## 命令

## -chmod（修改文件、目录的rwx权限）

### chmod -R ugo+w ./xxx

-R：给目录下所有文件添加权限

ugo：u 当前用户；g 同组用户；o其他组用户；a 所有用户

+w：添加读权限

### chmod 751 xxx.txt

7：111当前用户rwx权限

5：101同组用户r\_x权限

1：001其他组用户\_\_x权限

## -chgrp（修改文件、目录的群组）

### chgrp -R aaa ./xxx

将xxx目录下的所有文件和目录的群组设为aaa

### chgrp --reference=aaa.log bbb.log

修改bbb.loog文件的群组，设为和aaa文件一样

## -chown（修改文件、目录的用户、群组）

### chown -R aaa:bbb ./xxx

将xxx目录下的所有文件和目录的用户和群组设为aaa和bbb

## -useradd（添加用户）

### useradd -g bbb -d /aaa aaa

创建名为aaa的用户，群组为bbb，用户主目录是/aaa

## -usermod（修改用户的群组）

### usermod -g aaa bbb

将用户bbb的群组修改成aaa

## -userdel（删除用户）

### userdel xxx

删除xxx用户，但是保留它的家目录

### userdel -r xxx

删除xxx用户，它的家目录也一起删除

### Id xxx

查看xxx的用户信息：uid、gid、组

### cat /etc/passwd

用户的配置文件。查看用户名，组id，家目录等

用户名：口令：用户标识号：组标识号：注释性描述：主目录：登录shell

### cat /etc/shadow

口令的配置文件

登录名：加密口令：最后一次修改时间：最小时间间隔：最大时间间隔：警告时间：不活动时间：失效时间：标志

### cat /etc/group

组的配置文件

组名：口令：组标识号：组内用户列表

### groups xxx

查看xxx群组下的用户

### passwd xxx

给xxx用户指定密码

## -vim

### 5yy p

复制当前行向下的5行，粘贴

### 5dd

删除当前行向下的5行

### /xxx n

查找xxx，n查找下一个

### :set nu :set nonu

显示行号 取消行号

### gg G

到达文档首行 到达文档末行

### u ctrl+r

撤销上次输入 取消撤销

### 40 shift+g

在显示行号的情况下，先按行号，再按shift+g，则跳转到指定行

## -iptables

### iptables -L -n

-L：列表形式显示

-n：IP、端口以数字显示

### iptables -F

-F：清空所有链

### iptables -nL --line-number

显示编号

### iptables -I INPUT 3 -p tcp -s 1.1.1.1 --dport 8080 -j ACCEPT

将这条规则放到编号3的地方

## -mkdir

### mkdir -p /aaa/bbb

-p：创建多级目录

## -rmdir

### rmdir xxx.txt

删除空目录

## -rm

### rm -rvfi /aaa/bbb

r：递归删除

v：显示删除过程

f：强制删除

i：交互，提示是否删除

## -cp

### cp -r source dest

r：复制整个目录（包括文件也包括目录结构）

### cp source/\* dest

将source下的文件复制到dest，不包括source目录，如果source下有子目录，需要加-r

## -cat

### cat -n xxx.txt | more(less)

n：显示行号

More(less)：管道连接使用more或者less显示

## -more

ctrl+f：向下翻页

ctrl+b：向上翻页

q：退出

## -less

比more强大，适合显示大型文件。

[pagedown]：向下翻页

[pageup]：向上翻页

**/**xxx：向下查找xxx，**n** 向下查找（同向），**N**向上查找（反向）

?xxx：向上查找xxx，n 向上查找，N向下查找

q：退出

## - >

ll > xxx.txt

ll输出的内容会将xxx.txt中内容覆盖掉（删除）

## - >>

ll >> xxx.txt

ll输出的内容追加到xxx.txt中

## -echo

### echo $PATH

输出linux环境变量。$PATH是/etc/profile中配置的

## -head

### head -n 100 xxx.txt

显示xxx.txt前100行

## -tail

### tail -n 100 xxx.txt

显示xxx.txt后100行

### tail -100f xxx.txt

f：实时刷新显示xxx.txt

## -ln

### ln -s source name

s：软连接

source：源文件或目录

name：链接名

软连接：cd进入到软连接下，pwd显示的就是软连接的路径，而不是源目录的路径。删除时，不要在链接名后面加 /，rm -rf name即可。

## -history

### history 10

10：显示最近10条命令

### !25

直接执行序号为25的指令

## -find

### find / -name hiverc

-name：按照文件名查找

### find / -user root

-user：按照用户名查找

### find / -size +20M -20M 20M \*.txt

-size：按照文件大小查找

+20M：比20m大的 -20M：比20M小的 20M：等于20M的

## -locate

### locate xx.txt

第一次使用locate命令前需要先执行updatedb。

## -tar

### tar -zxvf xx.tar.gz -C ../

z：解压.gz格式

x：解压

v：显示解压过程

f：指定文件

-C：指定解压位置，目录需存在

### tar -zcvf xx.tar.gz aa.txt bb.txt

-c：压缩成tar.gz

## -zip

-r：压缩目录

## -unzip

-d：指定解压后文件的存放目录

## -gzip

压缩成.gzip文件，且不会保留源文件

**压缩后保存源文件：gzip -c 源文件 > 压缩文件**

## -gunzip

解压.gzip文件，且不会保留源压缩包

**解压后保存源压缩文件：gzip -d 压缩文件**

## -cat emp.txt | awk -F ',' '{print $4}' | sort | uniq -c | tee bbb

## -awk

-F ‘,’：修改分隔符为’,’

‘{print $4} ’：打印每行第4个元素

## -sort

排序

## -tee

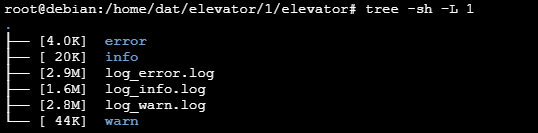
读取数据，输出到文件

## -tree

debian下安装tree：**apt-get install tree**

tree -L 1 目录显示一级

tree -sh 列出文件或者目录大小，



## -du

### du -sh

-s：查看目录总容量

-a：查看所有文件

### du -h --max-depth=1

--max-depth=1：查看目录深度为1的目录大小

## -df

查看磁盘分区上的磁盘使用情况

### sysctl -w vm.max\_map\_count=262144

单个JVM的最大线程数，在/etc/sysctl.conf下配置vm.max\_map\_count=262144，重启也生效

## -lsof -i:端口

lsof （list open files：列出当前系统打开文件，Linux中任何事物都是以文件的形式存在）

## -netstat -tuanpl | grep

-t 查看tcp协议的端口情况

-u 查看udp协议的端口情况

-a 查看所有**选项**

-n 数字代替域名显示

-p 显示端口占用程序信息

-l 显示LISTENER状态的连接

列：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Proto | Recv-Q | Send-Q | Local Address | Foreign Address | State | 占用端口的程序 |
| 协议 | 接收数据 | 发送数据 | 本地地址:端口 | 外部地址:端口 | 端口状态  LISTEN  CLOSE\_WAIT等 | 15129/java  进程ID为15129的java程序 |

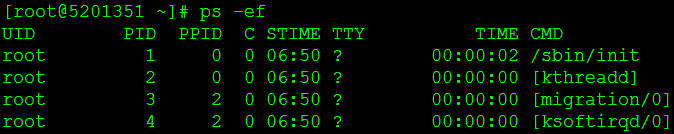
https://www.cnblogs.com/dadonggg/p/8778318.html

查看TCP连接数

**netstat -ant|awk '/^tcp/ {++S[$NF]} END {for(a in S) print (a,S[a])}'**

## -ps

### ps -ef



UID：用户名

PID：进程ID

PPID：父进程ID

C：进程占用CPU百分比

STIME：进程启动到现在的时间

**TTY**：该进程在哪个终端上运行，*若与终端无关*（啥叫和终端无关？？）显示问号？

CMD：进程执行的命令

### ps -aux

USER：用户名

PID：进程ID

**%CPU**：进程占用的CPU百分比

**%MEM**：进程占用的内存百分比

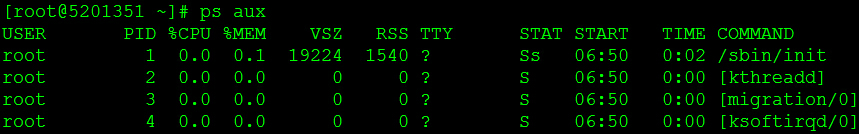
**VSZ**：该进程使用的虚拟内存量（KB）

**RSS**：该进程使用的固定内存量（KB）

**STAT**：进程状态 S休眠 R 运行

START：进程触发执行时间

TIME：进程实际使用CPU的时间



### 查找僵尸进程

ps -A -ostat,ppid,pid,cmd |grep -e '^[Zz]'

grep -e 表示 or操作

^[Zz]表示以Z或z开头

## nohup 和 &

### nohup

1. 默认输出到nohup.out文件
2. ctrl+c发送SIGINT信号，程序会关闭
3. 关闭session窗口发送SIGHUP信号，程序免疫

### &

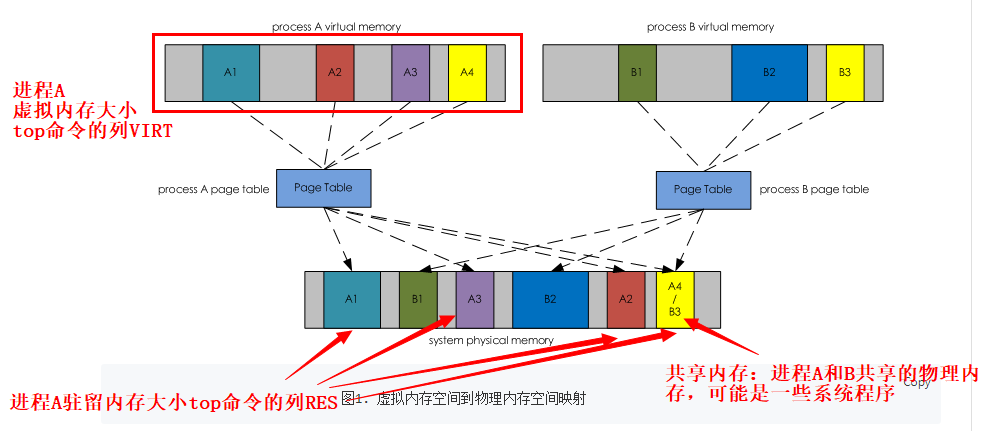
1. 默认输出到终端
2. ctrl+c发送SIGINT信号，程序免疫
3. 关闭session窗口发送SIGHUP信号，程序会关闭

**所以可以使用 nohup ./aa.sh & 来同时免疫ctrl+c和关闭session**

## Top

### OS理论知识

虚拟内存、驻留内存、共享内存



#### 虚拟内存VIRT

#### 驻留内存RES

#### 共享内存SHR

#### 独占内存

要计算一个进程独占的内存大小，只需要驻留内存-共享内存即可（RES-SHR）

### 操作技巧

#### 排序

shift+>或者shift+<，分别按照某个列排序

ctrl+m按照内存%MEM排序，ctrl+p按照%CPU排序

#### 添加列

按f，光标移动到指定列上，按d，出现\*号，按q返回，这是就会出现该列信息

## -service和-systemctl

### service docker start / systemctl start docker

启动docker服务

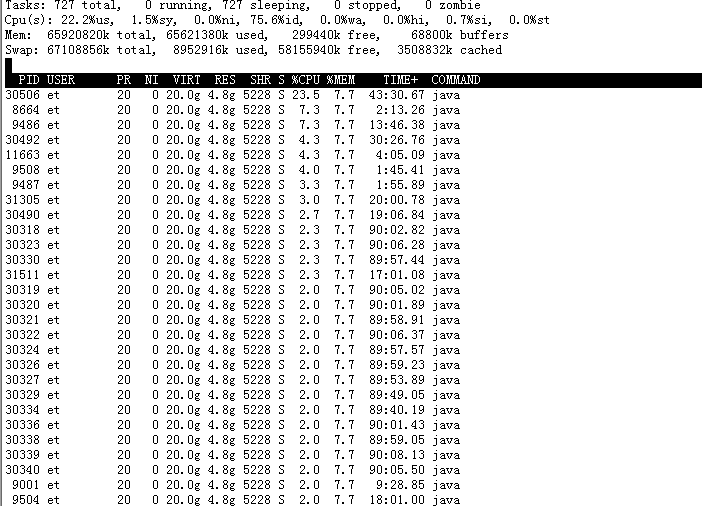
### systemctl enable docker

开机自启动

### top+jstack 分析线程

1. top -Hp pid

查询某个进程下所有线程的运行情况



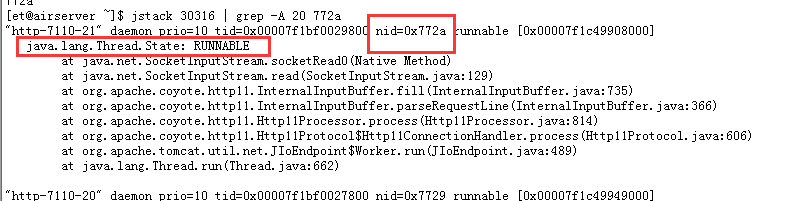
shift+p：按CPU排序

shift+m：按内存排序

printf ‘%x\n’ PID：找到PID，打印出十六进制。

jstack PID >> dump.txt：把瞬时堆栈信息打印到dump.txt文件中

在dump.txt文件中查找指定的十六进制PID即可



## ssh远程登录

目标：

服务器B要使用ssh登录服务器A

ssh

1. A和B的root目录下命令生成.ssh目录、秘钥和公钥 ssh-keygen -t rsa
2. A的.ssh下创建文件authorized\_keys（权限644），将B的公钥放到里面
3. A的/etc/ssh/sshd\_config下面增加 “root@B服务器IP”
4. A的/etc/ssh/sshd\_config设置PermitRootLogin yes 表示可以使用root用户登录
5. A重启ssh server sshd restart

如果A也要使用ssh登录服务器B，则进行相反操作，即将A的公钥放到B的/root/.ssh/authorized\_keys中，root@ip放到/etc/sshd/sshd\_config中

原理：生成私钥和公钥，将公钥放到想要ssh登录的服务器中，在使用ssh登录远程服务器时，发送私钥，这样如果远程服务器中有当前服务器的公钥，则私钥和公钥就能够匹配，就能够实现ssh登录。