Linux 运维必备的 40 个命令总结，收好了~

DevOps技术栈 今天

> 作者：AIOPS_DBA
>
> 链接：https://blog.51cto.com/wangwei007/1100991

1、删除0字节文件

```
find -type f -size 0 -exec rm -rf {} \;
```

2、查看进程

按内存从大到小排列

```
PS -e -o "%C : %p : %z : %a"|sort -k5 -nr
```

3、按 CPU 利用率从大到小排列

```
ps -e -o "%C : %p : %z : %a"|sort -nr
```

4、打印 cache 里的URL

```
grep -r -a jpg /data/cache/* | strings | grep "http:" | awk -F'http:' '{print
```

5、查看 http 的并发请求数及其 TCP 连接状态：

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

6、 `sed -i '/Root/s/no/yes/' /etc/ssh/sshd_config` sed 在这个文里 Root 的一行，匹配 Root 一行，将 no 替换成 yes。

7、如何杀掉 MySQL 进程

```
ps aux |grep mysql |grep -v grep  |awk '{print $2}' |xargs kill -9 (从中了解到a

killall -TERM mysqld

kill -9 `cat /usr/local/apache2/logs/httpd.pid`    试试查杀进程PID
```

8、显示运行 3 级别开启的服务:

```
ls /etc/rc3.d/S* |cut -c 15-    (从中了解到cut的用途，截取数据)
```

9、如何在编写 SHELL 显示多个信息，用 EOF

```
cat << EOF
+----------------------------------------------------------------+
|       === Welcome to Tunoff services ===                      |
+----------------------------------------------------------------+
EOF
```

10、for 的巧用（如给 MySQL 建软链接）

```
cd /usr/local/mysql/bin
for i in *
do ln /usr/local/mysql/bin/$i /usr/bin/$i
done
```

11、取 IP 地址

```
ifconfig eth0 |grep "inet addr:" |awk '{print $2}'| cut -c 6-
或者
ifconfig | grep 'inet addr:'| grep -v '127.0.0.1' | cut -d: -f2 | awk '{ print
```

12、内存的大小

```
free -m |grep "Mem" | awk '{print $2}'
```

13、查看80端口建立连接

```
netstat -an -t | grep ":80" | grep ESTABLISHED | awk '{printf "%s %s\n",$5,$6}
```

14、查看 Apache 的并发请求数及其 TCP 连接状态

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

15、因为同事要统计一下服务器下面所有的 jpg 的文件的大小，写了个 SHELL 给他来统计。原来用 xargs 实现，但他一次处理一部分。搞的有多个总和……，下面的命令就能解决。

```
find / -name *.jpg -exec wc -c {} \;|awk '{print $1}'|awk '{a+=$1}END{print a}
```

CPU 的数量（多核算多个CPU，`cat /proc/cpuinfo |grep -c processor`）越多，系统负载越低，每秒能处理的请求数也越多。

16、CPU负载

```
cat /proc/loadavg
```

检查前三个输出值是否超过了系统逻辑 CPU 的4倍。

17、 CPU负载

```
mpstat 1 1
```

检查 %idle 是否过低（比如小于5%）。

18、内存空间

```
free
```

检查 free 值是否过低，也可以用 `# cat /proc/meminfo`

19、SWAP 空间

```
free
```

检查 swap used 值是否过高，如果 swap used 值过高，进一步检查 swap 动作是否频繁：

```
vmstat 1 5
```

观察 si 和 so 值是否较大

20、磁盘空间

```
df -h
```

检查是否有分区使用率（Use%）过高（比如超过90%）如发现某个分区空间接近用尽，可以进入该分区的挂载点，用以下命令找出占用空间最多的文件或目录：

```
du -cks * | sort -rn | head -n 10
```

21、磁盘 I/O 负载

```
iostat -x 1 2
```

检查I/O使用率（%util）是否超过 100%

22、网络负载

```
sar -n DEV
```

检查网络流量（rxbyt/s, txbyt/s）是否过高

23、网络错误

```
netstat -i
```

检查是否有网络错误（drop fifo colls carrier），也可以用命令：# cat /proc/net/dev

24、网络连接数目

```
netstat -an | grep -E “^(tcp)” | cut -c 68- | sort | uniq -c | sort -n
```

25、进程总数

```
ps aux | wc -l
```

检查进程个数是否正常 (比如超过250)

26、可运行进程数目

```
vmwtat 1 5
```

列给出的是可运行进程的数目，检查其是否超过系统逻辑 CPU 的 4 倍

27、进程

```
top -id 1
```

观察是否有异常进程出现。

28、用户

```
who | wc -l
```

检查登录用户是否过多 (比如超过50个)　也可以用命令：# uptime。

29、系统日志

```
# cat /var/log/rflogview/*errors
```

检查是否有异常错误记录　也可以搜寻一些异常关键字，例如：

```
grep -i error /var/log/messages
grep -i fail /var/log/messages
```

30、核心日志

```
dmesg
```

检查是否有异常错误记录。

31、系统时间

```
date
```

检查系统时间是否正确。

32、打开文件数目

```
lsof | wc -l
```

检查打开文件总数是否过多。

33、日志

```
# logwatch –print
```

配置 /etc/log.d/logwatch.conf，将 Mailto 设置为自己的 email 地址，启动 mail 服务(sendmail或者postfix)，这样就可以每天收到日志报告了。

缺省 logwatch 只报告昨天的日志，可以用 # logwatch –print –range all 获得所有的日志分析结果。

可以用 # logwatch –print –detail high 获得更具体的日志分析结果(而不仅仅是出错日志)。

34、杀掉80端口相关的进程

```
lsof -i :80|grep -v “ID”|awk ‘{print “kill -9”,$2}’|sh
```

35、清除僵死进程

```
ps -eal | awk '{ if ($2 == "Z") {print $4}}' | kill -9
```

36、tcpdump 抓包，用来防止80端口被人攻击时可以分析数据

```
tcpdump -c 10000 -i eth0 -n dst port 80 > /root/pkts
```

37、然后检查IP的重复数并从小到大排序 注意 “-t\ +0” 中间是两个空格

```
# less pkts | awk {'printf $3"\n"'} | cut -d. -f 1-4 | sort | uniq -c | awk {'
```

38、查看有多少个活动的 php-cgi 进程

```
netstat -anp | grep php-cgi | grep ^tcp | wc -l
```

39、查看系统自启动的服务

```
chkconfig --list | awk '{if ($5=="3:on") print $1}'
```

40、kudzu 查看网卡型号

```
kudzu --probe --class=network
```

常用正则表达式

1.匹配中文字符的正则表达式： `[\u4e00-\u9fa5]`

> 评注：匹配中文还真是个头疼的事，有了这个表达式就好办了

2.匹配双字节字符(包括汉字在内)： `[^\x00-\xff]`

> 评注：可以用来计算字符串的长度（一个双字节字符长度计2，ASCII字符计1）

3.匹配空白行的正则表达式： `\n\s*\r`

> 评注：可以用来删除空白行

4.匹配 HTML 标记的正则表达式： `<(\S*?)[^>]*>.*?</\1>|<.*? />`

> 评注：网上流传的版本太糟糕，上面这个也仅仅能匹配部分，对于复杂的嵌套标记依旧无能为力

5.匹配首尾空白字符的正则表达式： `^\s*|\s*$`

> 评注：可以用来删除行首行尾的空白字符(包括空格、制表符、换页符等等)，非常有用的表达式

6.匹配Email地址的正则表达式：

```
\w+([-+.]\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*
```

> 评注：表单验证时很实用

7.匹配网址URL的正则表达式：`[a-zA-z]+://[^\s]*`

> 评注：网上流传的版本功能很有限，上面这个基本可以满足需求

8.匹配帐号是否合法(字母开头，允许5-16字节，允许字母数字下划线)：`^[a-zA-Z][a-zA-Z0-9_]{4,15}$`

> 评注：表单验证时很实用

9.匹配国内电话号码：`\d{3}-\d{8}|\d{4}-\d{7}`

> 评注：匹配形式如 0511-4405222 或 021-87888822

10.匹配腾讯QQ号：`[1-9][0-9]{4,}`

> 评注：腾讯QQ号从10000开始

11.匹配中国邮政编码：`[1-9]\d{5}(?!\d)`

> 评注：中国邮政编码为6位数字

12.匹配×××：`\d{15}|\d{18}`

> 评注：中国的×××为15位或18位

13.匹配ip地址：`\d+\.\d+\.\d+\.\d+`

> 评注：提取 IP 地址时有用

14.匹配特定数字：

```
^[1-9]\d*$     //匹配正整数
^-[1-9]\d*$   //匹配负整数
^-?[1-9]\d*$    //匹配整数
^[1-9]\d*|0$  //匹配非负整数（正整数 + 0）
^-[1-9]\d*|0$   //匹配非正整数（负整数 + 0）
^[1-9]\d*\.\d*|0\.\d*[1-9]\d*$    //匹配正浮点数
^-([1-9]\d*\.\d*|0\.\d*[1-9]\d*)$  //匹配负浮点数
^-?([1-9]\d*\.\d*|0\.\d*[1-9]\d*|0?\.0+|0)$  //匹配浮点数
^[1-9]\d*\.\d*|0\.\d*[1-9]\d*|0?\.0+|0$   //匹配非负浮点数（正浮点数 + 0）
^(-([1-9]\d*\.\d*|0\.\d*[1-9]\d*))|0?\.0+|0$  //匹配非正浮点数（负浮点数 + 0）
```

评注：处理大量数据时有用，具体应用时注意修正

15.匹配特定字符串：

```
^[A-Za-z]+$  //匹配由26个英文字母组成的字符串
^[A-Z]+$  //匹配由26个英文字母的大写组成的字符串
^[a-z]+$  //匹配由26个英文字母的小写组成的字符串
^[A-Za-z0-9]+$  //匹配由数字和26个英文字母组成的字符串
^\w+$  //匹配由数字、26个英文字母或者下划线组成的字符串
```

评注：最基本也是最常用的一些表达式

- END -

推荐阅读

运维老兵对运维中常见技术类问题剖析

系统架构性能优化思路

Kubernetes 的这些原理，你一定要了解

tcpdump抓包利器：从网络获取原始数据

入侵Linux服务器，黑客惯用手法：提权

一文搞懂蓝绿发布、灰度发布和滚动发布

如何部署一个生产级别的 Kubernetes 应用

k8s技术圈

金字塔原理

架构师之路