目录

[Linux 基本命令 3](#_Toc477642280)

[目录、文件、路径管理 3](#_Toc477642281)

[ls 3](#_Toc477642282)

[1.只查看隐藏文件和目录 3](#_Toc477642283)

[2.只显示目录 4](#_Toc477642284)

[3.显示人类可读的文件大小 5](#_Toc477642285)

[chmod 5](#_Toc477642286)

[cat 5](#_Toc477642287)

[pwd 5](#_Toc477642288)

[ln 6](#_Toc477642289)

[1.软链接 6](#_Toc477642290)

[2.硬链接 6](#_Toc477642291)

[tree 6](#_Toc477642292)

[find 7](#_Toc477642293)

[1.文件类型 7](#_Toc477642294)

[2.-exec 用法 8](#_Toc477642295)

[3 .-mtime查找多少天以前的文件 8](#_Toc477642296)

[4.-maxdepth 设置查找最深层级 8](#_Toc477642297)

[5.-size 以文件大小为条件 8](#_Toc477642298)

[6. find 与 xargs 联合使用 9](#_Toc477642299)

[cp 9](#_Toc477642300)

[mkdir 9](#_Toc477642301)

[rm 10](#_Toc477642302)

[rmdir 10](#_Toc477642303)

[touch 10](#_Toc477642304)

[mv 10](#_Toc477642305)

[echo 10](#_Toc477642306)

[rename 11](#_Toc477642307)

[压缩与解压缩 11](#_Toc477642308)

[tar 11](#_Toc477642309)

[zip 11](#_Toc477642310)

[unzip 11](#_Toc477642311)

[gzip 11](#_Toc477642312)

[用户权限管理命令 11](#_Toc477642313)

[useradd 11](#_Toc477642314)

[userdel 12](#_Toc477642315)

[passwd 12](#_Toc477642316)

[groupadd 12](#_Toc477642317)

[groupdel 12](#_Toc477642318)

[rpm 12](#_Toc477642319)

[chown 12](#_Toc477642320)

[hash 12](#_Toc477642321)

[网络查看、管理命令 12](#_Toc477642322)

[netstat 13](#_Toc477642323)

[1.最常用的参数 13](#_Toc477642324)

[2.查看sshd侦听进程是否启动 13](#_Toc477642325)

[3查看所有TCP连接 13](#_Toc477642326)

[4.禁用反射域名解析 14](#_Toc477642327)

[5.只列表侦听中的连接 14](#_Toc477642328)

[6.显示统计数据 14](#_Toc477642329)

[7.显示内核路由信息 15](#_Toc477642330)

[8.打印网络接口信息 15](#_Toc477642331)

[traceroute 16](#_Toc477642332)

[ping 16](#_Toc477642333)

[ifconfig 16](#_Toc477642334)

[route 16](#_Toc477642335)

[1.显示内核路由信息 16](#_Toc477642336)

[telnet 16](#_Toc477642337)

[查看telnet 客户端是否安装(默认是安装的) 16](#_Toc477642338)

[查看telnet 服务器端是否安装 16](#_Toc477642339)

[开启telnet-server 17](#_Toc477642340)

[登录服务器 17](#_Toc477642341)

[让root 登录telnet 方法 17](#_Toc477642342)

[wget 18](#_Toc477642343)

[tcpdump 18](#_Toc477642344)

[指定端口、协议 18](#_Toc477642345)

[dig 18](#_Toc477642346)

[iptables 19](#_Toc477642347)

[curl 19](#_Toc477642348)

[环境变量 19](#_Toc477642349)

[alias 19](#_Toc477642350)

[export 20](#_Toc477642351)

[系统信息命令 20](#_Toc477642352)

[uptime 20](#_Toc477642353)

[top 20](#_Toc477642354)

[top 交互命令 21](#_Toc477642355)

[sar 21](#_Toc477642356)

[查看cpu实时负载 22](#_Toc477642357)

[查看IO历史负载 22](#_Toc477642358)

[查看系统瓶颈 23](#_Toc477642359)

[free 23](#_Toc477642360)

[iostat 23](#_Toc477642361)

[1.查看CPU 报表3次，每次间隔1秒 24](#_Toc477642362)

[vmstat 24](#_Toc477642363)

[df 24](#_Toc477642364)

[以人类可读方式显示磁盘空间大小占用 24](#_Toc477642365)

[指定文件类型磁盘 25](#_Toc477642366)

[ps 25](#_Toc477642367)

[查看所有用户进程 25](#_Toc477642368)

[查看前10个最占用内存的进程 25](#_Toc477642369)

[进程管理 26](#_Toc477642370)

[kill 27](#_Toc477642371)

[列出所有信号 kill -l 27](#_Toc477642372)

[通过信号名称获取信号数值 27](#_Toc477642373)

[killall 27](#_Toc477642374)

[常用工具 28](#_Toc477642375)

[计算相关 28](#_Toc477642376)

[bc 28](#_Toc477642377)

[字符串处理 28](#_Toc477642378)

[cut 28](#_Toc477642379)

[sort 28](#_Toc477642380)

[系统管理 29](#_Toc477642381)

[yum 29](#_Toc477642382)

[其他 29](#_Toc477642383)

[seq 29](#_Toc477642384)

**整理：梁毅强**

**邮箱:398504533@qq.com**

# Linux 基本命令

## 目录、文件、路径管理

### ls

linux 以.（点）开头的文件视为隐藏文件

#### 1.只查看隐藏文件和目录

ls –AF |grep ‘^\.’

. 号要用\转义，^表示以xx开头

A ---不显示目标.和.. 参数a会显示.和..,

F表示会在目录后加/斜杠

如果文件的末尾什么都没有表示这是一个普通的文件；

如果文件以"/"结尾，则表示这是一个目录；

如果文件以"@"结尾，则表示这是一个链接文件；

如果文件以“\*”结尾，则表示该文件具有可执行的属性；

[root@mytest ~]# ls -AF |grep '^\.'

.bash\_history

.bash\_logout

.bash\_profile

.bashrc

.cache/

.coffee\_history

.config/

.cshrc

.dbus/

.eclipse/

.gconf/

1. 显示/etc 目录下所有以大写p或小写p开头的文件或目录

ls -d /etc/[pP]\*

1. 显示/var 目录下的所有文件或目录本身，并将显示结果中的小写字母转换为大写后显示

ls -d /var/\*

#### 2.只显示目录

ls -d \*/

[root@mytest ~]# ls -d \*/

a/ Desktop/ dj14/ logs/ news/ scripts/ usr/

datedata/ dictdata/ djang14/ media/ project/ src/ work/

dd\_sender\_log/ disp\_log/ hawkeye/ mysqldata/ runlog/ tmp/ workspace/

ls -F |grep "/$"

[root@mytest ~]# ls -F |grep "/$"

a/

datedata/

dd\_sender\_log/

Desktop/

dictdata/

disp\_log/

dj14/

djang14/

hawkeye/

logs/

media/

mysqldata/

news/

project/

runlog/

scripts/

#### 3.显示人类可读的文件大小

[root@mytest tmp]# ls -hml

total 440K

drwxr-xr-x 2 root root 4.0K Feb 27 17:06 aaa

drwxr-xr-x 3 root root 4.0K May 16 2016 django-uwsgi-nginx-master

-rw-r--r-- 2 root root 15 Feb 27 17:19 exe2

-rw-r--r-- 2 root root 15 Feb 27 17:19 exe3

drwx------ 2 leon leon 4.0K Jul 21 2014 keyring-FVqEIU

drwx------ 2 leon leon 4.0K Sep 30 2014 keyring-HkuD9m

drwx------ 2 leon leon 4.0K Aug 15 2014 keyring-RrUIrz

drwx------ 2 leon leon 4.0K Oct 24 2014 keyring-X9njjp

-rw------- 1 root root 374K Feb 27 18:21 nohup.out

#### 4.显示长格式

ls -l /bin/awk

[root@localhost scripts]# ls -l /bin/awk

lrwxrwxrwx. 1 root root 4 Oct 26 2014 /bin/awk -> gawk

### chmod

更改文件的权限

chmod +x filename #为文件添加执行权限

### cat

显示行号

cat -n filename

显示结束符

cat -E exe2

[root@mytest tmp]# cat -nE exe2

1. test hard link$

在命令行下创建多行文件

#]cat >> newfile <<EOF

#!/bin/sh

echo “`date +%s%N`”

EOF

脚本功能：打印当前的纳秒数

cat 配合EOF完成

### pwd

pwd -L 显示逻辑链接路径

pwd -P 显示物理链接路径

[root@mytest init.d]# cd /etc/init.d/

[root@mytest init.d]# pwd -L

/etc/init.d

[root@mytest init.d]# pwd -P

/etc/rc.d/init.d

### ln

#### 1.软链接

mkdir aaa

ln -s aaa bbb

创建目录，并创建bbb软链接到aaa

drwxr-xr-x 2 root root 4096 Feb 27 17:06 aaa

lrwxrwxrwx 1 root root 3 Feb 27 17:08 bbb -> aaa

通过ls可以看出，bbb并不是一个目录而是一个l开头的链接文件

删除链接,就是直接删除bbb即可

rm –rf bbb #注意bbb后面不要带/

#### 2.硬链接

1)硬链接不能用于目录

2)删除文件并不会影响相互影响，但修改则会相互影响

[root@mytest tmp]# echo "test hard link" >> exe1

[root@mytest tmp]# ln exe1 exe2

[root@mytest tmp]# ll

-rw-r--r-- 2 root root 15 Feb 27 17:19 exe1

-rw-r--r-- 2 root root 15 Feb 27 17:19 exe2

[root@mytest tmp]# rm -rf exe1

[root@mytest tmp]# ls -al

-rw-r--r-- 1 root root 15 Feb 27 17:19 exe2

### tree

查看目录结构

tree dirname -L 3 –a

-a :包含隐藏文件

-L 定义显示层级，默认是所有都显示

mkdir -p ./vv/b/a/dd/d/dd/a/g3/d

[root@mytest tmp]# tree vv/

vv/

└── b

└── a

└── dd

└── d

└── dd

└── a

└── g3

└── d

8 directories, 0 files

[root@mytest tmp]# tree -L 3 vv/

vv/

└── b

└── a

└── dd

3 directories, 0 files

### find

#### 1.文件类型

-type 参数可接受的

b - 块设备文件。

d - 目录。

c - 字符设备文件。

p - 管道文件。

l - 符号链接文件。

f - 普通文件。

查找当前目录下以exe开头的链接类型的文件

[root@mytest tmp]# find ./ -type l -name "exe\*"

./exe4

一般最常用是-type f 用来查找普通文件

查出来后排序

[root@mytest tmp]# find ./ -type f -name "exe\*"|sort

./exe2

./exe3

#### 2.-exec 用法

-exec 查找出来后进行后续操作

格式：**-exec 执行命令 {}\;**

**重要：{} 代表前面用find查找出来的文件名**

[root@mytest tmp]# find ./ -type l -name "exe\*"

./exe4

./exe5

[root@mytest tmp]# find ./ -type l -name "exe\*" -exec ls -al {} \;

lrwxrwxrwx 1 root root 4 Feb 27 17:42 ./exe4 -> exe2

lrwxrwxrwx 1 root root 4 Feb 27 17:47 ./exe5 -> exe2

找到后复制

find ./ -type f -name "\*.log" -exec cp {} aa.log \;

#### 3 .-mtime查找多少天以前的文件

-mtime n n\*24

查找5天前的文件

[root@mytest runlog]# find ./ -type f -name "\*.log" -mtime +5

#### 4.-maxdepth 设置查找最深层级

[root@mytest tmp]# find ./ -maxdepth 1 -type f

#### 5.-size 以文件大小为条件

支持的参数

‘b’ for 512-byte blocks (this is the default if no suffix is used)

‘c’ for bytes

‘w’ for two-byte words

‘k’ for Kilobytes (units of 1024 bytes)

‘M’ for Megabytes (units of 1048576 bytes)

‘G’ for Gigabytes (units of 1073741824 bytes)

查找大于100k的文件

[root@mytest tmp]# find ./ -maxdepth 1 -type f -size +100k

./nohup.out

以人类可读的方式显示 ls -hml

查找小于10k的文件

[root@mytest tmp]# find ./ -maxdepth 1 -type f -size -10k -exec ls {} -hml \;

-r-------- 1 root root 1.7K Oct 27 15:37 ./test.pem

-rw-r--r-- 2 root root 15 Feb 27 17:19 ./exe3

-rw-r--r-- 2 root root 15 Feb 27 17:19 ./exe2

-rw-r--r-- 1 root root 3 Feb 27 18:28 ./vsss.txt

查找大于10k的文件

[root@mytest tmp]# find ./ -maxdepth 1 -type f -size +10k -exec ls {} -hml \;

-rw------- 1 root root 374K Feb 27 18:21 ./nohup.out

#### 6. find 与 xargs 联合使用

find 找到文件名后，将文件里的内容输出到new.log

1. xargs 通过管理接收参数，意味着换行和空白会被空格取代
2. 默认用echo 处理

[root@mytest runlog]# find ./ -type f -name "disp\_log.log" -exec cat {} \;|xargs > new.log

### cp

文件、目录复制命令，常用作备份

常用参数：

-R -r --recursive 递归复制，复制目录时必用，会将子目录一起复制

-f 如果目标已经存在，强制替换

### mkdir

创建目录

常用参数：

-p 递归创建多个目录

[root@mytest tmp]# mkdir -p bb/cc/dd/ff

[root@mytest tmp]# tree bb/

bb/

└── cc

└── dd

└── ff

3 directories, 0 files

-m 创建时自定义权限

默认权限是 drwxr-xr-x

[root@mytest tmp]# mkdir -m 777 cc

[root@mytest tmp]# ls -la cc

total 24

drwxrwxrwx 2 root root 4096 Mar 14 17:33 .

drwxrwxrwt. 17 root root 20480 Mar 14 17:33 ..

### rm

删除目录或文件

参数：

-r ,-R, --recursive 递归删除，删除目录及其子内容

-f, --force 不提示

-i 每次删除都提示

### rmdir

删除空目录

### touch

1.创建中间有空白符的文件

touch a\ b

2.创建以linux为前缀的1~10文件

学习大括号{} 的用法

touch linux-{1..10}

{n..m} 两点号分隔表示范围，表示创建从n到m的

{n,m,x,y,z} 逗号分隔表示只创建括号里的

### mv

移动文件或目录，亦具有重命名功能

mv dir1 dir2 将dir1重命名为dir2

mv file1 ./dir2 将file1移动到dir2目录下

### echo

显示一行文本

复杂例子：

### printf

与c语言的printf类似

1. 自定义宽度

printf “%-10s” teststring

1. 自定义leading

printf “%02d” number 少于10的整数都会带上0, 如果是03d,就是少于100的都带上0

### rename

修改文件名

语法：rename from to file...

如：将mm-1改为nn-1

rename mm-1 nn-1 ./mm-1

## 压缩与解压缩

### tar

归档工具

压缩

tar –xzcf archive\_file.tar.gz dir\_name

-c 压缩参数

解压缩

tar -xzvf archive\_file.tar.gz

-v 解压参数

### zip

### unzip

### gzip

## 用户权限管理命令

### useradd

创建新用户或更新用户信息

参数：

-d --home 为新用户指定家目录,如不指定，默认是在/home/username下

### userdel

删除用户

参数 ：

-r --remove 将用户的家目录也一并删除，如果不加这个参数，默认是不会删除家目录的

### passwd

修改用户密码

常用参数：

--stdin 从标准输入里读取新密码，一般从pipe中

echo “newpass” | passwd username --stdin

### groupadd

### groupdel

### rpm

安装rpm包，redhat类发行版特有

安装rpm包：rpm –ivh file.rpm

### chown

### hash

hash: 列出

hash –d :command :删除

hash –r :清空

## 网络查看、管理命令

### netstat

#### 1.最常用的参数

-l, --listening 仅显示侦听socket

-n, --numeric 显示ip地址而不是主机名

-p, --program 显示PID和进程名称

-t, --tcp

-u, --udp

#### 2.查看sshd侦听进程是否启动

[root@mytest tmp]# netstat -lntup|grep sshd

tcp 0 0 0.0.0.0:22 0.0.0.0:\* LISTEN 2603/sshd

tcp 0 0 :::22 :::\* LISTEN 2603/sshd

如果不加p则只能查询sshd对应的端口来获取数据

sshd 配置文件地址/etc/ssh/sshd\_config 配置的默认端口为 Port 22

#### 3查看所有TCP连接

-a 表示包括侦听和未侦听的

ESTABLISHED 表示已经建立的连接

[root@mytest tmp]# netstat -at

Active Internet connections (servers and established)

Proto Recv-Q Send-Q Local Address Foreign Address State

tcp 0 0 mytest.com:8008 \*:\* LISTEN

tcp 0 0 \*:mysql \*:\* LISTEN

tcp 0 0 localhost:6379 \*:\* LISTEN

tcp 0 0 \*:ssh \*:\* LISTEN

tcp 0 0 localhost:ipp \*:\* LISTEN

tcp 0 0 localhost:smtp \*:\* LISTEN

tcp 0 0 mytest.com:ssh 192.168.72.1:49636 ESTABLISHED

tcp 0 0 mytest.com:ssh 192.168.72.1:49565 ESTABLISHED

tcp 0 0 mytest.com:ssh 192.168.72.1:49559 ESTABLISHED

tcp 0 0 mytest.com:ssh 192.168.72.1:49629 ESTABLISHED

tcp 0 0 mytest.com:ssh 192.168.72.1:49528 ESTABLISHED

#### 4.禁用反射域名解析

默认情况下 netstat 会通过反向域名解析技术查找每个 IP 地址对应的主机名。这会降低查找速度。如果你觉得 IP 地址已经足够，而没有必要知道主机名，就使用 -n 选项禁用域名解析功能。

[root@mytest tmp]# netstat -ant

Active Internet connections (servers and established)

Proto Recv-Q Send-Q Local Address Foreign Address State

tcp 0 0 192.168.72.130:8008 0.0.0.0:\* LISTEN

tcp 0 0 0.0.0.0:3306 0.0.0.0:\* LISTEN

tcp 0 0 127.0.0.1:6379 0.0.0.0:\* LISTEN

tcp 0 0 0.0.0.0:22 0.0.0.0:\* LISTEN

tcp 0 0 127.0.0.1:631 0.0.0.0:\* LISTEN

#### 5.只列表侦听中的连接

[root@mytest tmp]# netstat -tnl

Active Internet connections (only servers)

Proto Recv-Q Send-Q Local Address Foreign Address State

tcp 0 0 192.168.72.130:8008 0.0.0.0:\* LISTEN

tcp 0 0 0.0.0.0:3306 0.0.0.0:\* LISTEN

tcp 0 0 127.0.0.1:6379 0.0.0.0:\* LISTEN

tcp 0 0 0.0.0.0:22 0.0.0.0:\* LISTEN

tcp 0 0 127.0.0.1:631 0.0.0.0:\* LISTEN

tcp 0 0 127.0.0.1:25 0.0.0.0:\* LISTEN

tcp 0 0 :::873 :::\* LISTEN

tcp 0 0 :::22 :::\* LISTEN

tcp 0 0 ::1:631 :::\* LISTEN

tcp 0 0 ::1:25 :::\* LISTEN

#### 6.显示统计数据

[root@mytest tmp]# netstat -s

Ip:

370390 total packets received

0 forwarded

0 incoming packets discarded

370390 incoming packets delivered

262725 requests sent out

Icmp:

0 ICMP messages received

0 input ICMP message failed.

ICMP input histogram:

3 ICMP messages sent

0 ICMP messages failed

ICMP output histogram:

destination unreachable: 3

IcmpMsg:

OutType3: 3

Tcp:

1387 active connections openings

4229 passive connection openings

#### 7.显示内核路由信息

相当于route 命令

[root@mytest tmp]# netstat -rn

Kernel IP routing table

Destination Gateway Genmask Flags MSS Window irtt Iface

192.168.72.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0

169.254.0.0 0.0.0.0 255.255.0.0 U 0 0 0 eth0

0.0.0.0 192.168.72.2 0.0.0.0 UG 0 0 0 eth0

#### 8.打印网络接口信息

[root@mytest tmp]# netstat -i

Kernel Interface table

Iface MTU Met RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flg

eth0 1500 0 317839 0 0 0 238477 0 0 0 BMRU

lo 65536 0 56260 0 0 0 56260 0 0 0 LRU

### traceroute

### ping

用了TCP/IP ICMP协议实现

参数：

-c ping的次数

### ifconfig

### route

#### 1.显示内核路由信息

[root@mytest tmp]# route

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

192.168.72.0 \* 255.255.255.0 U 0 0 0 eth0

link-local \* 255.255.0.0 U 1002 0 0 eth0

default 192.168.72.2 0.0.0.0 UG 0 0 0 eth0

### telnet

telnet 协议的用户接口，telnet协议属于TCP/IP协议簇的应用层

#### 查看telnet 客户端是否安装(默认是安装的)

rpm –q telnet

#### 查看telnet 服务器端是否安装

rpm –q telnet-server

用yum 安装telnet-server

yum install telnet-server

#### 开启telnet-server

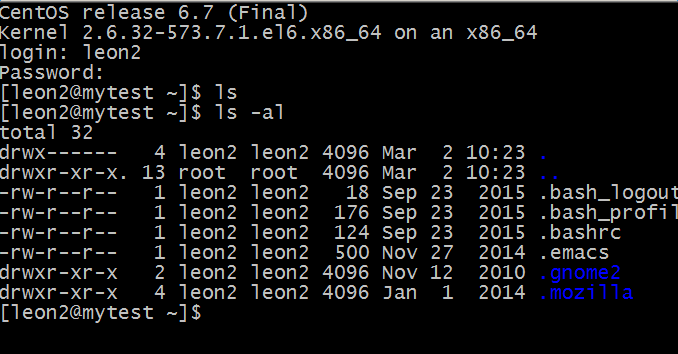
编辑 /etc/xinted.d/telnet 将disable = yes 改为 no

chkconfig telnet on

service xinetd restart

#### 登录服务器

telnet host



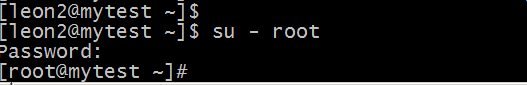
注：出于安全原因，默认root用户是无法通过telnet 登录的

#### 让root 登录telnet 方法

1.注释 /etc/pam.d/remote 这行: （未验证）

auth required pam\_securetty.so

2.更简单的方法是先登录一个普通用户，再su – root 通过密码登录



### wget

### tcpdump

#### 指定端口、协议

tcpdump udp port 53865 只监测udp协议及端口为53865的数据包

收到类似这样显示的包

17:56:35.791712 IP lily.com.53865 > mytest.com.8096: UDP, length 939

意思是

数据从为lily.com 端口为53865流向st.com 端口为8096

### dig

dig (domain information groper)

QUESTION SECTION：查询的内容

ANSWER SECTION：相应的内容，一般会得到至少一条A记录，否则就还没定义

AUTHORITY SECTION：授权信息(权威DNS)

ADDITIONAL SECTION：每个授权服务器的IP地址

SERVER：查询的dns服务器，可能会被缓存

域名解析过程：

dig [www.baidu.com](http://www.baidu.com)

解析过程是这样的：

    · 1.DNS客户端发包到DNS服务端请求www.baidu.com的IP地址，由于有一条CNAME记录；

    · 2.DNS客户端会访问www.a.shifen.com的域名服务器（相当于是访问www.baidu.com的域名服务器）；

    · 3.在这里www.a.shifen.com的域名服务器是一个负责DNS查询流量均衡的调度器，负责把DNS请求调度到ns5.a.shifen.com、ns6.a.shifen.com等域名服务器上

    · 4.最后这些标记了NS的域名服务器会把百度的IP返回给DNS客户端

    · 然后我们就得到了www.baidu.com的IP地址

dig [www.baidu.com](http://www.baidu.com) +trace 查看整个解析过程，从根域开始

dig [www.baidu.com](http://www.baidu.com) +noall +authority 只查看权威DNS

[root@mytest ~]# dig www.baidu.com +noall +authority

; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.37.rc1.el6\_7.4 <<>> www.baidu.com +noall +authority

;; global options: +cmd

a.shifen.com. 5 IN NS ns3.a.shifen.com.

a.shifen.com. 5 IN NS ns4.a.shifen.com.

a.shifen.com. 5 IN NS ns5.a.shifen.com.

a.shifen.com. 5 IN NS ns1.a.shifen.com.

a.shifen.com. 5 IN NS ns2.a.shifen.com.

note:

+noall 要放在前面

### iptables

配置文件位置：/etc/sysconfig/iptables

### curl

常用参数

-o 将文件下载下来，保存到指定的文件

curl [http://www.baidu.com/login.png -o /tmp/bg.png](http://www.baidu.com/login.png%20-o%20/tmp/bg.png)

## 环境变量

### alias

特点：临时设置，重启后失效

语法：alias 别名=’原命令 参数’

如：

alias vi=’vim’

alias grep=’grep –color=auto’ 为grep搜索到的结果添加颜色

因为shell里设置后重启后就会失效，所以如果要永久设置，还必须写进用户的环境变量配置文件(~/.bashrc)中或者管理员写进全局环境变量配置文件(/etc/bashrc)中。

### export

## 系统信息命令

### date

系统日期时间设置

语法：date [OPTION]... [+FORMAT]

注：格式前要用 +号，因为要与参数区分

最常用的格式：

%Y 年

%m 月 (e.g, 01~12)

%d 日 (e.g, 01)

%H 时

%M 分

%S 秒

%s 小写s,时间戳

%b 月份名称缩写 (e.g. Mar Jun)

--rfc-3339=”TIMESPEC” 以 rfc-3339格式输出， TIMESPEC参数有三个，分别是 date,seconds,ns

#### 显示当前时间戳

从1970-01-01 00:00:00 UTC 到现在

]# date '+%s'

1489716988

+%s 秒

+%N 纳秒 (influxdb 里要求用纳秒记录时间)

#### 获取rfc3339格式时间

$ date --rfc-3339=date

2014-03-19

$ date --rfc-3339=seconds

2014-03-19 18:00:05-04:00

$ date --rfc-3339=ns

2014-03-19 18:00:08.179780629-04:00

### cal

显示日历

语法：cal [-smjy13] [[[day] month] year]

示例：

1. 显示当前的日历

直接 cal

1. 显示指定的日历

cal 2 2017

1. 显示某月份最后一天

[root@mytest logs]# echo $(cal 2 2017)|awk '{print $(NF)}'

28

### uptime

用一行信息来显示系统平均负载，分别是1分钟、5分钟、15分钟

[root@mytest ~]# uptime

17:05:59 up 3:24, 3 users, load average: 0.03, 0.02, 0.00

单核处理器下：

load 持续>= 0.7就要引起重视了，查找问题原因，防止恶化

load 持续>=1.0 必须手动解决问题，将值降下来

load 达到5.0 时，严重问题，接近死机

多核处理器下：

只要每个核心的load不超过1.0，表明正常

粗略的判定方法：

如，主机双核心CPU,那么最高load 应该只能是到2,如此类推

### top

实时显示系统资源及各进程资源占用情况

默认以%CPU排序，可用通过shift + < 或 + >改变排序key

top –c 显示进程详细命令

top –p 进程ID 显示指定进程ID

top - 09:55:42 up 3 days, 23:25, 6 users, load average: 0.00, 0.00, 0.00

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

Cpu(s): 0.0%us, 0.0%sy, 0.0%ni,100.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st

Mem: 4213836k total, 2243160k used, 1970676k free, 342812k buffers

Swap: 1015804k total, 0k used, 1015804k free, 926852k cached

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

3116 mysql 20 0 1394m 109m 5164 S 0.0 2.7 2:11.40 mysqld

#### top 交互命令

在top 命令执行过程中可以使用的一些交互命令。这些命令都是单字母的，如果在命令行中使用了s 选项， 其中一些命令可能会被屏蔽。

h 显示帮助画面，给出一些简短的命令总结说明

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

r 重新安排一个进程的优先级别

S 切换到累计模式

s 改变两次刷新之间的延迟时间（单位为s），如果有小数，就换算成m s。输入0值则系统将不断刷新，默认值是5 s

f或者F 从当前显示中添加或者删除项目

o或者O 改变显示项目的顺序

l 切换显示平均负载和启动时间信息

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

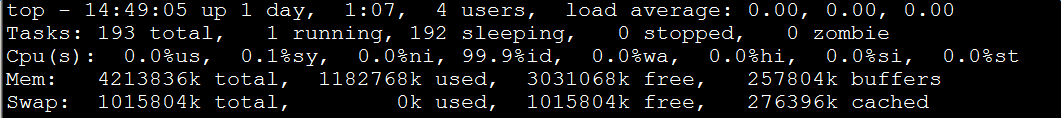
M 根据驻留内存大小进行排序

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

W 将当前设置写入~/.toprc文件中

示例：



Top命令注意看输出的第二、第三行。第二行（Tasks）表示系统运行的进程数，这个数值一般不能太大，比如超过300、甚至400-500，这个比较容易理解，我们一般开启的服务、进程外加系统进程都不会太多，太多肯定有问题。第三行表示CPU的使用情况。  
**Top可以按 M按内存使用排序，P按CPU使用情况排序。**

### sar

收集、报告、保存操作系统当前活动信息

-A 输入所有

-b 输入I/O 信息，默认是cpu

#### 查看cpu实时负载

sar 1

[root@mytest tmp]# sar 1

Linux 2.6.32-573.7.1.el6.x86\_64 (mytest.com) 03/01/2017 \_x86\_64\_ (4 CPU)

09:50:05 AM CPU %user %nice %system %iowait %steal %idle

09:50:06 AM all 0.00 0.00 0.25 0.00 0.00 99.75

09:50:07 AM all 0.00 0.00 0.25 0.00 0.00 99.75

09:50:08 AM all 0.00 0.00 0.50 0.00 0.00 99.50

09:50:09 AM all 0.00 0.00 0.25 0.00 0.00 99.75

09:50:10 AM all 0.00 0.00 0.25 0.00 0.00 99.75

09:50:11 AM all 0.00 0.00 0.50 0.00 0.00 99.50

* %user 用户模式下消耗的CPU时间的比例；
* %nice 通过nice改变了进程调度优先级的进程，在用户模式下消耗的CPU时间的比例
* %system 系统模式下消耗的CPU时间的比例；
* %iowait CPU等待磁盘I/O导致空闲状态消耗的时间比例；
* %steal 利用Xen等操作系统虚拟化技术，等待其它虚拟CPU计算占用的时间比例；
* %idle CPU空闲时间比例；

参考：<http://linuxtools-rst.readthedocs.io/zh_CN/latest/tool/sar.html>

#### 查看IO历史负载

sar –b

##### IO各项指标

tps：每秒钟物理设备的 I/O传输总量

rtps：每秒钟从物理设备读入的数据总量

wtps：每秒钟向物理设备写入的数据总量

bread/s：每秒钟从物理设备读入的数据量，单位为块/s

bwrtn/s：每秒钟向物理设备写入的数据量，单位为块/s

#### 查看系统瓶颈

要判断系统瓶颈问题，有时需几个 sar 命令选项结合起来

怀疑CPU存在瓶颈，可用 sar -u和 sar -q 等来查看

怀疑内存存在瓶颈，可用 sar -B、sar -r和 sar -W 等来查看

怀疑I/O存在瓶颈，可用 sar -b、sar -u 和 sar -d等来查看

参考：<http://blog.csdn.net/mig_davidli/article/details/52149993>

### free

查看系统中总内存及消耗的内存

free –m 以mb为单位显示

[root@mytest cmds]# free -m

total used free shared buffers cached

Mem: 4115 1154 2960 0 251 269

-/+ buffers/cache: 633 3481

Swap: 991 0 991

**注意，重点看的是第二行，-/+ buffers/cache行，第一个数字是实际消耗的内存，第二个是实际空闲的内存。**

### iostat

用来查看CPU、磁盘、网络实时状态报表

-c CPU

-d 驱动器

-n NFS network filesystem

#### 1.查看CPU 报表3次，每次间隔1秒

[root@mytest disp]# iostat -c 1 3

Linux 2.6.32-573.7.1.el6.x86\_64 (mytest.com) 03/01/2017 \_x86\_64\_ (4 CPU)

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

0.07 0.00 0.14 0.34 0.00 99.44

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

0.00 0.00 0.25 0.00 0.00 99.75

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

0.00 0.00 0.00 0.00 0.00 100.00

检测io的建康状况

iostat –x

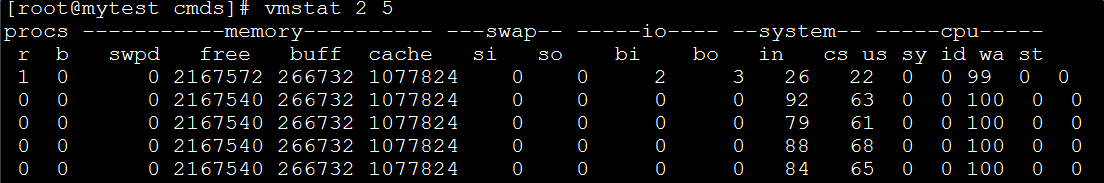
-x 参数的作用是显示过去请求磁盘IO的百分比，如果接近100%，说明请求太多

### vmstat

vmstat和IO差不过，不过重点是监控cpu、内存等系统资源。

对内存监控，我们比较关心swpd、free、si、so。一般系统不繁忙的状态下，我们看到swpd,so的值不会持续很高，经常为0。如果swpd过高，那么就是系统内存经常不够用。

对CPU监控，我们可以查看r（运行进程数）、us、sy、id（空闲CPU），如果r的数字大于系统CPU个数，则面临CPU不够用的危险，通过id（CPU空闲比例）分析，如果过小，则可以判断是CPU不足。



### df

检测系统文件空间占用情况

#### 以人类可读方式显示磁盘空间大小占用

[root@mytest ~]# df -h

Filesystem Size Used Avail Use% Mounted on

/dev/sda2 19G 16G 2.2G 88% /

tmpfs 2.1G 0 2.1G 0% /dev/shm

/dev/sda1 283M 65M 203M 25% /boot

#### 指定文件类型磁盘

[root@mytest ~]# df -t ext4

Filesystem 1K-blocks Used Available Use% Mounted on

/dev/sda2 19207884 15998404 2227108 88% /

/dev/sda1 289293 66504 207429 25% /boot

### ps

#### 查看所有用户进程

ps aux

[root@mytest tmp]# ps aux|head -10

USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND

root 1 0.0 0.0 19356 1540 ? Ss Feb27 0:06 /sbin/init

root 2 0.0 0.0 0 0 ? S Feb27 0:00 [kthreadd]

root 3 0.0 0.0 0 0 ? S Feb27 0:02 [migration/0]

root 4 0.0 0.0 0 0 ? S Feb27 0:02 [ksoftirqd/0]

root 5 0.0 0.0 0 0 ? S Feb27 0:00 [stopper/0]

root 6 0.0 0.0 0 0 ? S Feb27 0:00 [watchdog/0]

root 7 0.0 0.0 0 0 ? S Feb27 0:04 [migration/1]

root 8 0.0 0.0 0 0 ? S Feb27 0:00 [stopper/1]

root 9 0.0 0.0 0 0 ? S Feb27 0:02 [ksoftirqd/1]

head -10是只显示前十个

#### 查看前10个最占用内存的进程

ps auxw|head -1;ps aux|sort -rn -k4|head -10

[root@mytest tmp]# ps auxw|head -1;ps aux|sort -rn -k4|head -10

USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND

mysql 3116 0.0 2.6 1362516 111292 pts/0 Sl Feb27 1:35 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql --user=mysql --log-error=/var/log/mysqld.log --pid-file=/var/run/mysqld/mysqld.pid --socket=/var/lib/mysql/mysql.sock

redis 2249 0.1 2.4 237680 103512 ? Ssl Feb27 3:26 /usr/bin/redis-server 127.0.0.1:6379

root 22122 0.1 1.2 418188 52676 pts/1 Sl+ 10:01 0:01 /usr/local/bin/python2.7 /usr/bigip2/manage.py runserver 192.168.72.130:8008

root 22113 0.0 1.2 263460 50832 pts/1 S+ 10:01 0:01 /usr/local/bin/python2.7 -u /usr/bigip2/manage.py runserver 192.168.72.130:8008

root 2276 0.0 0.1 82912 4576 ? Ss Feb27 0:02 NetworkManager --pid-file=/var/run/NetworkManager/NetworkManager.pid

root 21978 0.0 0.1 100172 4588 ? Ss 09:59 0:00 sshd: root@pts/1

root 21846 0.0 0.1 100112 4480 ? Ss 09:59 0:00 sshd: root@notty

root 21714 0.0 0.1 100112 4480 ? Ss 09:59 0:00 sshd: root@notty

root 21582 0.0 0.1 100112 4476 ? Ss 09:59 0:00 sshd: root@notty

root 21450 0.0 0.1 100112 4480 ? Ss 09:59 0:00 sshd: root@notty

ps auxw|head -1 显示列的名称

USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND

几个常用参数：

-e 选择所有进程，作用与-A相同

-f full-format

UID PID PPID C STIME TTY TIME CMD

-o 格式

m min\_flt number of minor page faults

M maj\_flt number of major page faults

n cmin\_flt cumulative minor page faults

N cmaj\_flt cumulative major page faults

o session session ID

p pid process ID

P ppid parent process ID

r rss resident set size

R resident resident pages

s size memory size in kilobytes

S share amount of shared pages

t tty the device number of the controlling tty

T start\_time time process was started

U uid user ID number

u user user name

v vsize total VM size in kB

y priority kernel scheduling priority

## 进程管理

### kill

终止前台进程可以用ctrl+c ,但要终止后台进程就须用kill,要终止进程必须先通过ps/top/ptree等获取到进程ID，再通过kill向进程ID发送指定的信号来结束进程。

#### 列出所有信号 kill -l

[root@mytest mywiki]# kill -l

1) SIGHUP 2) SIGINT 3) SIGQUIT 4) SIGILL 5) SIGTRAP

6) SIGABRT 7) SIGBUS 8) SIGFPE 9) SIGKILL 10) SIGUSR1

11) SIGSEGV 12) SIGUSR2 13) SIGPIPE 14) SIGALRM 15) SIGTERM

16) SIGSTKFLT 17) SIGCHLD 18) SIGCONT 19) SIGSTOP 20) SIGTSTP

21) SIGTTIN 22) SIGTTOU 23) SIGURG 24) SIGXCPU 25) SIGXFSZ

26) SIGVTALRM 27) SIGPROF 28) SIGWINCH 29) SIGIO 30) SIGPWR

31) SIGSYS 34) SIGRTMIN 35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3

38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8

43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13

48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12

53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9 56) SIGRTMAX-8 57) SIGRTMAX-7

58) SIGRTMAX-6 59) SIGRTMAX-5 60) SIGRTMAX-4 61) SIGRTMAX-3 62) SIGRTMAX-2

63) SIGRTMAX-1 64) SIGRTMAX

第9号SIGKILL 为无条件终止信号

#### 通过信号名称获取信号数值

[root@mytest mywiki]# kill -l SIGKILL

9

### killall

根据名称杀死进程

killall -9 vi

### exec

用于调用并执行指令，如果在终端使用此命令，则会在执行完后立即退出终端

|  |  |
| --- | --- |
| exec命令 | 作用 |
| exec ls | 在shell中执行ls，ls结束后不返回原来的shell中了 |
| exec <file | 将file中的内容作为exec的标准输入 |
| exec >file | 将file中的内容作为标准写出 |
| exec 3<file | 将file读入到fd3中 |
| sort <&3 | fd3中读入的内容被分类 |
| exec 4>file | 将写入fd4中的内容写入file中 |
| ls >&4 | Ls将不会有显示，直接写入fd4中了，即上面的file中 |
| exec 5<&4 | 创建fd4的拷贝fd5 |
| exec 3<&- | 关闭fd3 |

# 常用工具

## 计算相关

### bc

### md5sum

计算字符串随机数

用纳秒来做随机数

[root@localhost oldboy]# echo `date +%s%N`|md5sum

cdb9980ed6df1bad205503b984ee42d5 -

最后显示的 - 表示标准输入

## 字符串\数据流处理

### cut

剪切数据

三种定位方法：

1）bytes -b

[root@mytest cmds]# who|cut -b 1-4

root

root

root

root

root

2）characters -c

3）fields -f

域提取

]# cat /etc/passwd | cut -d":" -f 1

取词的范围：

N

N-

N-M

-M

### sort

对文本以行为单位排序，一般是结合其他命令使用

实用例子

1. 查找占用CPU、内存等最多的程序

ps -eo rss,pmem,pcpu,vsize,args | sort -k 1 -r -n | head -10

options:

-k 以哪一列为基准排序

-r 倒序

-n 数值排序

### tail

获取文件最后指定的行数

cat file.txt | tail -10 #获取file.txt 最后10行

### head

获取文本最前指定的行数

cat file.txt | head -10 #获取最前面10行

## 系统管理

### yum

更换国内源

第一步：备份你的原镜像文件，以免出错后可以恢复。

mv /etc/yum.repos.d/CentOS-Base.repo etc/yum.repos.d/CentOS-Base.repo.backup

第二步：下载新的CentOS-Base.repo 到/etc/yum.repos.d/

wget -O /etc/yum.repos.d/CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-6.repo

第三步：运行yum makecache生成缓存

yum makecache

## 其他

### seq

生成一个数字序列

SYNOPSIS

seq [OPTION]... LAST

seq [OPTION]... FIRST LAST

seq [OPTION]... FIRST INCREMENT LAST

例：

1.生成1~10的数字序列

~]$ seq 10

1

2

3

4

5

6

7

8

9

10

1. 生成带指定范围的序列

~]$ seq 3 4

3

4

1. 生成以逗号为分隔的数字序列

]$ seq -s “,” 1 4

1,2,3,4

如果加 –s 参数，序列会以-s 后的字符串作为分隔符

1. –w 参数在前面加0以等宽显示

]$ seq -s " " -w 10

01 02 03 04 05 06 07 08 09 10