etc/passwd:2: disorder: daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
$ echo $?
1
```

版权所有

sort示例

```
$ du -s /usr/share/* | sort -nr | head -2
204004  /usr/share/fonts
85204   /usr/share/icons
$ echo -e 'banana\napple\npear\norange\npear'  | sort -u
apple
banana
orange
pear
$ echo -e 'banana  3\napple 4\npear 5\norange 7\npear 7'  | sort -u -k 2,2n
banana  3
apple 4
pear 5
orange 7
```

/usr/share下占用空间前2位的子目录。du –s输出中第一个字段为占用空间，通过sort根据空间大小逆序排列，通过head取前面2行

进行排序，认为是重复的行被移走，即pear的行只有一行保留

基于第2个字段，且按照数值顺序排序，按照排序规则认为是重复的行仅仅保留1行，因此orange 7和pear 7认为是重复行

复旦大学版权所有

sort示例

```
$ cat distros.txt
SUSE    10.2    12/07/2006
Fedora  10      11/25/2008
SUSE    11.0    06/19/2008
Ubuntu  8.04    04/24/2008
Fedora  8       11/08/2007
SUSE    10.3    10/04/2007
Ubuntu  6.10    10/26/2006
Fedora  7       05/31/2007
Ubuntu  7.10    10/18/2007
Ubuntu  7.04    04/19/2007
SUSE    10.1    05/11/2006
Fedora  6       10/24/2006
Fedora  9       05/13/2008
Ubuntu  6.06    06/01/2006
Ubuntu  8.10    10/30/2008
Fedora  5       03/20/2006
```

```
$ sort -k 1,1 -k 2n distros.txt
Fedora  5       03/20/2006
Fedora  6       10/24/2006
Fedora  7       05/31/2007
Fedora  8       11/08/2007
Fedora  9       05/13/2008
…
$ sort -k 3.7nbr -k 3.1nbr -k 3.4nbr distros.txt
Fedora  10      11/25/2008
Ubuntu  8.10    10/30/2008
SUSE    11.0    06/19/2008
Fedora  9       05/13/2008
$ sort -nbr -k 3.7 -k 3.1 -k 3.4 distros.txt
...
```

先按发行版名，然后版本号(数值)排序

按照年份、月和日期逆序排列

复旦大学版权所有

查找重复行uniq

```
$ echo -e '1\n1\n2\n4\n3\n3' > tmp.txt
$ uniq tmp.txt
1
2
4
3
```

```
$ uniq -d tmp.txt
1
3
$ uniq -u tmp.txt
2
4
```

uniq [OPTION]... [INPUT [OUTPUT]]

- sort –u 排序后去除重复的行
- uniq并不要求已经排序好，而是检查连续的多行来判断是否重复，去掉那些多余的行
- 缺省相当于 -u -d，即去掉那些多余的行
- 从标准输入或INPUT读取文本，输出到标准输出或文件OUTPUT中

-u, --unique	仅输出那些不重复的行
-d, --repeated	仅输出重复行(多个重复行仅保留一行)，不输出那些不重复的行
-c, --count	在输出的行之前添加该行的重复次数
-f N, --skip-fields=N	每行的前面N个字段忽略，只检查后面字段的重复性。字段之间为空格或制表符分割
-s N, --skip-chars=N	忽略前面N个字符。如果-f和-s同时使用，首先-f然后-s
-i, --ignore-case	判断重复时大小写无关
-w N, --check-chars=N	在忽略相应字符后仅仅比较最多N个字符，缺省到行尾

```
$ cut -d: -f7 /etc/passwd | sort | uniq -c
      4 /bin/bash
      6 /bin/false
      1 /bin/sync
     35 /usr/sbin/nologin
```

取得口令文件的shell字段后排序，最后去掉那些重复的行，
每行前面添加重复次数，即**统计shell的使用次数**

复旦大学版权所有

替换或删除(Translate)字符 tr [-cds]... SET1 [SET2]

从标准输入读取数据后进行相应的转换(删除某些字符，替换某些字符，压缩某些字符），最后写到标准输出

-d, --delete	删除SET1中出现的字符
-s, --squeeze-repeats	表示将最后一个SET中多个连续字符替换为一个单独的字符，该操作最后完成，即首先进行替换或删除，最后再挤压
-t, --truncate-set1	将SET1截取到与SET2一样的长度。缺省是将SET2的最后一个字符重复到与SET1相同的长度
-c, --complement	表示对于不在SET1的字符进行相应的动作

tr -d SET1	删除出现在SET1中的字符
tr SET1 SET2	SET1中的字符替换为SET2对应位置字符
tr -s SET1	多个连续的字符(该字符在SET1)压缩为一个
tr -s SET1 SET2	SET1中的字符替换为SET2对应位置字符，然后压缩SET2中的连续字符
tr -ds SET1 SET2	首先删除SET1中出现的字符，接下来对SET2连续字符压缩

```
$ echo "lowercase letters  UPPERCASE LETTERS"
| tr [:lower:][:upper:] [:upper:][:lower:]
LOWERCASE LETTERS  uppercase letters
```

```
$ tr a-zA-Z  A-Za-z
$ echo -e '1\n2' |
tr -d '\015\n';echo
# 删除\r \n
```

- 将SET1中出现的字符替换为SET2中的字符；SET1和SET2中的字符个数（长度）应该一致
  - 如果SET2长度更小， SET2的最后一个字符重复多次直到与SET1相同的长度
  - 采用-t选项时，如果SET2长度更小，将SET1截取到与SET2一样的长度
  - SET2还可用[C*N]的形式描述，表示字符C出现N次，比如[y*3]相当于yyy
  - 包括转义序列(\n\t\040等)；也可以采用连字符表示字符范围，如0-9A-Z
  - 也可包括预定义字符类，如[:upper:] 。但注意只有upper和lower展开为固定的顺序，其他类展开的字符顺序不确定，因此不适合用于替换(translate)

复旦大学版权所有

tr示例

abcdefghijklmnopqrstuvwxyz
nopqrstuvwxyzabcdefghijklm

```
$ tr a-zA-Z n-za-mN-ZA-M < /etc/hosts
127.0.0.1       ybpnyubfg
127.0.1.1       znef

# Gur sbyybjvat yvarf ner qrfvenoyr sbe VCi6 pncnoyr ubfgf
::1     vc6-ybpnyubfg vc6-ybbconpx
sr00::0 vc6-ybpnyarg
ss00::0 vc6-zpnfgcersvk
ss02::1 vc6-nyyabqrf
ss02::2 vc6-nyyebhgref
$ tr -dc 'a-zA-Z0-9@#%+/~-' < /dev/urandom | head -c 8; echo
YPnDv%Ja

$ echo -e 'hello\n\n\nworld' | tr -s '\n'
hello
world

$ echo -e 'hello\n\n\nworld' | tr -s '\n'  ' '; echo
hello world
```

实现ROT13编码，按照英文字母循环移动13个位置。再次使用ROT13编码还原成最初的内容

产生长度为8的随机密码。仅允许指定的字符作为密码，注意SET1中的-应该放在最前面或最后，否则表示范围，也可用\055表示字符-

多个连续的\n压缩为一个\n，即去掉空行

先将\n替换为空格，再将多个空格压缩为一个空格

复旦大学版权所有