**head命令：**

默认显示头10行。

显示头k行：

head -k /etc/passwd

head -n k /etc/passwd

head -n +k /etc/passwd

显示除最后k行外所有内容：

head -n -k /etc/passwd

**tail命令：**

默认显示最后10行。

显示最后k行：

tail -k /etc/passwd

tail -n k /etc/passwd

tail -n -k /etc/passwd

从开头第k行开始输出：

tail -n +k /etc/passwd

**xargs命令：**

管道是将输出内容重定向到下个命令stdin。

xargs是将输出作为下个命令的参数。

echo "--help" | cat

输出—help。因为—help被放进cat的stdin。

echo "--help" | cat

输出cat的帮助文档。因为—help被放进cat的参数列表。

-d指定分隔符：

echo "11@22" | xargs -d "@" echo -n

输出

11 22。

-n指定每次传几个参数：

echo "11@22@33@44" | xargs -d "@" -n 2 echo

输出：

11 22

33 44

**sort命令：**

默认对第一列按文本升序排序。

-r 按文本方式降序排序

-k 指定排序的列，如: -k 2，按第二列排序

-n 按数字方式排序

-t 指定列分隔符，如: -t:, 指定分隔符为’:’; -t ‘\0’, 指定分割符为NUL

支持按多列排序，如：sort -t . -k 1,1n -k 2,2n -k 3,3n -k 4,4n，给ip地址排序

**uniq命令：**

用于检查及删除文本文件中重复出现的行，当重复的行并不相邻时，uniq 命令是不起作用的，一般与 sort 命令结合使用。

-c或--count 在每列旁边显示该行重复出现的次数

-d或--repeated 仅显示重复出现的行列

-u或--unique 仅显示出一次的行列

-i, --ignore-case 忽略大小写

-s, --skip-chars =N 比较时跳过前N 个字符

-f, --skip-fields=N 比较时跳过前N 列

[**Linux程序前台后台切换**](https://www.cnblogs.com/huanghuanghui/p/9708350.html)**：**

**&**  
加在一个命令的最后，可以把这个命令放到后台执行，如:  
watch -n 10 sh test.sh & #每10s在后台执行一次test.sh脚本

**ctrl + z**  
可以将一个正在前台执行的命令放到后台，并且处于暂停状态。

**jobs**  
查看当前有多少在后台运行的命令  
jobs -l选项可显示所有任务的PID，jobs的状态可以是running, stopped, Terminated。但是如果任务被终止了（kill），shell 从当前的shell环境已知的列表中删除任务的进程标识。

**fg**  
将后台中的命令调至前台继续运行。如果后台中有多个命令，可以用fg %jobnumber（是命令编号，不是进程号）将选中的命令调出。

**bg**  
将一个在后台暂停的命令，变成在后台继续执行。如果后台中有多个命令，可以用bg %jobnumber将选中的命令调出。

**kill**  
法子1：通过jobs命令查看job号（假设为num），然后执行kill %num  
法子2：通过ps命令查看job的进程号（PID，假设为pid），然后执行kill pid  
前台进程的终止：Ctrl+c

**nohup**  
如果让程序始终在后台执行，即使关闭当前的终端也执行（之前的&做不到），这时候需要nohup。该命令可以在你退出帐户/关闭终端之后继续运行相应的进程。关闭中断后，在另一个终端jobs已经无法看到后台跑得程序了，此时利用ps（进程查看命令）

[**rsync命令**](https://www.cnblogs.com/huanghuanghui/p/9708350.html)**：**

rsync 命令是一个远程数据同步工具，可通过 LAN/WAN 快速同步多台主机间的文件。rsync 使用 rsync 算法来使本地和远程两个主机之间的文件达到同步，这个算法只传送两个文件的不同部分，而不是每次都整份传送，因此速度相当快。可以用于本地拷贝，也可用于远端拷贝，用法与scp相似。

<https://www.jianshu.com/p/258ceb7b2223>

|  |  |
| --- | --- |
| -v，--verbose | 详细模式输出。 |
| -q，--quiet | 精简输出模式。 |
| -c，--checksum | 打开校验开关，强制对文件传输进行校验。 |
| -a，--archive | 归档模式，表示以递归方式传输文件，并保持所有文件属性，等于 -rlptgoD。 |

**Free命令：**

$ free -h

total used free shared buffers cached

Mem: 252G 139G 112G 19G 154M 105G

-/+ buffers/cache: 33G 218G

Swap: 0B 0B 0B

第一部分Mem行：

total：内存总数

used：已经使用的内存数

free：空闲的内存数

shared：可用的共享内存

buffers：内存缓冲数

cached：内存缓存数

第二部分:(-/+ buffers/cache)

used: 除去被用作buffers和cache内存后已用的内存

free: 用作buffers和cache的内存加上Mem部分空闲的内存数

第三部分:（Swap）

用一部分磁盘当做内存用的“内存”

**free 与 available 的区别**

free 是真正尚未被使用的物理内存数量。  
available 是应用程序认为可用内存数量，available = free + buffer + cache (注：只是大概的计算方法)

**Buffer与cache的区别：**

cache：文件系统层级的缓存，从磁盘里读取的内容是存储到这里，这样程序读取磁盘内容就会非常快，比如使用grep和find等命令查找内容和文件时，第一次会慢很多，再次执行就快好多倍，几乎是瞬间。但如上所说，如果对文件的更新不关心，就没必要清cache，否则如果要实施同步，必须要把内存空间中的cache clean下

buffer：磁盘等块设备的缓冲，内存的这一部分是要写入到磁盘里的。这种情况需要注意，位于内存buffer中的数据不是即时写入磁盘，而是系统空闲或者buffer达到一定大小统一写到磁盘中，所以断电易失，为了防止数据丢失所以我们最好正常关机或者多执行几次sync命令，让位于buffer上的数据立刻写到磁盘里。

如何释放缓存：

<https://www.cnblogs.com/M18-BlankBox/p/5326484.html>

**查看进程的内存使用：**

top -p 2913

ps aux 第3、4、6列分别是cpu使用率、内存使用率和内存使用量(kB)

cat /proc/{pid}/status的VmRSS列