# 文件管理

## 01. diff - compare files line by line

描述：compare files line by line, 以逐行的方式比较给定的两个文本文件的不同处

$ diff a.txt b.txt

1c1

< 123456

---> 123457

**参数**：

-b, –ignore-space-change 不检查空格字符的不同

-B, –ignore-blank-lines 不检查空白行

-i, –ignore-case 不检查大小写的不同

-I, –ignore-matching-lines=RE 忽略RE的匹配

-w, –ignore-all-space 忽略全部空格字符

语法

diff [OPTION]... FILES

## 02. vimdiff

****描述****：:edit two or more file with Vim and show differences, 用vim编辑器同时打开两到四个不同版本的文件，并且在不同之处高亮显示

****语法****

vimdiff [options] file1 file2 [file3 [file4]]

## 03. more

****描述****： file perusal filter for crt viewing . 用来查看一个文本文件的内容，当填满一屏后停下，可以用快捷键来控制显示内容

****快捷键****：

|  |  |
| --- | --- |
| <空格键> 或 f | 显示下一屏文本 |
| <回车键> | 显示下 1 行文本 |
| d 或 ctrl-D | 滚动 k 行 初始 |
| b 或 ctrl-B | 跳过上面 1 屏文本 |
| ’ | 转到上次搜索开始处 |
| = | 显示当前行号 |
| /<正则表达式> | 搜索正则表达式第 1 次出现处 less下会高亮显示 |
| n | 搜索前一正则表达式下次出现处 |
| !<cmd> 或 :!<cmd> | 在子 shell 中执行 命令 |
| v | 在当前行启动 /usr/bin/vi |
| . | 重复前一命令 |

## 04. less

****描述****： less 和more非常像，但是less支持用户向前向后浏览文件，可以用pageup，pagedown 来上下翻页。

****快捷键****

|  |  |
| --- | --- |
| /pattern | Search forward for (N-th) matching line 向后搜索 |
| ?pattern | Search backward for (N-th) matching line 向前搜索 |
| &pattern | Display only matching lines 只显示匹配到的行 |
| n | Repeat previous search (for N-th occurrence) 重复前面的搜索（向后搜） |
| N | Repeat previous search in reverse direction 重复前面的搜索（向前搜） |
| -N | 显示行号 |
| -i | 搜索时忽略大小写 |

## 05. head

****描述****： output the first part of files, 打印出文件的前10（默认）行内容

**参数**：

-c, –bytes=[-]K 打印出前K字节

-n, –lines=[-]K 打印出前K行

****语法****

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

## 06. tail

****描述****： Output the last part of files, 打印文件最后10行的内容

****参数****： 同head  
****语法****

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

## 07. cat

**描述**： Concatenate files and print on the standard output, 连接文件并打印到标准输出设备上，用来显示文件的内容，内容会全部显示。

**参数**：

-n 从1开始显示行号

****语法****

cat [OPTION]... [FILE]...

cat 也可以用来进行输入内容到指定文件，事实上这是”<, «, », >”的功能，

$ cat >>/data/boy/oldboy.txt <<EOFI

am study linux；

EOF

两个EOF可以被两个其它任意内容替换第一个EOF前的\在命令行内需要删除，但是必须相同。上面命令的意思是把I am study linux输入到oldboy.txt里

## 08. xargs

**描述**： build and execute command lines from standard input, 将标准输入数据转换成命令行参数。 也可能将单行或者多行文件输入转换成其它格式。

**参数**：

-n 指定每一行显示的内容的数量

-d 指定以什么为分隔符

****举例****

$ echo "nameXnameXnameXname" | xargs -dX -n2

name name name name

$ find /tmp -name core -type f -print | xargs /bin/rm -f

## 09. seq

**描述**：Print a sequence of numbers 打印从first到last步找为increment连续的整数

**参数**：

-f, –format=FORMAT 使用printf样式的浮点格式

-s, –separator=STRING 指定分隔符 (default: \n)

-w, –equal-width 在前面添加0,使得数字宽度相同

****语法****

seq [OPTION]... LAST seq [OPTION]... FIRST LAST seq [OPTION]... FIRST INCREMENT LAST

****举例****

$ seq -s '< ' 10 $ seq -s '< ' 10 |xargs -n 3

## 10. cut

****描述****：remove sections from each line of files, 删除行中的指定部分，还可以连接文件 cut f1 f2 >f3 便是把f1, f2的内容连接起来输入到f3

参数：

-b: 仅显示行中指定范围字节(bytes)

-c: 仅显示行中指定范围的字符(characters)

-d: 指定分隔符(delimiter)，用指定分隔符来替换TAB(默认）

-f: 只选择指定的范围，

****语法****

cut OPTION... [FILE]...

****举例****

$ who

sandow pts/0 2015-10-21 09:36 (10.0.0.1)

sandow pts/1 2015-10-21 11:27 (10.0.0.1)

$ who |cut -b -5,10

sandop sandop

可以看到-5就是从开关显示到第五，同理5-, 在取英文的时候 -b -c都一样，但是取中文的话只能用-c，用-b便会出现乱码，因为中文一个字占两个字节。

$ cat /etc/passwd|head -5

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

bin:x:1:1:bin:/bin:/sbin/nologin

daemon:x:2:2:daemon:/sbin:/sbin/nologin

adm:x:3:4:adm:/var/adm:/sbin/nologin

lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin

$ cat /etc/passwd|head -5|cut -d : -f 1

root bin daemon adm lp

用-d来指定分隔符后便可以用-f来取出指定范围的内容了，事实上就是取第几列，其中1也可以用上面的情况比如-3 ; 2-4 ; 5- ; 4,7。

## 11. wc

****描述****：print newline, word, and byte counts for each file 统计指定文件中的行数，字数，字节数，并将结果显示输出。如果文件名为“-”那么会从标准输入设备读取数据。

参数：

-l 只显示行数，

-w 只显示字数

-c 只显示字节数 （bytes)

-m 只显示字符数 (character)

****语法****

wc [OPTION]... [FILE]... wc [OPTION]... --files0-from=F

****举例****

$ cat /etc/passwd |wc

26 38 1154

$ cat /etc/passwd |wc -l

26

$ cat /etc/passwd |wc -w

38

$ cat /etc/passwd |wc -c

1154

## 12. chattr

****描述****：change file attributes修改linux文件系统里的文件属性   
****语法****:

chattr [ -RVf ] +-=[acdeijstuADST] [ -v version ] [ mode ] files...

+是给文件增加属性，-是移除文件属性，=是更新指定属性 ‘acdeijstuADST’ 的各自意思是 append only (a), compressed (c), no dump (d), extent format (e), immutable (i), data journalling (j), secure deletion (s), no tail-merging (t), undeletable (u), no atime updates (A), synchronous directory updates (D), synchronous updates (S), and top of directory hierarchy(T).

A 文件或者目录的atime不可修改，这是预防手提电脑I/O错误的发生  
c 设定文件是否通过内核自动压缩后储存  
d 设定文件不能成为dump程序的备份目标  
i 设定文件不允许被修改删除，设定链接。  
u 预防意外删除文件，虽然删除但还是在磁盘中 s 保密性删除文件或目录， 硬盘空间被全部收回

**参数：**

-R 递归创建文件属性

**举例**

chattr +i /etc/passwd /etc/shadow /etc/group /etc/gshadow /etc/inittab lsattr /etc/passwd

----i--------e- /etc/passwd

## 13. lsattr

****描述****：list file attributes on a Linux second extended file system 列出linux EXT2文件系统的属性   
****语法****：

lsattr [ -RVadv ] [ files... ]

**参数：**

-R Recursively list attributes of directories and their contents.递归显示目录以及其包含内容的属性

-V Display the program version.显示程序版本

-a List all files in directories, including files that start with ‘.’ 列出所有目录下的文件，包括以.开头的文件

**举例**

$ lsattr a.txt

-------A-----e- a.txt

## 14. stat

****描述****：display file or file system status 显示文件或者文件系统的状态，包括I-node里的所有信息。

$ stat a.txt

File: "a.txt" Size: 21 Blocks: 8 IO Block: 4096 普通文件

Device: 803h/2051d Inode: 162831 Links: 1

Access: (0645/-rw-r--r-x) Uid: ( 501/ sandow) Gid: ( 501/ sandow)

Access: 2015-10-21 11:26:22.160688313 +0800

Modify: 2015-10-21 11:26:17.751689045 +0800

Change: 2015-10-21 13:02:52.982024877 +0800

**参数：**

-L follow links 支持符号链接

-f display file system status instead of file status显示文件系统而不是文件状态

-c 指定需要显示的信息

-t 以简洁方式输出信息

**语法**

stat [OPTION]… FILE…

****举例****

$ stat -c %a a.txt

645

-c后面的格式 %a access rights in octal 用数字来显示文件的权限（645）  
%A access rights in human readable form(-rw-r–r-x)  
%b number of blocks allocated (see %B)  
%B the size in bytes of each block reported by %b  
%C SELinux security context string  
%d device number in decimal (2051)  
%D device number in hex (803)  
%F file type (普通文件)  
%g group ID of owner(组ID)  
%G group name of owner(组名)  
%h number of hard links  
%i inode number  
%N quoted file name with dereference if symbolic link  
%s total size, in bytes  
%u user ID of owner  
%U user name of owner  
%w time of file birth, human-readable; - if unknown  
%W time of file birth, seconds since Epoch; 0 if unknown  
%x time of last access, human-readable  
%X time of last access, seconds since Epoch  
%y time of last data modification, human-readable  
%Z time of last status change, seconds since Epoch

## 15. chgrp

****描述****：Change the group of each FILE to GROUP. 更改每个文件的GROUP

**参数：**

-c, –changes like verbose but report only when a change is made

****语法****

chgrp [OPTION]... GROUP FILE... chgrp [OPTION]... --reference=RFILE FILE...

****举例****

chgrp staff /u

#Change the group of /u to "staff".

chgrp -hR staff /u

#Change the group of /u and subfiles to "staff".

## 16. chown

****描述****：change file owner and group 更改文件的所有者以及文件所属组

用法用chgrp类似

## 17. chmod

****描述****：change file mode bits 更改文件的权限，文件有读取写入，执行3种权限（rwx），用stat -c %A a.txt 命令里的所看到的内容，第一个字符是文件类型，2-4字符为所有者的权限，5-7是所属组的权限，8-10为其它用户的权限。各缩写必须能够记住 u(user), g(group), o(other), a(all), r(read, 4), w(write, 2), x(execute, 1) -(无权限，0）

**参数：**

-c, –changes 效果类似 -v ，但仅回报更改的部分

-v，–verbose 显示指令执行过程

-R, –recursive

-reference=(参考文件或目录) 把指定文件或目录的所属组全部设定和参考文件或目录的所属组相同

****语法****

chmod [OPTION]... MODE[,MODE]... FILE..

.chmod [OPTION]... OCTAL-MODE FILE...

chmod [OPTION]... --reference=RFILE FILE...

## 18. which

****描述****：shows the full path of (shell) commands 显示命令的绝对路径

****语法****

which [options] [--] programname [...]

## 19. whereis

****描述****： whereis 只能用于程序名的搜索， 会从数据库中查找数据，find是遍历硬盘来查找。

****语法****

whereis [-bmsu] [-BMS directory... -f] filename...

## 20. touch

****描述****：Change file timestamps更改文件的时间戳，若不加参数-c -h，文件不存在便会创建一个文件。   
****语法****:

touch [OPTION]... FILE..

****参数****：

-a change only the access time 改变档案的读取时间记录

-m change only the modification time 改变档案的修改时间记录

-c, –no-create do not create any files

-d, –date=STRING parse STRING and use it instead of current time

-r, –reference=FILE use this file times instead of current time

-t STAMP use [[CC]YY]MMDDhhmm[.ss] instead of current time

****举例****

$ touch testfile

修改文件的时间属性(若不存在)； 创建名为”testfile”的新空白文件

$ touch /tmp/stu{1..12}.txt

批量创建多个文件