# 第六篇 Linux

## 6.1 Linux系统资源分析

### 6.1.1 top监控某个进程

–p pid 监控某个进程

-H show thread, top –Hp pid

Tasks:进程数。-H切换成线程数。

4 users：当前登录user数

PR 优先级

VIRT 进程使用的虚拟内存总量，单位kb。VIRT=SWAP+RES

RES 进程使用的、未被换出的物理内存大小，单位kb。RES=CODE+DATA

SHR 共享内存大小，单位kb

%CPU  CPU时间占用百分比

%MEM  进程使用的物理内存百分比

TIME+  进程使用的CPU时间总计，单位1/100秒  
CMD 进程名称（命令名/命令行）

**快捷键**

P:CPU排序。默认按cpu占用量排序。

M:内存排序

H: show thread

1:show cups 列出单个cpu的负荷情况

**Load Average**

一段时间 (1 分钟、5分钟、15分钟) 内平均 Load

系统负荷正常范围：

* 大于0.7，必须开始调查了，问题出在哪里，防止情况恶化。
* 持续大于1.0，必须动手寻找解决办法，把这个值降下来。
* 当系统负荷达到5.0，就表明系统有很严重的问题，长时间没有响应，或者接近死机了。

以上指标基于单CPU的，四核机器的负载最好保持在3(4\*0.7 = 2.8)以下。

%CPU %MEM

**命令补充**

监控最小单位是进程，看不到java线程和客户连接数，可用ps和netstat两个命令来补充top的不足。

### 6.1.2 ps

查看线程。Process Status

a 显示现行终端机下的所有程序，包括其他用户的程序

f 用ASCII字符显示树状结构，表达程序间的相互关系

e 列出程序时，显示程序所使用的环境变量

-A 显示所有进程

-u 以用户为主的格式来显示程序状况

-x 显示所有程序，不以终端机来区分

常用

ef



USER 用户名

UID User ID

PID Process ID（进程），当然jps命令也可以查到进行ID

PPID parent pid

TIME 进程使用的总cpu时间

COMMAND 正在执行的命令行命令

aux



SID 会话ID（Session id）

%CPU 进程的cpu占用率

%MEM 进程的内存占用率

VSZ 进程所使用的虚存的大小（Virtual Size）

RSS 进程使用的驻留集大小或者是实际内存的大小，Kbytes字节。

TTY 与进程关联的终端（tty）

STAT 进程的状态：进程状态使用字符表示的（STAT的状态码）

START 进程启动时间和日期

### 6.1.3 netstat

可用于列出系统上所有的网络套接字连接情况，包括 tcp, udp 以及 unix 套接字。

Active Internet connections，称为有源TCP连接

Active UNIX domain sockets，称为有源Unix域套接口(和网络套接字一样，但是只能用于本机通信，性能可以提高一倍)

Param

-a (all)显示所有选项，默认不显示LISTEN相关

-t (tcp)仅显示tcp相关选项

-n 拒绝显示别名(域名)，能显示数字的全部转化成数字(ip)。

-u (udp)仅显示udp相关选项  
-p 显示建立相关链接的程序名

-l 仅列出有在 Listen (监听) 的服務状态

-x**只列出所有监听 UNIX 端口**

提示：LISTEN和LISTENING的状态只有用-a或者-l才能看到

-p 显示建立相关链接的程序名  
-r 显示路由信息，路由表  
-e 显示扩展信息，例如uid等  
-s 按各个协议进行统计  
-c 每隔一个固定时间，执行该netstat命令

**常用**

1)netstat -at|grep slc01boa

2)统计80端口连接数

netstat -nat|grep -i "8080"|wc –l

3)Linux下查看tomcat连接数

netstat -na | grep ESTABLISHED | grep 8080 | wc -l

1. 统计httpd协议连接数  
   ps -ef|grep httpd|wc –l
2. 统计已连接上的，状态为“established

netstat -na|grep ESTABLISHED|wc -l

3)找出程序运行的端口

netstat -ap | grep java

**4)**[根据端口查找进程](http://www.cnblogs.com/paul8339/p/6638370.html)

netstat -apn | grep 8989

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

windows

netstat -ano -p tcp -n 1|find "8080"

-n 1间隔1s自动刷新



PID是进程

统计

netstat -an|find "ESTABLISHED" /c

title

local address本机的地址

foreign address 外部电脑电脑有联系的ip地址

### free

[Reference](http://www.cnblogs.com/coldplayerest/archive/2010/02/20/1669949.html)



-/+ buffers/cache: 应用程序认为系统被用掉多少内存

设置缓存大小：

### df

Disk File

-i 查看inodes

df –hl

df -h /home/nadong

### du

Disk usage of each File

du –sh

分区

### jps

显示java进程的pid



## 6.2 文件

### 文件属性

文件颜色

绿色文件---------- 可执行文件，可执行的程序

红色文件-----------压缩文件或者包文件

蓝色文件----------目录

白色文件----------一般性文件，如文本文件，配置文件，[源码](https://www.2cto.com/ym/)文件等

浅蓝色文件----------链接文件，主要是使用ln命令建立的文件



链接文件

分配不同权限

缩写路径

软链接 VS 硬链接

软链接和Windows的快捷方式类似

Chmod

chmod u+w test

chmod ugo+r test;

### 文件搜索

find

find [dir] -name filename

1.在当前及子文件夹查找

find –name common.jar

2.在etc及子文件夹查找

find /etc -name httpd.conf

3.模糊查找

find –name ‘httpd\*'

Grep

-n ：输出行号

-r ：在某一目录下递归[所有子目录]查找某一字串:

grep -r ‘str\*' . (注意此处有点)

grep -r a /scratch

Whereis

whereis useradd

### 文件查看及编辑

more

Enter:一行

Space:翻页

less

less 的用法比起 more 更加的有弹性。在 more 的时候，我们并没有办法向前面翻， 只能往后面看

Space:翻页

G:倒序查看

vim

**全部删除：**按esc后，然后dG

lsof

### jar

-c 创建一个jar包  
-t 显示jar中的内容列表  
-x 解压jar包  
-u 添加文件到jar包中  
-f 指定jar包的文件名  
-v生成详细的报造，并输出至标准设备

1）解压

Jar -xvf test.ear

2）创建

Jar -cvf test.jar ./\*

3）更新

jar -uvf RESTWebService.ear APP-INF/lib/AdfHcmUsersRestModel.jar

4）查看

jar tvf test.jar

5）搜索：

jar tvf ZhcLib.jar | grep 'pattern'

搜索整个目录下的所有jar文件：

find . -name "\*.jar"|awk '{print "jar -tvf "$1}' | sh | grep ".\*.xml"

## 6.3 用户权限

### 用户组

**1)groups**:  查看当前登录用户的组内成员

**2)/etc/group**

group\_name:passwd:GID:user\_list

**3) groupadd**

group add group1

**4) groupdel**

删除群组。先删除group的用户，方能删除群组

**usermod**

usermod -a -G root nadong

将一个用户添加到用户组中c

**chown**

将指定文件的拥有者改为指定的用户或组

chown -R mysql: mysql ./

### 用户

**useradd**

adduser moon

新建用户::useradd -d /home/moon -m moon

修改用户的密码: passwd tes

**userdel**

userdel  moon

1. 将新用户添加到管理组gpasswd -a test admin

3、给 test 用户创建自己的目录：

cd /home

mkdir test

chown test /home/test

4、重新启动，

reboot 然后用 test 登录，

**su**

5. :Su switch use /sudo输入当前用户的密码。

**sudo**

1.需要root赋予sudo权限

su root🡪 hadoop ALL=(ALL) ALL

## 6.4 CMD

### Nginx

1)启动操作  
nginx -c /usr/local/nginx/conf/nginx.conf  
2)停止操作  
从容停止Nginx  
快速停止Nginx：  
kill -TERM 主进程号  
  
强制停止Nginx：  
kill -9 主进程号  
3)  
nginx -s reload

### mysql

#启动服务 service mysqld start  
#连接 ./mysql/bin/mysql -uroot  
DDL  
show databases/tables  
desc table\_name;

### redis

1)启动： redis-server.exe redis.windows.conf

开启memcache，并连接测试：  
以守护进程模式启动memcache  
memcached-1.5.2/bin/memcached -d -l slc11fsp.us.oracle.com -p 11211 -m 2048 -u root

## 6.5 SHELL

变量

$# 是传给脚本的参数个数

$0 是脚本本身的名字

$1 是传递给该shell脚本的第一个参数

$2 是传递给该shell脚本的第二个参数

$@ 是传给脚本的所有参数的列表

$\* 是以一个单字符串显示所有向脚本传递的参数，与位置变量不同，参数可超过9个

$$ 是脚本运行的当前进程ID号

$? 是显示最后命令的退出状态，0 = success, 其他 = error

${var}等价于$var（简写）

if/else

[Reference](http://www.cnblogs.com/kangyoung/p/3556173.html)

比较

if ([ condition ] || [condition])

then

cmd

else

cmd

fi

if method()

method return 0=true ,1=false

文件表达式

-e filename 如果 filename存在，则为真  
-d filename 如果 filename为目录，则为真   
-f filename 如果 filename为常规文件，则为真  
-L filename 如果 filename为符号链接，则为真  
-r filename 如果 filename可读，则为真   
-w filename 如果 filename可写，则为真   
-x filename 如果 filename可执行，则为真  
-s filename 如果文件长度不为0，则为真  
-h filename 如果文件是软链接，则为真

整数变量

eq:=

ne:!=

gt:>

ge:>=

lt: <

le <=

函数

LaunchJDeveloper(){

DisplayBanner

}

### expr

手工命令行计数器



模式匹配