<https://blog.csdn.net/Butterfly_resting/article/details/89668744?utm_medium=distribute.pc_relevant_t0.none-task-blog-BlogCommendFromMachineLearnPai2-1.add_param_isCf&depth_1-utm_source=distribute.pc_relevant_t0.none-task-blog-BlogCommendFromMachineLearnPai2-1.add_param_isCf>

14、改变权限：chmod

chmod [-R] xyz 文件或目录

u,g,o表示user,group,other这3种身份的权限，a表示all，即全部的身份。+加入，-除去，=设置。r读，w写，x执行，4读，2写，1执行。

chmod 777 /home/hadoop/a.txt

chmod u+rwx,g+rx,o+r /home/hadoop/a.txt

20、查看文件与目录：ls

ls [-aAdfFhilnrRSt] 目录名称

ls [–color={never,auto,always}] 目录名称

ls [–full-time] 目录名称

-a：全部的文件，连同隐藏文件（开头为.的文件）一起列出来（

-d：仅列出目录本身，而不是列出目录内的文件数据（常用

-f：直接列出结果，而不进行排序(ls默认会以文件名排序)

-F：根据文件、目录等信息给予附加数据结构，例如：\*：代表可执行文件，/：代表目录，=：代表socket文件，|：代表FIFO文件

-h：将文件容量以人类较易读的方式(例如GB，KB等)列出来

-l：列出长数据串，包含文件的属性与权限等数据（常用）

-r：将排序结果反向输出，例如：原本文件名由小到大，反向则为由大到小

-R：连同子目录内容一起列出来，等于该目录下的所有文件都会显示出来

-S：以文件容量大小排序，而不是用文件名排序

-t：依时间排序，而不是用文件名

–color=never：不要依据文件特性给予颜色显示

–color=always：u显示颜色

–color=auto：让系统自行依据设置来判断是否给予颜色

–full-time：以完整时间模式（包含年、月、日、时、分）输出

–time={atime，ctime}：输出访问时间或改变权限属性时间（ctime），而非内容更改时间（modification time）

24、直接查看文件内容：cat（concatenate）

cat [-AbEnTv]

-A：相当于-vET的整合参数，可列出一些特殊字符，而不是空白而已

-n：打印出行号，连同空白行也会有行号，与-b的参数不同

-b：列出行号，仅针对非空白行做行号显示，空白行不标行号

-E：将结尾的断行字符$显示出来

-T：将[Tab]按键以^I显示出来

-v：列出一些看不出来的特殊字符

27、可翻页查看(只向后)：more，按q停止

【<https://blog.csdn.net/qq_15256443/article/details/81664081>】



28、可翻页查看(可向前或向后)：less，按q停止



30、取出后面几行：tail

tail [-n number] 文件

tail -n 20 /home/test

tail -n +100 /home/test //只是列出100行以后的数据

35、从whereis数据库文件中寻找特定的文件：whereis

whereis [-bmsu] 文件或目录

-b：只找二进制格式的文件

-m：只找在说明文件manual路径下的文件

-s：只找source源文件

-u：查找不在上述三个选项当中的其他特殊文件

whereis ifconfig

50、分析一行信息，若匹配想找的数据则将该行取出来：grep

grep [-acinv] [–color=auto] ‘查找字符串’ filename

-a：将binary文件以text文件的方法查找数据

-c：计算找到“查找字符串”的次数

-i：忽略大小写的不同，所以大小写视为相同

-n：顺便输出行号

-v：顺便输出行号

–color=auto：可以将找到的关键字部分加上颜色显示

history | grep ‘ls’

51、排序命令：sort

sort [-fbMnrtuk] [file or stdin]

-f：忽略大小写的差异，例如A与a视为编码相同

-b：忽略最前面的空格符部分

-M：以月份的名字来排序，例如JAN,DEC等的排序方法

-n：使用“纯数字”进行排序（默认是以文字类型来排序的）

-r：反向排序

-u：就是uniq，相同的数据中，仅出现一行代表

-t：分隔符，默认是用[Tab]键来分隔

-k：以那个区间（field）来进行排序的意思

cat /etc/passwd | sort

cat /etc/passwd | sort -t ‘:’ -k 3

52、排序完成但是将重复的数据仅列出一个显示：uniq

uniq [-ic]

-i：忽略大小写字符的不同

-c：进行计数

last | cut -d ’ ’ -f 1 | sort | uniq -c

53、统计文件的字、行、字符：wc

wc [-lwm]

-l：仅列出行

-w：仅列出多少字（英文单字）

-m：多少字符

cat /etc/man.config | wc //输出三个数字分别为行、字数、字符数

55、数据处理工具：awk

sed常常作用于一整行的处理，awk则比较倾向于将一行分成数个“字段”来处理，awk相当适合处理小型的数据处理。

awk ‘条件类型1{动作1} 条件类型2{动作2} … ’ filename

awk 主要是处理每一行的字段内的数据，而默认的字段的分隔符为空格键或[tab]键

awk的处理流程是：

1）读入第一行，并将第一行的数据填入$0,$1,$3等变量当中，$0代表一整行数据

2）依据条件类型的限制，判断是否需要进行后面的动作

3）做完所有的动作与条件类型

4）所还有后续的“行”的数据，则重复上面1～3的步骤，直到所有的数据都读完为止

awk的内置变量：

NF ： 每一行（$0）拥有的字段总数

NR ： 目前awk所处理的是“第几行”数据

FS ： 目前的分隔字符，默认是空格键

awk运算符：

>：大于 <：小于 >=：大于或等于 <=：小于或等于 ：等于 ！=：不等于

last -n 5 | awk ‘{print $1 “\t” $3}’ //打印第一列和第3列

last -n 5 | awk ‘{print $1 "\t lines: " NR “\t columes:” NF}’ //打印第一列，第几行，每一行的列数

cat /etc/passwd | awk '{FS=":"} $3 < 10 {print $1 “\t” KaTeX parse error: Expected 'EOF', got '}' at position 2: 3}̲' //分隔为:，… cat pay.txt | awk 'NR1{printf “%10s %10s %10s %10s %10s\n”,$1,$2,$3,$4,“Total”}

NR>=2{total=$2+$3+$4

printf “%10s %10d %10d %10d %10.2f\n”,$1,$2,$3,$4,total}’

Name 1st 2nd 3th Total

VBird 23000 24000 25000 72000.00

DMTsai 21000 20000 23000 64000.00

Bird2 43000 42000 41000 126000.00