**Java程序员最常用的Linux命令**

[](http://www.jianshu.com/u/ffd67abc5f8c)

[刘振锋](http://www.jianshu.com/u/ffd67abc5f8c) 关注

2017.05.08 18:50\* 字数 4143 阅读 2379评论 9喜欢 58

大家都知道，Linux系统提供了非常多非常多的命令或工具，这些命令都各有所长，都是系统需要的。但我们精力有限，要掌握全部的命令不太现实，所以只需要掌握其中部分常用的命令即可。这里，我就从一个Java程序员的角度，总结出我常用的一些Linxu命令，供大家参考。

**文件目录基本操作**

* **ls** 命令用来显示目标列表，在Linux中是使用率较高的命令。ls命令的输出信息可以进行彩色加亮显示，以分区不同类型的文件。  
  常用选项：

ls（选项）（参数）

-a 显示所有档案及目录（ls默认不会列出隐藏文件）；

-l 所有输出信息用单列格式输出，不输出为多列；

--color[=WHEN]：使用不同的颜色高亮显示不同类型的。

实际应用时，我经常会使用ls -l --color=auto，并给它指定一个别名ll：

alias ll='ls -l --color=auto'

效果如图1：

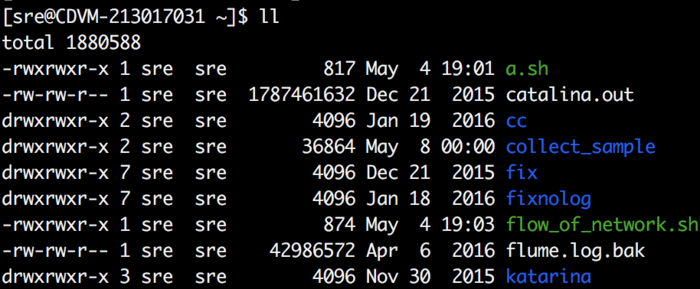


图1

* **cd** 用来切换工作目录至dirname。 其中dirname表示法可为绝对路径或相对路径。若目录名称省略，则变换至使用者的home directory。  
  常用选项：

cd 进入用户主目录

cd ~ 进入用户主目录

cd - 返回进入此目录之前所在的目录

cd .. 返回上级目录（若当前目录为“/“，则执行完后还在“/"；".."为上级目录的意思）

cd ../.. 返回上两级目录

cd !$ 把上个命令的参数作为cd参数使用

* **mkdir** 创建目录。该命令创建由dirname命名的目录。如果在目录名的前面没有加任何路径名，则在当前目录下创建由dirname指定的目录；如果给出了一个已经存在的路径，将会在该目录下创建一个指定的目录。  
  常用选项：

-p 若所要建立目录的上层目录目前尚未建立，则会一并建立上层目录

* **rm** 删除一个目录中的一个或多个文件或目录，也可以将某个目录及其下属的所有文件及其子目录均删除掉。对于链接文件，只是删除整个链接文件，而原有文件保持不变。  
  常用选项：

-f：强制删除文件或目录

-i：删除已有文件或目录之前先询问用户

-r或-R：递归处理，将指定目录下的所有文件与子目录一并处理

**注意：使用rm命令要格外小心。因为一旦删除了一个文件，就无法再恢复它。**

* **cp** 将一个或多个源文件或者目录复制到指定的目的文件或目录。  
  常用选项：

-f：强行复制文件或目录，不论目标文件或目录是否已存在

-i：覆盖既有文件之前先询问用户

-R/r：递归处理，将指定目录下的所有文件与子目录一并处理

示例：

cp /home/sre/tom.log . 将文件/home/sre/tom.log复制到当前目录

cp -r katarina /home/sre/logs/ 将目录katarina复制到/home/sre/logs/目录下

* **mv** 对文件或目录重新命名，或者将文件从一个目录移到另一个目录中。source表示源文件或目录，target表示目标文件或目录。如果将一个文件移到一个已经存在的目标文件中，则目标文件的内容将被覆盖。  
  常用选项：

-f：若目标文件或目录与现有的文件或目录重复，则直接覆盖现有的文件或目录

-u：当源文件比目标文件新或者目标文件不存在时，才执行移动操作

示例：

mv katarina/\* logs/ 将目录katarina下所有文件复制到目录logs下

* **pwd** 以绝对路径的方式显示用户当前工作目录  
  如图2：

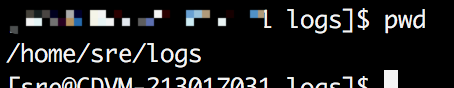


图2

* **tree** 以树状图列出目录的内容  
  如图3

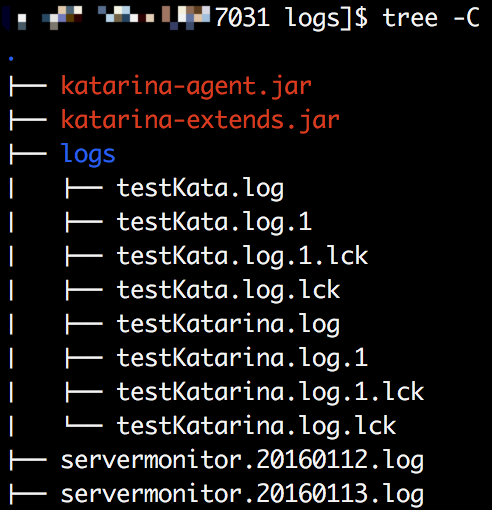


图3

* **touch** 两个功能：一是创建新的空文件；二是用于把已存在文件的时间标签更新为系统当前的时间（默认方式），它们的数据将原封不动地保留下来。  
  示例：

touch /var/wd/logs/touch.txt 创建空文件/var/wd/logs/touch.txt

* **chmod** 变更文件或目录的权限。在UNIX系统家族里，文件或目录权限的控制分别以读取、写入、执行3种一般权限来区分，另有3种特殊权限可供运用。用户可以使用chmod指令去变更文件与目录的权限，设置方式采用文字或数字代号皆可。  
  示例：

chmod +x what\_cpu\_do.sh 给脚本what\_cpu\_do.sh增加可执行权限

chmod u+x,g+w test　　//为文件test设置自己可以执行，组员可以写入的权限

* **file** 探测给定文件的类型。  
  示例：

[sre@CDVM-213017031 ~]$ file what\_cpu\_do.sh

what\_cpu\_do.sh: Bourne-Again shell script text executable

[sre@CDVM-213017031 ~]$ file logs

logs: directory

[sre@CDVM-213017031 ~]$ file network\_last.log

network\_last.log: empty

[sre@CDVM-213017031 ~]$ file fix

fix: directory

**文件内容查看**

* **cat** 连接文件并打印到标准输出设备上，cat经常用来显示文件的内容。  
  示例:

[sre@CDVM-213017031 ~]$ cat flow\_of\_network.sh | grep "RX\_next"

RX\_next=$(cat /proc/net/dev | grep $ethn | sed 's/:/ /g' | awk '{print $2}')

RX=$((${RX\_next}-${RX\_pre}))

* **grep** 是一种强大的文本搜索工具，它能使用正则表达式搜索文本，并把匹配的行打印出来。  
  常用选项：

-A<显示列数> (After)除了显示符合范本样式的那一行之外，并显示该行之后的内容

-B (Before)在显示符合范本样式的那一行之外，并显示该行之前的内容

-C<显示列数> 除了显示符合范本样式的那一列之外，并显示该列之前后的内容

-i 忽略字符大小写。

示例：

[sre@CDVM-213017031 ~]$ cat what\_cpu\_do.sh | grep "process of" -A 5 -B 5 （等效于 cat what\_cpu\_do.sh | grep "jstack" -C 5 ）

echo "checking pid($pid)"

fi

if test -z "$(jps -l | cut -d '' -f 1 | grep $pid)"

then

echo "process of $pid is not exists"

exit

fi

lineNum=$2

if test -z $lineNum

* **tail** 输出文件中的尾部内容，默认在屏幕上显示指定文件的末尾10行。  
  常用选项：

-f：显示文件最新追加的内容（查看服务器上动态刷新的日志时经常使用）。

-n：输出文件的尾部N（N位数字）行内容

示例：

tail -100f /var/wd/logs/owl/owl.20170508.log 显示文件/var/wd/logs/owl/owl.20170508.log中的后100行，并且当文件有更新显示最新的内容。

* **head** 显示文件的开头的内容。在默认情况下，head命令显示文件的头10行内容。  
  常用选项：

-n<数字>：指定显示头部内容的行数

-c<字符数>：指定显示头部内容的字符数

-v：总是显示文件名的头信息

-q：不显示文件名的头信息

* **more** 一个基于vi编辑器文本过滤器，它以全屏幕的方式按页显示文本文件的内容，支持vi中的关键字定位操作。more名单中内置了若干快捷键，常用的有H（获得帮助信息），Enter（向下翻滚一行），空格（向下滚动一屏），Q（退出命令）。  
  常用选项：

Space键：显示文本的下一屏内容

Enter键：只显示文本的下一行内容

斜线符/：接着输入一个模式，可以在文本中寻找下一个相匹配的模式

h键：显示帮助屏，该屏上有相关的帮助信息

b键：显示上一屏内容

q键：退出rnore命令

* **less** 与more十分相似，都可以用来浏览文字档案的内容，不同的是less命令允许用户向前或向后浏览文件，而more命令只能向前浏览。用less命令显示文件时，用PageUp键向上翻页，用PageDown键向下翻页。要退出less程序，应按q键。
* **wc** 用来计算数字。利用wc指令我们可以计算文件的Byte数、字数或是列数。  
  常用选项：

-c：只显示Bytes数

-l：只显示列数

-w：只显示字数

* **tr** 对来自标准输入的字符进行替换、压缩和删除。它可以将一组字符变成另一组字符，经常用来编写优美的单行命令，作用很强大。  
  常用选项：

-c：取代所有不属于第一字符集的字符

-d：删除所有属于第一字符集的字符

-s：把连续重复的字符以单独一个字符表示

示例：

echo "HELLO WORLD" | tr 'A-Z' 'a-z' hello world 将输入字符由大写转换为小写

hello world

echo "hello 123 world 456" | tr -d '0-9' hello world 使用tr删除字符

hello world

* **sort** 将文件进行排序，并将排序结果标准输出。它可以从特定的文件，也可以从stdin中获取输入。  
  常用选项：

-d：排序时，处理英文字母、数字及空格字符外，忽略其他的字符

-f：排序时，将小写字母视为大写字母

-k: 来指定列数

-r：以相反的顺序来排序来自

示例：

[sre@CDVM-213017031 ~]$ ps -mp 25211 -o THREAD,tid,time | sort -r -k 2

USER %CPU PRI SCNT WCHAN USER SYSTEM TID TIME

sre 8.8 - - - - - - 4-04:00:56

sre 5.4 19 - futex\_ - - 25485 2-13:59:26

sre 0.8 19 - futex\_ - - 16755 00:00:24

sre 0.1 19 - ep\_pol - - 25663 01:31:00

sre 0.1 19 - ep\_pol - - 25487 01:38:57

sre 0.1 19 - ep\_pol - - 25486 01:38:52

sre 0.0 19 - skb\_re - - 24586 00:00:00

sre 0.0 19 - poll\_s - - 25678 00:44:56

sre 0.0 19 - poll\_s - - 25676 00:00:00

sre 0.0 19 - poll\_s - - 25213 00:00:01

sre 0.0 19 - futex\_ - - 7679 00:00:00

sre 0.0 19 - futex\_ - - 5952 00:00:00

上述命令功能：查看进程（pid为25211）占用CPU资源最多的线程信息（根据CPU占用率倒序排列）。

**文件查找与比较**

* **find** 在指定目录下查找文件。任何位于参数之前的字符串都将被视为欲查找的目录名。如果使用该命令时，不设置任何参数，则find命令将在当前目录下查找子目录与文件。并且将查找到的子目录和文件全部进行显示。  
  示例：

find /home -name "\*.txt" 在/home目录下查找以.txt结尾的文件名

find /home -iname "\*.txt" 同上，但忽略大小写

find /home ! -name "\*.txt" 找出/home下不是以.txt结尾的文件

* **which** 查找并显示给定命令的绝对路径，环境变量PATH中保存了查找命令时需要遍历的目录。which指令会在环境变量$PATH设置的目录里查找符合条件的文件。也就是说，使用which命令，就可以看到某个系统命令是否存在，以及执行的到底是哪一个位置的命令。  
  示例：

[sre@CDVM-213017031 ~]$ which java

/usr/java/default/bin/java

[sre@CDVM-213017031 ~]$ which pwd

/bin/pwd

[sre@CDVM-213017031 ~]$ which python

/usr/bin/python

* **whereis** 用来定位指令的二进制程序、源代码文件和man手册页等相关文件的路径。  
  和find相比，whereis查找的速度非常快，这是因为linux系统会将 系统内的所有文件都记录在一个数据库文件中，当使用whereis和下面即将介绍的locate时，会从数据库中查找数据，而不是像find命令那样，通 过遍历硬盘来查找，效率自然会很高。  
  示例：

[sre@CDVM-213017031 ~]$ whereis python

python: /usr/bin/python /usr/bin/python2.6 /usr/lib/python2.6 /usr/lib64/python2.6 /usr/include/python2.6 /usr/share/man/man1/python.1.gz

[sre@CDVM-213017031 ~]$ whereis lua

lua: /usr/bin/lua /usr/lib64/lua /usr/share/lua /usr/share/man/man1/lua.1.gz

[sre@CDVM-213017031 ~]$ whereis java

java: /usr/bin/java

* **locate** locate命令其实是find -name的另一种写法，但是要比后者快得多，原因在于它不搜索具体目录，而是搜索一个数据库/var/lib/locatedb，这个数据库中含有本地所有文件信息。Linux系统自动创建这个数据库，并且自动更新，由于更新非实时，所以使用locate命令查不到最新变动过的文件。为了避免这种情况，可以在使用locate之前，先使用updatedb命令，手动更新数据库。  
  示例：

locate /etc/sh 搜索etc目录下所有以sh开头的文件

locate -i ~/m 搜索用户主目录下，所有以m开头的文件，并且忽略大小写

**文件压缩与解压**

* **tar** tar命令可以为linux的文件和目录创建档案。  
  常用选项：

-c：建立新的备份文件

-z：通过gzip指令处理备份文件

-v：显示指令执行过程

-f：指定备份文件

-x：从备份文件中还原文件

示例：

tar -cvf log.tar log2012.log 仅打包，不压缩！

tar -zcvf log.tar.gz log2012.log 打包后，以 gzip 压缩

tar -jcvf log.tar.bz2 log2012.log 打包后，以 bzip2 压缩

* **zip** 可以用来解压缩文件，或者对文件进行打包操作。
* **unzip** 加压缩.zip包，不在详述。  
  另外，关于压缩、解压缩命令还有gzip、gunzip、bzip2、bunzip2等，读者如果感兴趣，可自行搜索了解。

**进程管理**

* **ps** 用于报告当前系统的进程状态，是最基本同时也是非常强大的进程查看命令，使用该命令可以确定有哪些进程正在运行和运行的状态、进程是否结束、进程有没有僵死、哪些进程占用了过多的资源等等，总之大部分信息都是可以通过执行该命令得到的。  
  由于ps命令能够支持的系统类型相当的多，所以选项多的离谱！我主要使用：

[sre@CDVM-213017031 ~]$ ps aux | grep "sudo"

sre 9524 0.0 0.0 103252 832 pts/1 S+ 16:16 0:00 grep sudo

root 19503 0.0 0.0 189992 2996 pts/1 S 13:37 0:00 sudo su - sre

root 22710 0.0 0.0 189992 3000 pts/0 S 11:42 0:00 sudo su - sre

以及上文提到的：

ps -mp 25211 -o THREAD,tid,time | sort -r -k 2

* **kill** 命令用来删除执行中的程序或工作。kill可将指定的信息送至程序。预设的信息为SIGTERM(15),可将指定程序终止。若仍无法终止该程序，可使用SIGKILL(9)信息尝试强制删除程序。程序或工作的编号可利用ps指令或job指令查看。  
  常用选项：

kill -9 强制终止进程

kill -15 正常终止进程

具有相似功能的命令，还有killall、pkill等。

* **watch** 以周期性的方式执行给定的指令，指令输出以全屏方式显示。  
  常用选项：

-n：指定指令执行的间隔时间（秒）

-d：高亮显示指令输出信息不同之处

-t：不显示标题。

示例：

watch ss

Every 2.0s: ss Mon May 8 16:23:58 2017

State Recv-Q Send-Q Local Address:Port Peer Address:Port

ESTAB 0 0 ::ffff:10.213.17.31:45473 ::ffff:10.213.18.49:10994

ESTAB 0 0 ::ffff:10.213.17.31:30046 ::ffff:10.209.19.143:10117

ESTAB 0 0 ::ffff:10.213.17.31:11233 ::ffff:10.209.26.154:10118

ESTAB 0 0 ::ffff:10.213.17.31:55524 ::ffff:10.209.33.69:eforward

ESTAB 0 0 ::ffff:10.213.17.31:56865 ::ffff:10.209.18.34:10218

ESTAB 0 0 ::ffff:10.213.17.31:25608 ::ffff:10.209.19.144:10117

ESTAB 0 0 ::ffff:10.213.17.31:19649 ::ffff:10.209.18.64:10620

* **service** service命令是Redhat Linux兼容的发行版中用来控制系统服务的实用工具，它以启动、停止、重新启动和关闭系统服务，还可以显示所有系统服务的当前状态。  
  服务名：自动要控制的服务名，即/etc/init.d目录下的脚本文件名  
  示例：

service mysqld status

mysqld (pid 1638) 正在运行...

service mysqld restart

停止 MySQL： [ 确定 ]

启动 MySQL： [ 确定 ]

**网络操作**

* **curl** curl命令是一个利用URL规则在命令行下工作的文件传输工具。它支持文件的上传和下载，所以是综合传输工具，但按传统，习惯称curl为下载工具。
* **wget** wget命令用来从指定的URL下载文件。  
  示例：

wget http://www.yourserverip.net/testfile.zip 下载文件

* **ping** 用来测试主机之间网络的连通性。

[sre@CDVM-213017031 ~]$ ping www.wanda.cn

PING www.wanda.cn (10.199.8.10) 56(84) bytes of data.

64 bytes from 10.199.8.10: icmp\_seq=1 ttl=244 time=50.8 ms

64 bytes from 10.199.8.10: icmp\_seq=2 ttl=244 time=51.8 ms

64 bytes from 10.199.8.10: icmp\_seq=3 ttl=244 time=51.0 ms

64 bytes from 10.199.8.10: icmp\_seq=4 ttl=244 time=53.3 ms

* **telnet** 用于登录远程主机，对远程主机进行管理。我常用它来检测端口。  
  示例：

[sre@CDVM-213017031 ~]$ telnet 10.213.17.32 10000

Trying 10.213.17.32...

telnet: connect to address 10.213.17.32: Connection refused

[sre@CDVM-213017031 ~]$ telnet 10.213.17.32 10062

Trying 10.213.17.32...

Connected to 10.213.17.32.

Escape character is '^]'.

^C

Connection closed by foreign host.

* **nslookup** 常用域名查询工具，就是查DNS信息用的命令。  
  示例：

[sre@CDVM-213017031 ~]$ nslookup www.baidu.com

Server: 10.209.11.13

Address: 10.209.11.13#53

Non-authoritative answer:

www.baidu.com canonical name = www.a.shifen.com.

Name: www.a.shifen.com

Address: 111.206.223.205

Name: www.a.shifen.com

Address: 111.206.223.206

* **ss** 用来显示处于活动状态的套接字信息。ss命令可以用来获取socket统计信息，它可以显示和netstat类似的内容。但ss的优势在于它能够显示更多更详细的有关TCP和连接状态的信息，而且比netstat更快速更高效。  
  常用选项：

-n：不解析服务名称，以数字方式显示

-a：显示所有的套接字

-l：显示处于监听状态的套接字

-m：显示套接字的内存使用情况

-p：显示使用套接字的进程信息

-i：显示内部的TCP信息；

-t：只显示tcp套接字

-u：只显示udp套接字

示例：

[sre@CDVM-213017031 ~]$ ss -t

State Recv-Q Send-Q Local Address:Port Peer Address:Port

ESTAB 0 0 ::ffff:10.213.17.31:45473 ::ffff:10.213.18.49:10994

ESTAB 0 0 ::ffff:10.213.17.31:14073 ::ffff:10.209.18.33:10620

ESTAB 0 0 ::ffff:10.213.17.31:30046 ::ffff:10.209.19.143:10117

ESTAB 0 0 ::ffff:10.213.17.31:11233 ::ffff:10.209.26.154:10118

ESTAB 0 0 ::ffff:10.213.17.31:20347 ::ffff:10.209.18.64:10620

ESTAB 0 0 ::ffff:10.213.17.31:41799 ::ffff:10.209.18.34:10620

ESTAB 0 0 ::ffff:10.213.17.31:39386 ::ffff:10.209.33.70:10434

ESTAB 0 0 ::ffff:10.213.17.31:55524 ::ffff:10.209.33.69:eforward

ESTAB 0 0 ::ffff:10.213.17.31:56865 ::ffff:10.209.18.34:10218

* **nc** nc是netcat命令的简称，都是用来设置路由器。我常用它来上传文件到服务器，具体可参考 [如何优雅的实现文件上传或下载](https://www.jianshu.com/p/fca34c2bf7f1)
* **ifconfig** 用于配置和显示Linux内核中网络接口的网络参数。用ifconfig命令配置的网卡信息，在网卡重启后机器重启后，配置就不存在。要想将上述的配置信息永远的存的电脑里，那就要修改网卡的配置文件了。  
  示例：

[sre@CDVM-213017031 ~]$ ifconfig

eth0 Link encap:Ethernet HWaddr FA:16:3E:7E:55:D1

inet addr:10.213.17.31 Bcast:10.213.23.255 Mask:255.255.248.0

inet6 addr: fe80::f816:3eff:fe7e:55d1/64 Scope:Link

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:31875618113 errors:0 dropped:0 overruns:0 frame:0

TX packets:28230970908 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:1000

RX bytes:13949365053248 (12.6 TiB) TX bytes:5198483437608 (4.7 TiB)

ifconfig eth0 down 关闭网卡eth0

ifconfig eht0 up 启动网卡eth0

**系统管理**

* **su** 用于切换当前用户身份到其他用户身份，变更时须输入所要变更的用户帐号与密码。
* **sudo** 以其他身份来执行命令，预设的身份为root。在/etc/sudoers中设置了可执行sudo指令的用户。
* **du** 查看使用空间。  
  常用选项：

-a 显示目录中个别文件的大小

-b 显示目录或文件大小时，以byte为单位

-c 除了显示个别目录或文件的大小外，同时也显示所有目录或文件的总和

-k 以KB(1024bytes)为单位输出

-m 以MB为单位输出

-s 仅显示总计，只列出最后加总的值

-h 以K，M，G为单位，提高信息的可读性

示例：

[sre@CDVM-213017031 ~]$ du -h \*

4.0K a.sh

1.7G catalina.out

317M collect\_sample

632K fix/plugin

372K fix/boot

8.0K fix/script

41G fix/log

6.1M fix/lib

**性能监测与优化**

* **top** 可以实时动态地查看系统的整体运行情况，是一个综合了多方信息监测系统性能和运行信息的实用工具。通过top命令所提供的互动式界面，用热键可以管理。  
  常用选项：

-d：屏幕刷新间隔时间

-u<用户名>：指定用户名

-p<进程号>：指定进程

-n<次数>：循环显示的次数

交互命令：

1：显示全部CPU信息

k：终止一个进程

i：忽略闲置和僵死进程，这是一个开关式命令

q：退出程序

o或者O：改变显示项目的顺序

m：切换显示内存信息

t：切换显示进程和CPU状态信息

c：切换显示命令名称和完整命令行

M：根据驻留内存大小进行排序

P：根据CPU使用百分比大小进行排序

T：根据时间/累计时间进行排序

示例：

top -p 25211

top - 17:14:06 up 559 days, 4:54, 2 users, load average: 0.06, 0.01, 0.00

Tasks: 1 total, 0 running, 1 sleeping, 0 stopped, 0 zombie

Cpu0 : 0.0%us, 0.3%sy, 0.0%ni, 99.7%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st

Cpu1 : 0.3%us, 0.3%sy, 0.0%ni, 99.3%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st

Cpu2 : 0.3%us, 0.3%sy, 0.0%ni, 99.3%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st

Cpu3 : 6.8%us, 3.4%sy, 0.0%ni, 89.8%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st

Mem: 4054588k total, 3916560k used, 138028k free, 114260k buffers

Swap: 0k total, 0k used, 0k free, 730196k cached

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND

25211 sre 20 0 4624m 1.9g 9m S 12.7 49.7 6014:18 java

上面是执行 top -p 25211后，单击“1”后的效果。

* **free** 显示当前系统未使用的和已使用的内存数目，还可以显示被内核使用的内存缓冲区。  
  常用选项：

-b：以Byte为单位显示内存使用情况

-k：以KB为单位显示内存使用情况

-m：以MB为单位显示内存使用情况

-t：显示内存总和列

示例：

[sre@CDVM-213017031 ~]$ free -m

total used free shared buffers cached

Mem: 3959 3825 134 0 111 713

-/+ buffers/cache: 3000 959

Swap: 0 0 0

* **sar** Linux下系统运行状态统计工具，它将指定的操作系统状态计数器显示到标准输出设备。sar工具将对系统当前的状态进行取样，然后通过计算数据和比例来表达系统的当前运行状态。它的特点是可以连续对系统取样，获得大量的取样数据。取样数据和分析的结果都可以存入文件，使用它时消耗的系统资源很小。我常用它查看网卡流量，具体请参考 [Linux查看实时网卡流量的几种方式](https://www.jianshu.com/p/b9e942f3682c)。
* **lsof** lsof命令用于查看你进程开打的文件，打开文件的进程，进程打开的端口(TCP、UDP)。  
  常用选项：

-a：列出打开文件存在的进程

-c<进程名>：列出指定进程所打开的文件

-p<进程号>：列出指定进程号所打开的文件

示例：

[sre@CDVM-213017031 ~]$ lsof | wc -l

1278

[sre@CDVM-213017031 ~]$ lsof | grep 10117 | wc -l

9

[sre@CDVM-213017031 ~]$ lsof | grep 10117

java 402 sre 143u IPv6 1879294557 0t0 TCP CDVM-213017031:63449->10.209.19.143:10117 (ESTABLISHED)

java 402 sre 144u IPv6 1879294563 0t0 TCP CDVM-213017031:63450->10.209.19.143:10117 (ESTABLISHED)

java 402 sre 145u IPv6 1879294564 0t0 TCP CDVM-213017031:25608->10.209.19.144:10117 (ESTABLISHED)

java 11538 sre 43u IPv6 720753145 0t0 TCP CDVM-213017031:30045->10.209.19.143:10117 (ESTABLISHED)

java 11538 sre 44u IPv6 720753147 0t0 TCP CDVM-213017031:30046->10.209.19.143:10117 (ESTABLISHED)

java 11538 sre 45u IPv6 720762177 0t0 TCP CDVM-213017031:47599->10.209.19.144:10117 (ESTABLISHED)

java 25211 sre 267u IPv6 1855869958 0t0 TCP CDVM-213017031:54013->10.209.19.143:10117 (ESTABLISHED)

java 25211 sre 271u IPv6 1855869959 0t0 TCP CDVM-213017031:16169->10.209.19.144:10117 (ESTABLISHED)

java 25211 sre 294u IPv6 1855870096 0t0 TCP CDVM-213017031:54030->10.209.19.143:10117 (ESTABLISHED)

当程序报too many open files 异常时，可以使用它查看是什么进程打开了太多的文件，主要查看是不是文件（包括网络链接）打开后是不是没有关闭，我之前的一个项目，就遇到了这种问题。

* **ulimit** 用来限制系统用户对shell资源的访问。支持以下各种类型的限制：所创建的内核文件的大小、进程数据块的大小、Shell 进程创建文件的大小、内存锁住的大小、常驻内存集的大小、打开文件描述符的数量、分配堆栈的最大大小、CPU 时间、单个用户的最大线程数、Shell 进程所能使用的最大虚拟内存。同时，它支持硬资源和软资源的限制。  
  常用选项：

-a：显示目前资源限制的设定

-n <文件数目>：指定同一时间最多可开启的文件数

-u <程序数目>：用户最多可开启的程序数目

示例：

[sre@CDVM-213017031 ~]$ ulimit -a

core file size (blocks, -c) 0

data seg size (kbytes, -d) unlimited

scheduling priority (-e) 0

file size (blocks, -f) unlimited

pending signals (-i) 31517

max locked memory (kbytes, -l) 64

max memory size (kbytes, -m) unlimited

open files (-n) 409600

pipe size (512 bytes, -p) 8

POSIX message queues (bytes, -q) 819200

real-time priority (-r) 0

stack size (kbytes, -s) 10240

cpu time (seconds, -t) unlimited

max user processes (-u) 65535

virtual memory (kbytes, -v) unlimited

file locks (-x) unlimited

* **vmstat** 显示虚拟内存状态（“Viryual Memor Statics”），但是它可以报告关于进程、内存、I/O等系统整体运行状态。  
  常用选项：

-a：显示活动内页

-f：显示启动后创建的进程总数

-m：显示slab信息

-n：头信息仅显示一次

-s：以表格方式显示事件计数器和内存状态

-d：报告磁盘状态

-p：显示指定的硬盘分区状态

-S：输出信息的单位。

示例：

[sre@CDVM-213017031 ~]$ vmstat 2

procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu-----

r b swpd free buff cache si so bi bo in cs us sy id wa st

0 0 0 126316 114292 736992 0 0 1 4 0 0 1 1 98 0 0

1 0 0 126432 114292 736992 0 0 0 0 2289 4432 2 1 97 0 0

0 0 0 126480 114292 736992 0 0 0 0 2572 5132 2 1 97 0 0

* **iostat** 监视系统输入输出设备和CPU的使用情况。它的特点是汇报磁盘活动统计情况，同时也会汇报出CPU使用情况。同vmstat一样，iostat也有一个弱点，就是它不能对某个进程进行深入分析，仅对系统的整体情况进行分析。  
  常用选项：

-c：仅显示CPU使用情况

-d：仅显示设备利用率

-k：显示状态以千字节每秒为单位，而不使用块每秒

-m：显示状态以兆字节每秒为单位

-x：显示扩展状态

示例：

[sre@CDVM-213017031 ~]$ iostat -x 1 2

Linux 2.6.32-431.el6.x86\_64 (CDVM-213017031) 05/08/2017 \_x86\_64\_ (4 CPU)

avg-cpu: %user %nice %system %iowait %steal %idle

0.94 0.00 0.53 0.10 0.02 98.41

Device: rrqm/s wrqm/s r/s w/s rsec/s wsec/s avgrq-sz avgqu-sz await svctm %util

vda 0.01 2.69 0.11 1.30 9.02 32.13 29.09 0.05 34.90 5.07 0.72

avg-cpu: %user %nice %system %iowait %steal %idle

2.77 0.00 2.27 0.00 0.25 94.71

Device: rrqm/s wrqm/s r/s w/s rsec/s wsec/s avgrq-sz avgqu-sz await svctm %util

vda 0.00 1.00 0.00 3.00 0.00 32.00 10.67 0.01 4.00 1.33 0.40

**Java常用工具**

* **java** 可用来执行jar包。  
  示例:

java -jar test.jar

* **jps** jps是jdk提供的一个查看当前Java进程的小工具， 可以看做是JavaVirtual Machine Process Status Tool的缩写。非常简单实用。  
  常用选项：

-l：输出完全的包名，应用主类名，jar的完全路径名

-v：输出jvm参数

示例：

[sre@CDVM-213017031 ~]$ jps -l

402 org.apache.flume.node.Application

13466 sun.tools.jps.Jps

11538 com.wanda.monitor.server.MonitorStartup

25211 org.apache.catalina.startup.Bootstrap

* **jmap** 打印出某个java进程内存中所有‘对象’的情况。  
  常用选项：

-dump:[live,]format=b,file=<filename> 使用hprof二进制形式,输出jvm的heap内容到文件=. live子选项是可选的，假如指定live选项,那么只输出活的对象到文件.

-heap 打印heap的概要信息，GC使用的算法，heap的配置及wise heap的使用情况.

-histo[:live] 打印每个class的实例数目,内存占用,类全名信息. VM的内部类名字开头会加上前缀”\*”. 如果live子参数加上后,只统计活的对象数量.

示例1，查看进程内存信息：

[sre@CDVM-213017031 ~]$ jmap -heap 25211

Attaching to process ID 25211, please wait...

Debugger attached successfully.

Server compiler detected.

JVM version is 24.65-b04

using parallel threads in the new generation.

using thread-local object allocation.

Concurrent Mark-Sweep GC

Heap Configuration:

MinHeapFreeRatio = 40

MaxHeapFreeRatio = 70

MaxHeapSize = 2147483648 (2048.0MB)

NewSize = 348913664 (332.75MB)

MaxNewSize = 348913664 (332.75MB)

OldSize = 697892864 (665.5625MB)

NewRatio = 2

SurvivorRatio = 8

...

示例2，查看包com.wanda中的类创建的对象占用内存信息：

[sre@CDVM-213017031 ~]$ jmap -histo 25211 | grep "com.wanda" | head -5

num #instances #bytes class name

----------------------------------------------

46: 4167 500040 com.wanda.arch.owl.domain.monitorelement.MIndicator

63: 8192 327680 com.wanda.fix.org.jboss.netty.util.internal.ConcurrentIdentityHashMap$Segment

73: 8192 262192 [Lcom.wanda.fix.org.jboss.netty.util.internal.ConcurrentIdentityHashMap$HashEntry;

102: 1428 148512 com.wanda.arch.owl.domain.monitorelement.LComponent

138: 2048 98304 com.wanda.fix.org.jboss.netty.util.internal.ConcurrentIdentityHashMap

* **jstat** Jstat用于监控基于HotSpot的JVM，对其堆的使用情况进行实时的命令行的统计，使用jstat我们可以对指定的JVM做如下监控：  
  类的加载及卸载情况，查看新生代、老生代及持久代的容量及使用情况，查看新生代、老生代及持久代的垃圾收集情况，包括垃圾回收的次数及垃圾回收所占用的时间，查看新生代中Eden区及Survior区中容量及分配情况等。  
  常用选项：

-gcutil 用于查看新生代、老生代及持代垃圾收集的情况

-class 用于查看类加载情况的统计

示例：

[sre@CDVM-213017031 ~]$ jstat -gcutil 25211 5000 2

S0 S1 E O P YGC YGCT FGC FGCT GCT

0.00 4.82 82.82 58.15 60.00 197263 2435.605 72 1.186 2436.791

5.36 0.00 28.22 58.16 60.00 197264 2435.620 72 1.186 2436.806

说明：

S0 Heap上的 Survivor space 0 区已使用空间的百分比

S1 Heap上的 Survivor space 1 区已使用空间的百分比

E Heap上的 Eden space 区已使用空间的百分比

O Heap上的 Old space 区已使用空间的百分比

P Perm space 区已使用空间的百分比

YGC 从应用程序启动到采样时发生 Young GC 的次数

YGCT 从应用程序启动到采样时 Young GC 所用的时间(单位秒)

FGC 从应用程序启动到采样时发生 Full GC 的次数

FGCT 从应用程序启动到采样时 Full GC 所用的时间(单位秒)

GCT 从应用程序启动到采样时用于垃圾回收的总时间(单位秒)，它的值等于YGC+FGC

* **jstack** 用于打印出给定的java进程ID或core file或远程调试服务的Java堆栈信息。  
  示例：

jstack 25211 > 25211.txt 将当前堆栈信息输出到文件。

以上就是我常用的Linux命令，可能有一些常用的一时没有想到，等到想到时再补充进来吧。  
**另外，文中 Java常用工具 中提到的几个命令，尤其重要，尤其常用，特别是当你的程序上线以后，它能方便而快速的帮你定位问题，解决问题。**