# linux基础

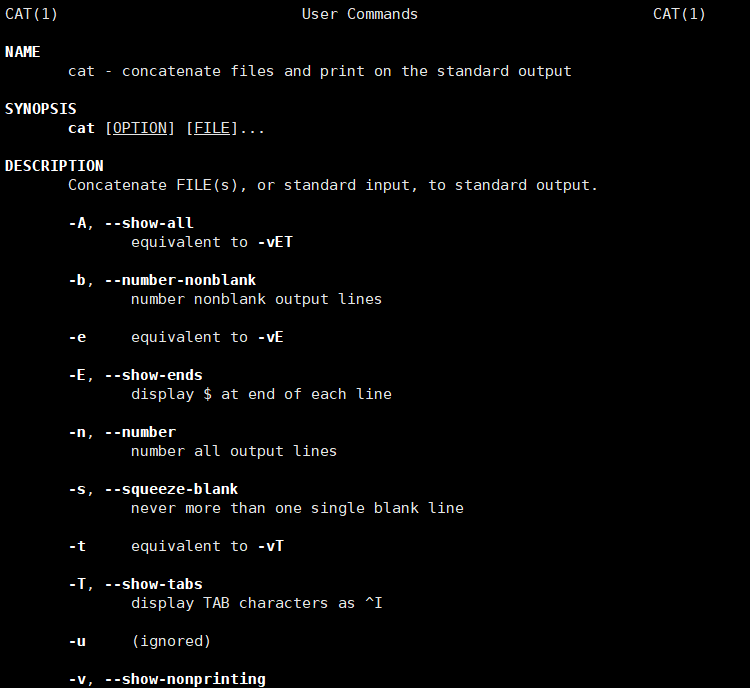
## 查看文本内容

**cat , tac , more , less ,head ,tail** 不能查看二进制文件

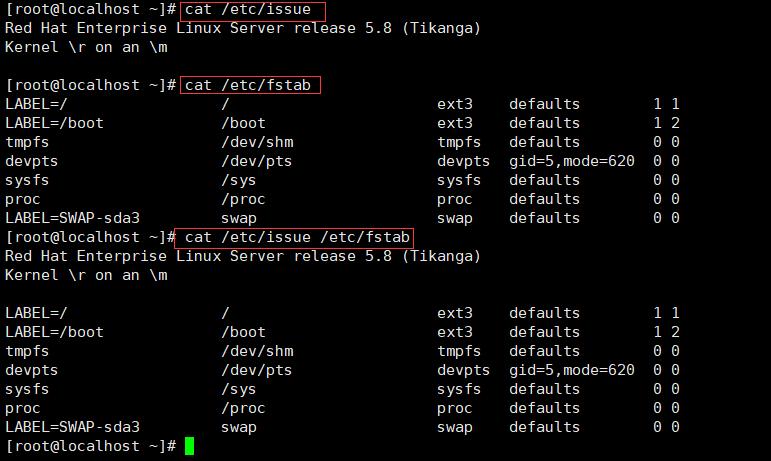
### cat

concatenate files and print on the standard output

**(连接多个文件并且打印输出)**



[root@localhost ~]# cat /etc/issue /etc/fstab 连接两个文件并显示



**-n**：--number

number all output lines (显示行号，并不是文件内容本身的行号。文件本身没有行号。)

[root@localhost ~]# cat /etc/issue /etc/fstab -n



**-E** : --show-ends

display $ at end of each line (显示每一行结束符 $)

[root@localhost ~]# cat /etc/issue /etc/fstab -nE



对于Linux系统的文本文件每一行的结束符是$符号。

windows系统的文本文件行结束符，是$和回车符，有两个，所以在把linux 系统的文本文件复制到windows系统，统统显示为一行，这是因为Linux结束符只$符号，没有回车符。

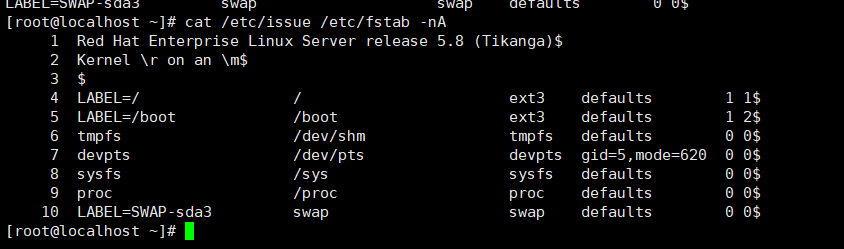
所以在windows中写的脚本，拿到Linux中通常运行不了，所以要主要，使用notepad++

就可以处理好这个问题。

**-A**, --show-all

equivalent to -vET (显示所用)

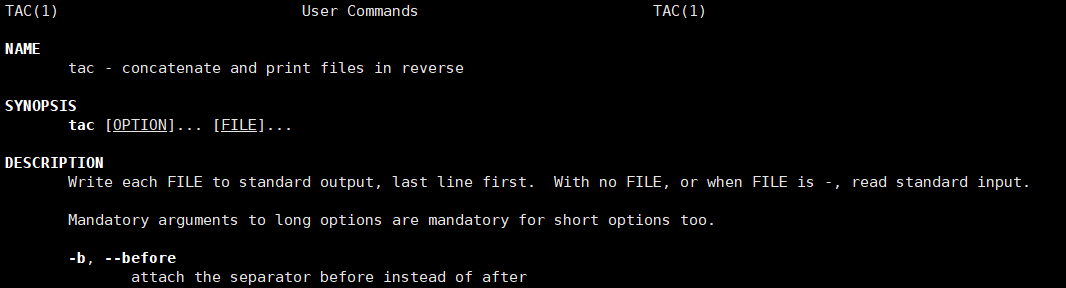
[root@localhost ~]# cat /etc/issue /etc/fstab -nA



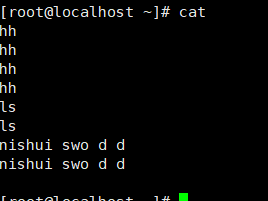
### tac

**- concatenate and print files in reverse**

**（反过来显示， cat --tac）**



如果忘记了带参数，cat命令就会回到默认的输入输出模式。这时候ctr+c退出

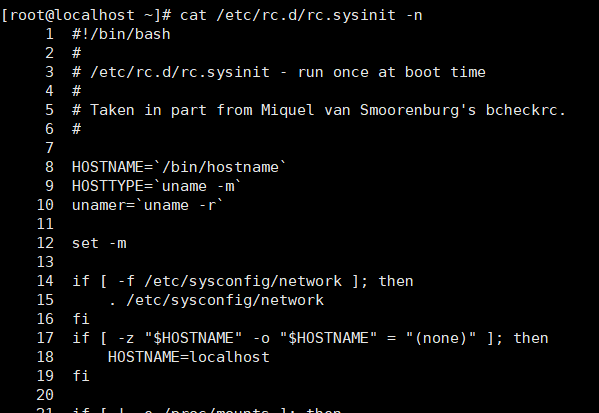


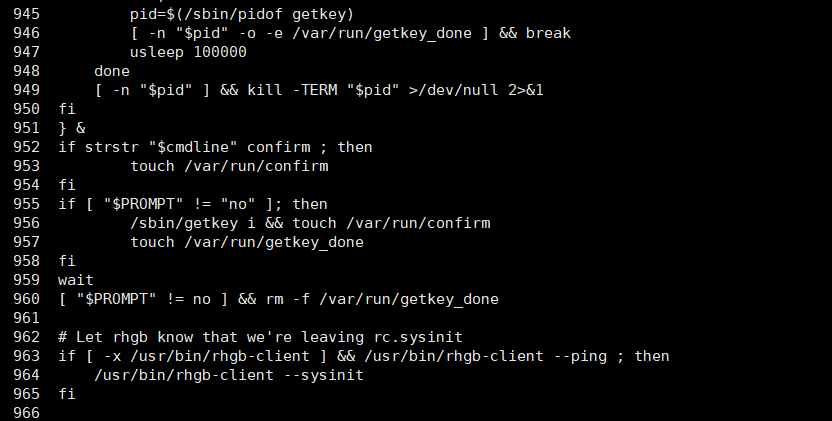
如果我不想执行一个命令，直接按ctr+c 就可以退出当前命令

[root@localhost ~]# cat /etc/issue

[root@localhost ~]#

cat 有个缺点：就是一下显示到最后，对内容比较多的文件，就比较头疼，

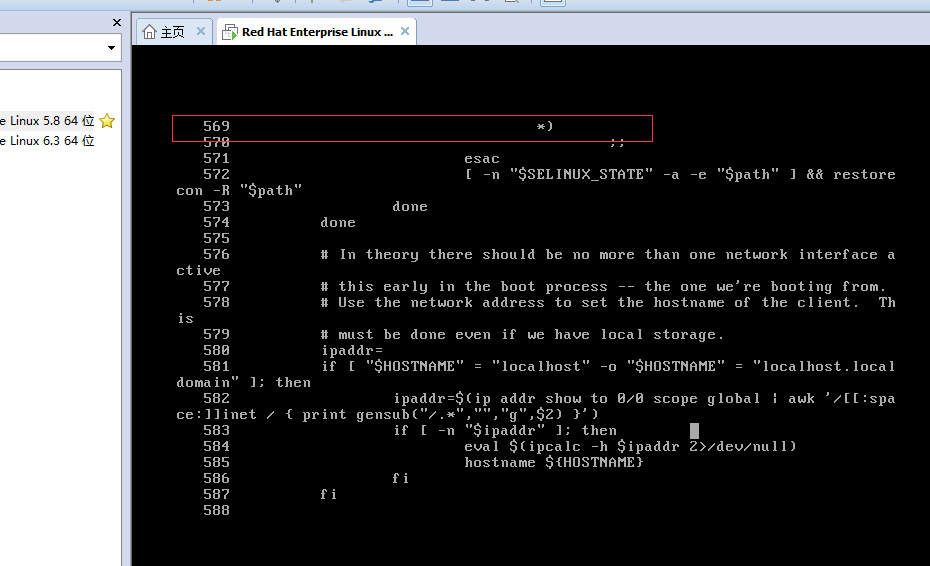




而在主机的终端上，向前翻是不行的，引文在内存中只有那么点空间来缓存内容，显示到最后，超出的内容，前面的就会丢掉。

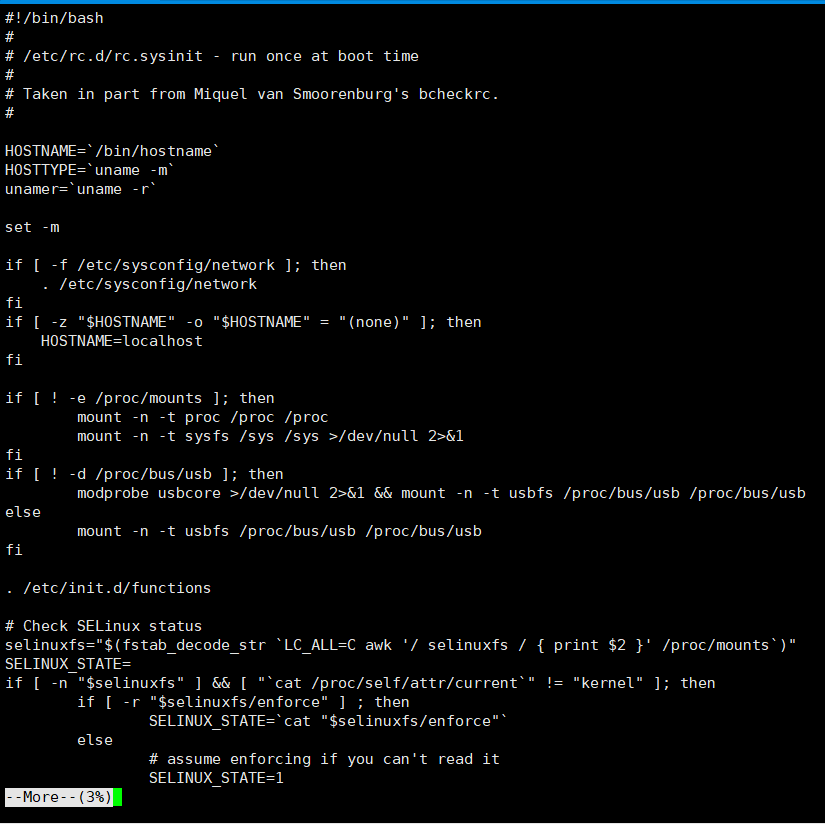
主机终端上 shift+pgup 向前翻，翻不动，但是还有内容。

第569行以前的就都丢了

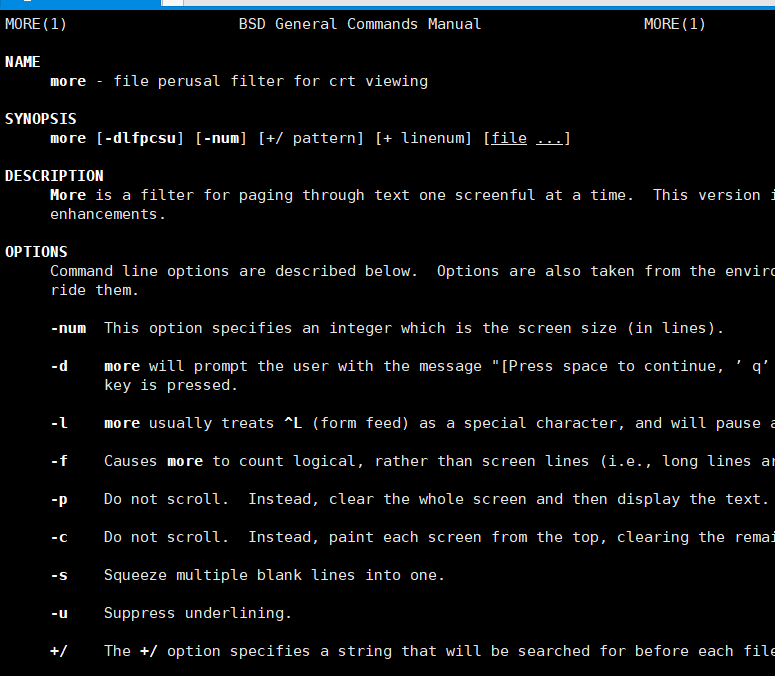


所以就有分屏，分页显示命令

**more less**



### more



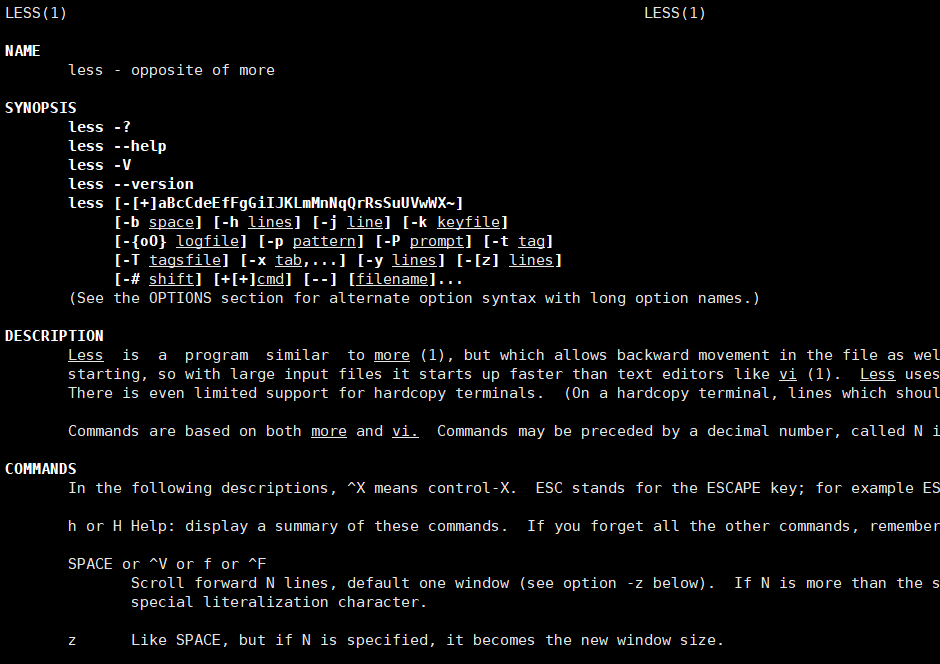
**space** 空格向后翻屏，

**b** 向前翻屏

**enter** 一行一行的往下看。

不支持**k** 不支持向前翻翻到最后就退出。

### less



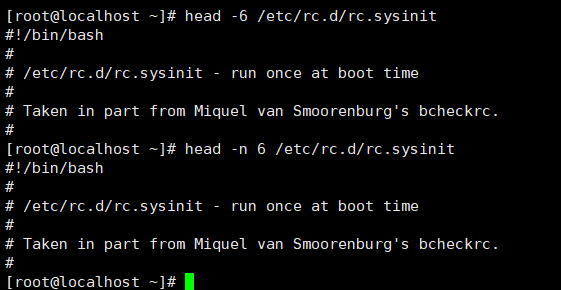
less 和more差不多，翻到最后不会自动退出，要q 命令退出。而且还可以支持k键，向前翻。常用。

### head

查看文件的前n行 默认是10行

head -n 6 /etc/rc.d/rc.sysinit

head -6 /etc/rc.d/rc.sysinit



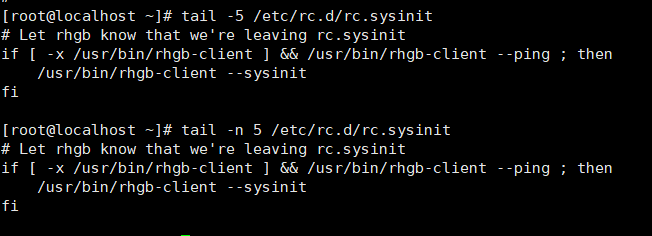
### tail

查看文件后n行 默认是10行

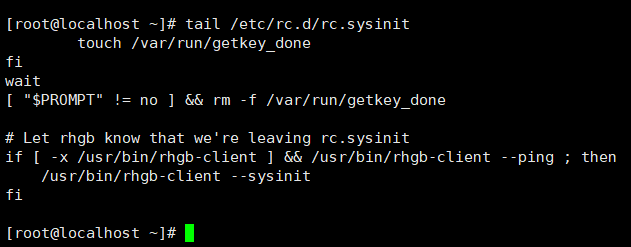
**-n** 20 支持显示前20行

tail -5 /etc/rc.d/rc.sysinit

tail -n 5 /etc/rc.d/rc.sysinit



tail /etc/rc.d/rc.sysinit 不加行数，默认10行



**tail**

**-f, --follow[={name|descriptor}]**

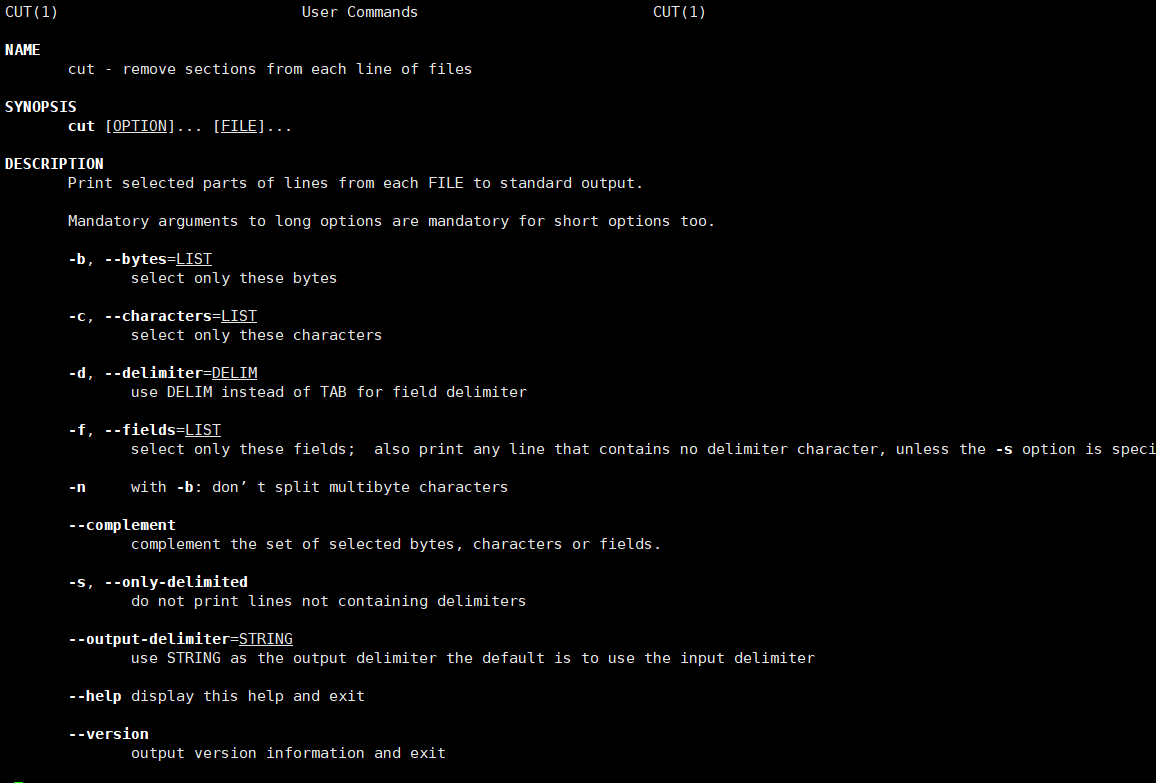
**output appended data as the file grows; -f, --follow, and --follow=descriptor are equivalent**

**tail -f** 表示查看最后并不退出，等待后续追加的内容并继续显示。相当有用。查看正在进行的日志等。。

## 文本处理的命令

**cut** , **join**, **grep** (管道) ,**sed**（stream editor）, **awk**(难，强大)

### cut



剪切，针对内容剪切。

**-d** : --delimiter=DELIM

use DELIM instead of TAB for field delimiter

delimiter 分隔符

**-f**：--fields=LIST

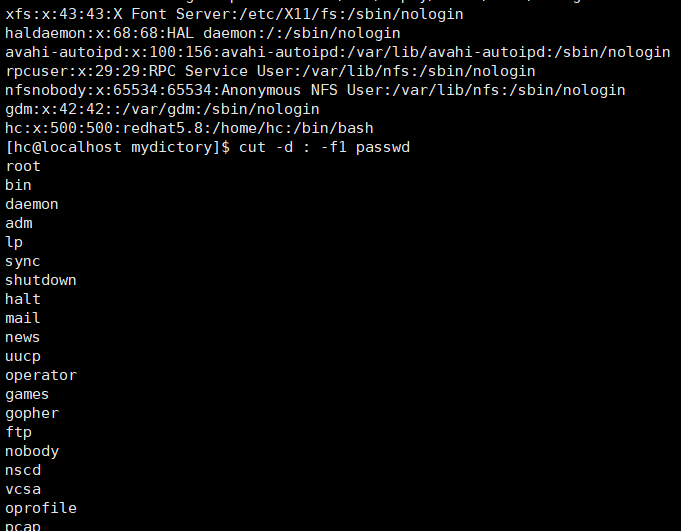
select only these fields; also print any line that co

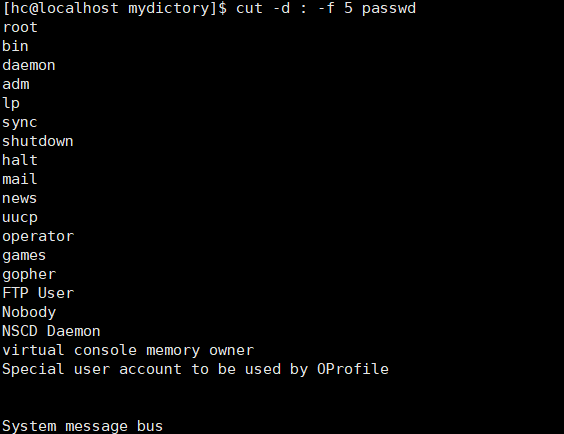
（只选择这些字段;也可以打印任何行）

cut -d : -f 1 passwd

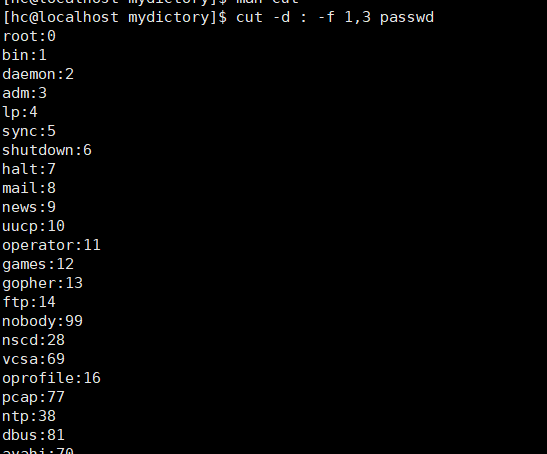
-d : 指定 : 为分隔符，

-f 1 指定第一个字段要打印



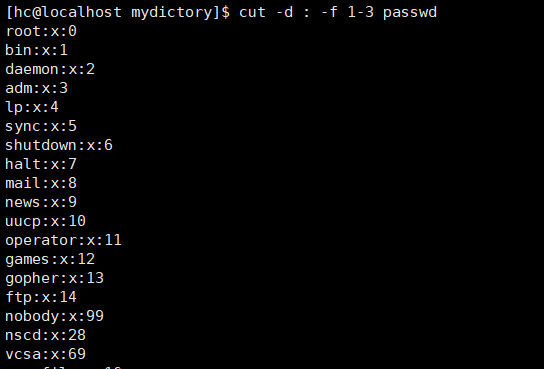


cut -d : -f 1,3 passwd 显示1和3字段



cut -d : -f 1-3 passwd

显示1到3字段



你无论怎么cut ，都不会影响源文件的。

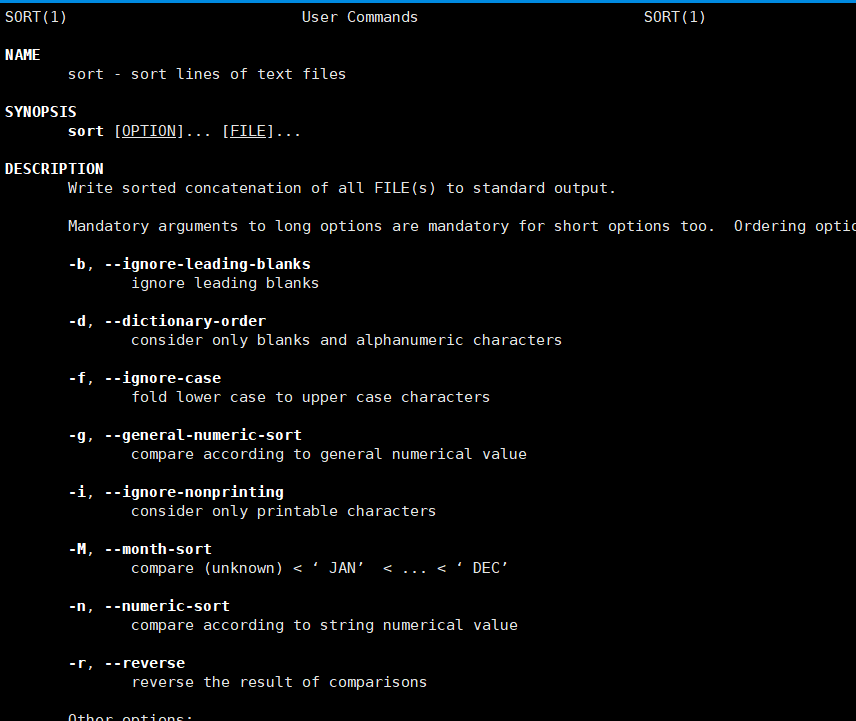
### join

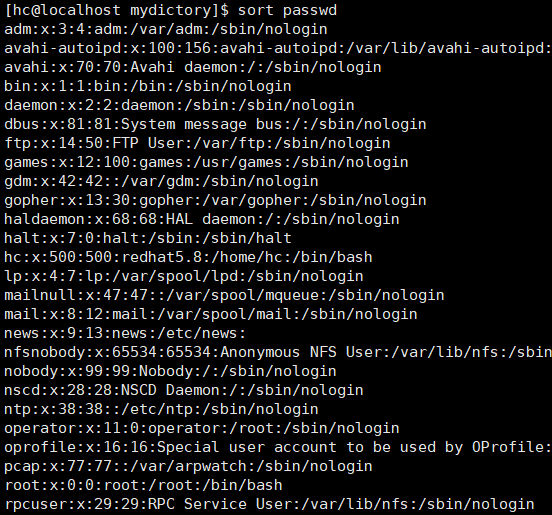
**join lines of tow files on a common field**



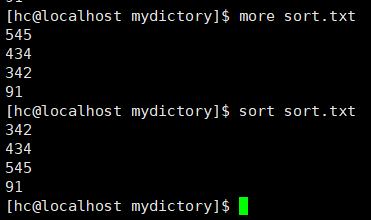
## 文本排序

### sort



[hc@localhost mydictory]$ sort passwd 默认使用升序排列 ascII码表 a-z 

**对数字排序 不是按照数字的大小来排序，而是Ascii表中的升序来拍的。**

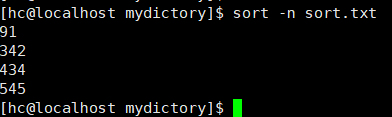


**-n**, --numeric-sort

compare according to string numerical value

（比较数值的大小排序）

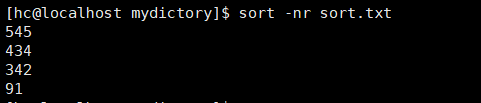
sort -n sort.txt 就不用ascii码表进行排序了



**-r**, --reverse(逆序反转)

reverse the result of comparisons

比较的结果逆序排列



**-t,** --field-separator=SEP

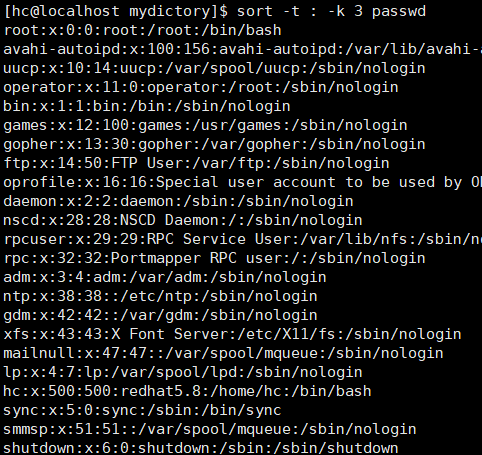
use SEP instead of non-blank to blank transition

（字段分隔符）

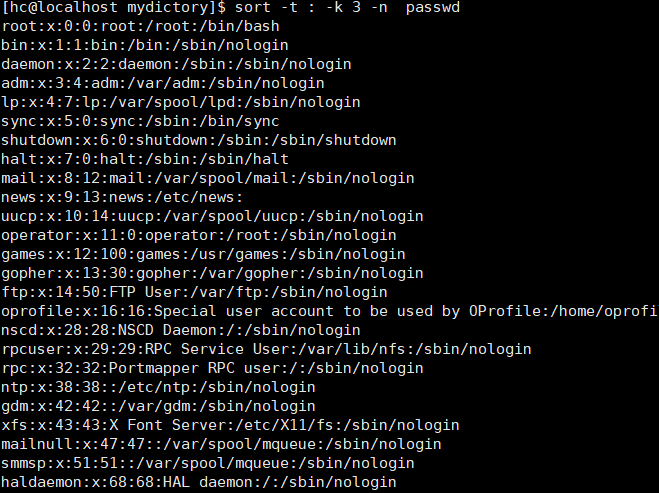
**-k,** --key=POS1[,POS2]

start a key at POS1, end it at POS2 (origin 1)

sort -t : -k 3 passwd



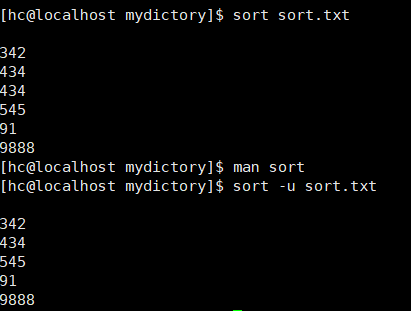
sort -t : -k 3 -n passwd 按第3个字段进行数值大小排序



-**u**, --unique （唯一）

with -c, check for strict ordering; without -c, output only the first of an equal run

sort -u sort.txt 排序后，相同的行只显示一次。



**-f,-**-ignore-case

fold lower case to upper case characters

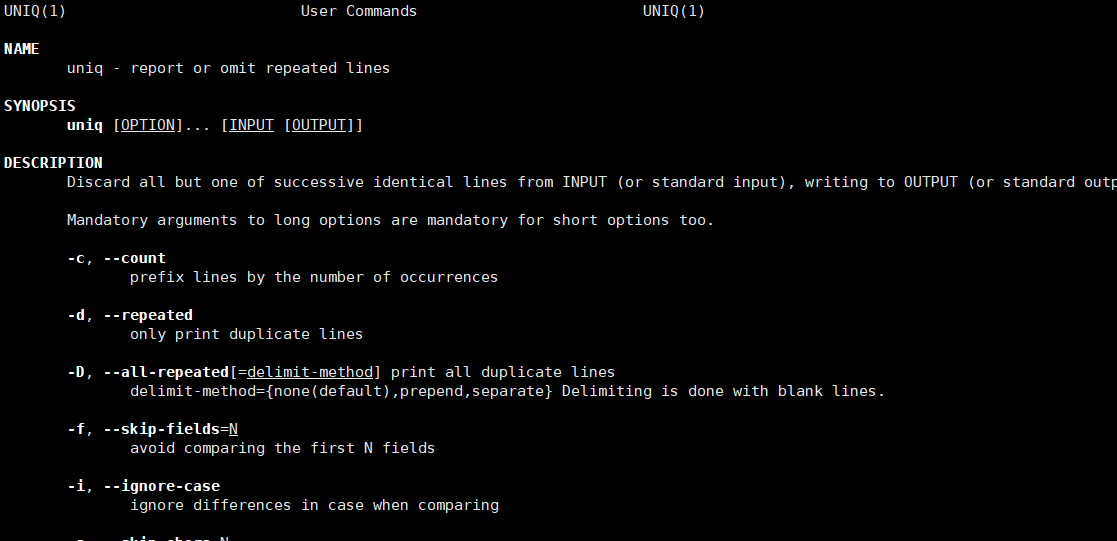
（排序时，忽略字符的大小写）

### uniq

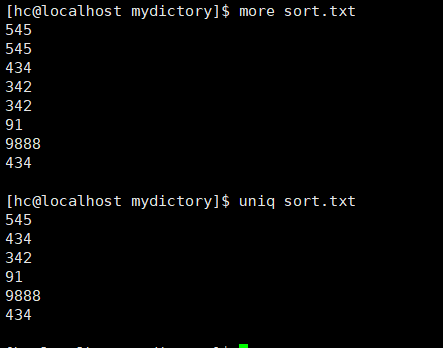
**report or omit repeated lines**

**(报告或省略重复的行)**

**重复行相邻。不相邻不统计显示**



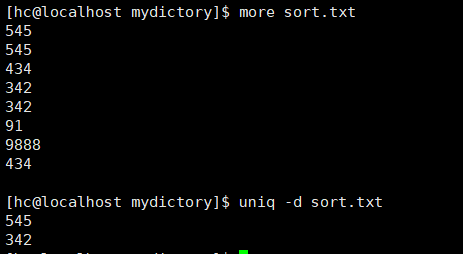
如果使用uniq去掉重复的行，那么两行要相邻。



**-d,** --repeated

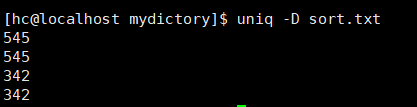
only print duplicate lines

（只打印重复的行）



**-D**, --all-repeated[=delimit-method] print all duplicate lines

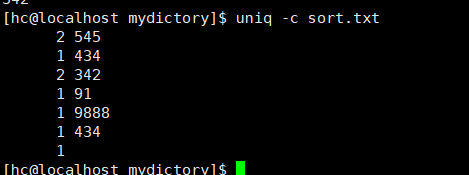
delimit-method={none(default),prepend,separate} Delimiting is done with blank lines.



**-c,** --count

prefix lines by the number of occurrences

（显示某一行重复的次数）



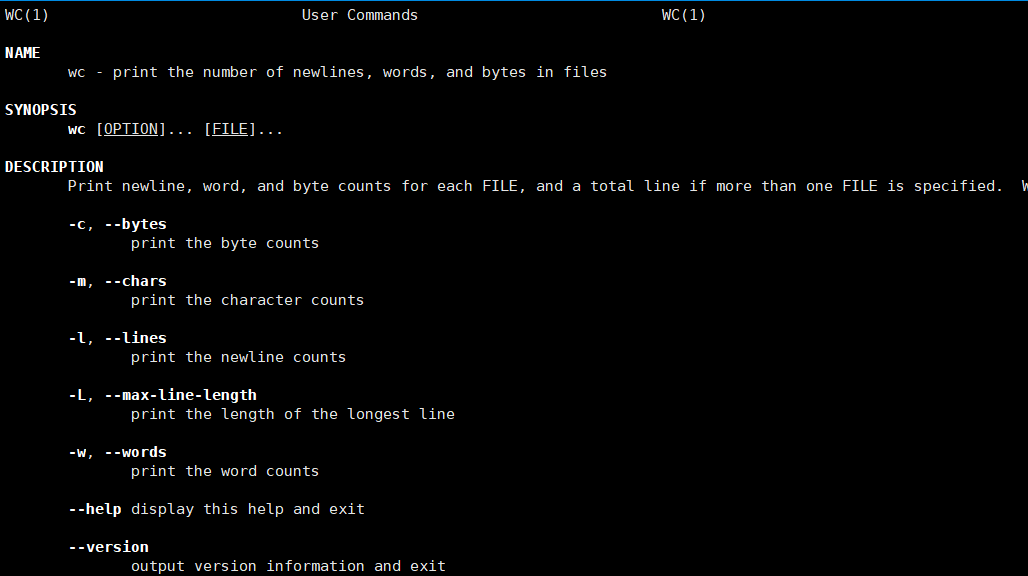
## 文本统计

### wc

word count 文字统计

print the number of newlines,words,and bytes in files

(打印显示多少行，多少个单词，多少字节)



wc passwd



passwd 34行 59个单词 1590个字节。

**-c**, --bytes

print the byte counts

只显示字节数

**-m**, --chars

print the character counts

只显示字符数

**-l,** --lines

print the newline counts

只显示行数

**-w,** --words

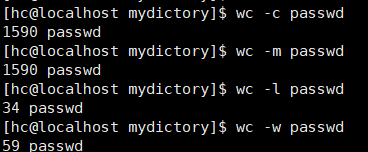
print the word counts

只显示单词数

**-L,** --max-line-length

print the length of the longest line

最长的一行包含了多少个字符



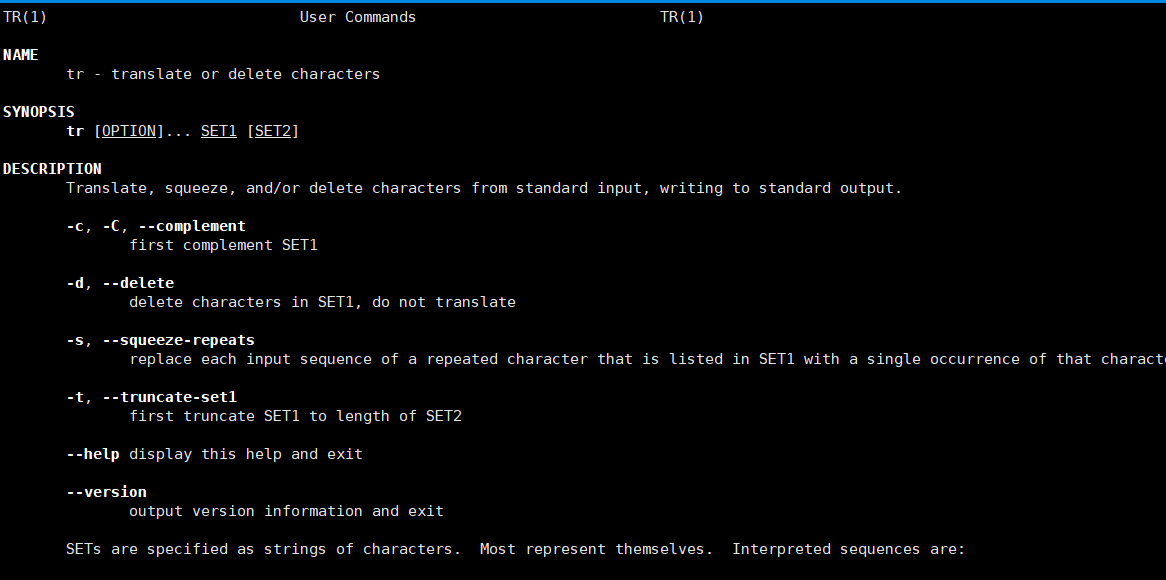


## 字符处理命令

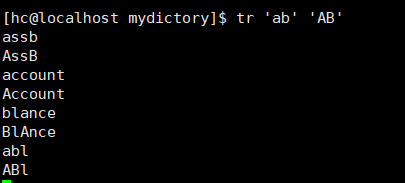
### tr

tr - translate or delete characters

转换或者删除字符

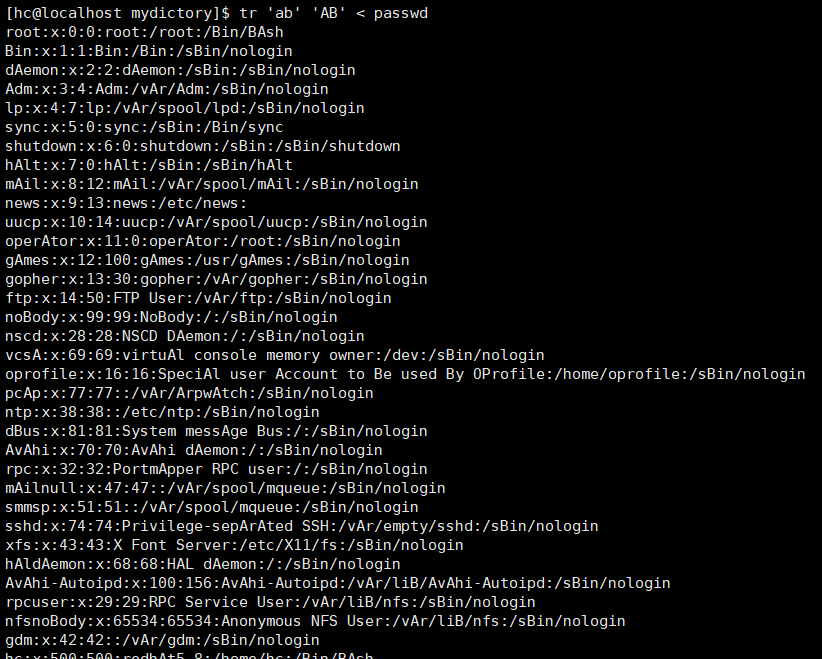


tr ‘ab’ ‘AB’ 将含有a 和 b的字符替换成A 和 B字符



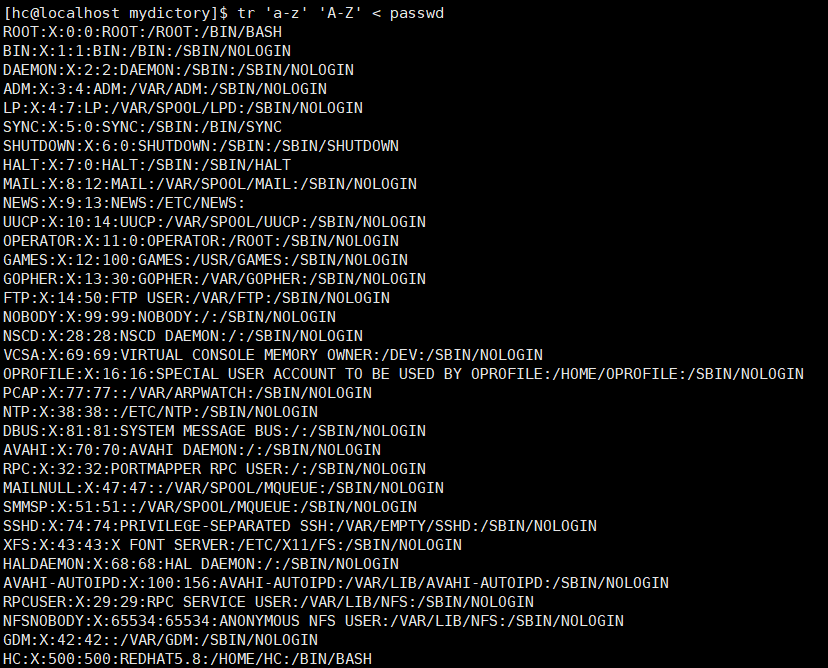
[hc@localhost mydictory]$ tr 'ab' 'AB' < passwd

< 资源重定向



[hc@localhost mydictory]$ tr 'a-z' 'A-Z' < passwd

全部将小写换成大写

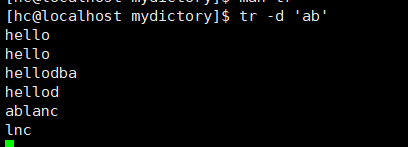


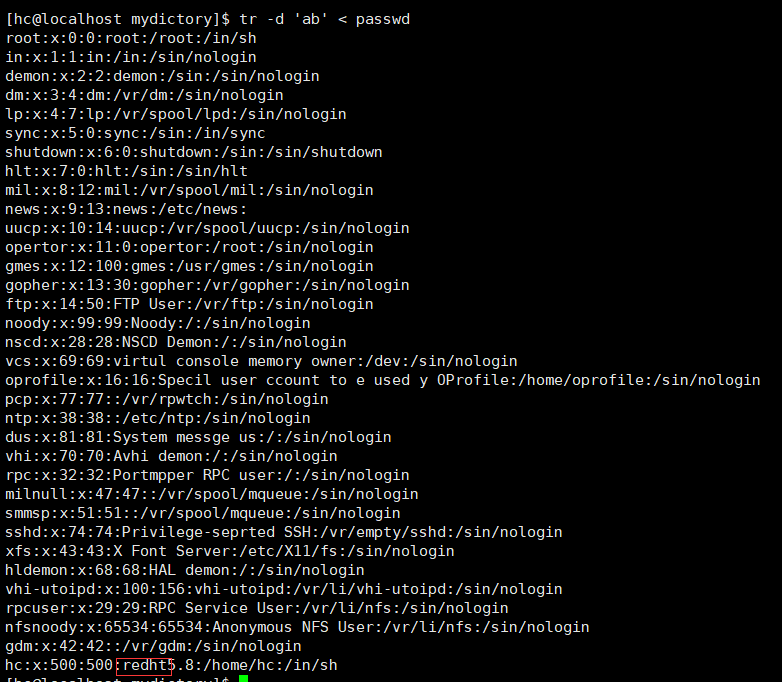
**-d**, --delete

delete characters in SET1, do not translate

删除指定的字符

tr -d ‘ab’ 如果出现ab 就删除





[hc@localhost mydictory]$ tr -d 'a-z' < passwd

