# Linux 系统管理

## 进程管理

### 信号

#### 所有支持的信号

[root@localhost test]# 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

#### 发送停止信号

[root@localhost test]# kill -STOP 2120

再看2120进程的信息

通过 top -p 2120

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

2120 influxdb 20 0 563m 77m 8556 T 0.0 7.8 39:23.52 influxd

可以看出，S这列状态为T

S列表示进程状态，有下面5种  
            D=不可中断的睡眠状态  
            R=运行  
            S=睡眠  
            T=跟踪/停止  
            Z=僵尸进程

#### 发送恢复进程

kill -CONT 2120

通过top -p 2120 可以看到S列的状态已经变为S

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

2120 influxdb 20 0 563m 77m 8556 S 1.3 7.8 39:25.34 influxd

#### 重启进程

让程序优雅地重启，在更改配置文件后，通过SIGHUP信号，再重启运行进程，可以让daemon重启

kill -HUP PID

如果HUP信号也无法让程序退出，则可以试试TERM,如果还不行，再用KILL

#### 无法捕捉和忽略的信号

##### KILL

要小心的使用这个命令，因为如果程序就正常终止，很可有会产生残余数据，如没有清理Lock文件，会对下次启动造成问题。

##### STOP

## 内核管理

### 选择默认启动的内核版本

vi /boot/grub/grub.conf

将default由原来的0改为要默认的值

### 删除多余的内核

1. uname -a 查看当前运行的内核版本
2. rpm -qa|kernel 查看已经安装的所有内核版本
3. yum remove kernel-xxxxx 删除多余的内核

## 系统管理

### vmware 安装的centos6.5 无法解析域名

dns用宿主的就解决了

### 修改hostname

vi /etc/sysconfig/network -> HOSTNAME=new\_hostname

echo new\_hostname > /proc/sys/kernel/hostname

## 网络管理

### /etc/sysctl.conf

#### 优化

vim /etc/sysctl.conf

#编辑文件，加入以下内容：

net.ipv4.tcp\_syncookies = 1

net.ipv4.tcp\_tw\_reuse = 1

net.ipv4.tcp\_tw\_recycle = 1

net.ipv4.tcp\_fin\_timeout = 30

#然后执行 /sbin/sysctl -p 让参数生效。

配置说明：

net.ipv4.tcp\_syncookies = 1 表示开启SYN Cookies。当出现SYN等待队列溢出时，启用cookies来处理，可防范少量SYN攻击，默认为0，表示关闭；

net.ipv4.tcp\_tw\_reuse = 1 表示开启重用。允许将TIME-WAIT sockets重新用于新的TCP连接，默认为0，表示关闭；

net.ipv4.tcp\_tw\_recycle = 1 表示开启TCP连接中TIME-WAIT sockets的快速回收，默认为0，表示关闭；

net.ipv4.tcp\_fin\_timeout=30修改系統默认的 TIMEOUT 时间。

再进一步

net.ipv4.tcp\_keepalive\_time = 1200

#表示当keepalive起用的时候，TCP发送keepalive消息的频度。缺省是2小时，改为20分钟。

net.ipv4.ip\_local\_port\_range = 1024 65000

#表示用于向外连接的端口范围。缺省情况下很小：32768到61000，改为1024到65000。

net.ipv4.tcp\_max\_syn\_backlog = 8192

#表示SYN队列的长度，默认为1024，加大队列长度为8192，可以容纳更多等待连接的网络连接数。

net.ipv4.tcp\_max\_tw\_buckets = 5000

#表示系统同时保持TIME\_WAIT套接字的最大数量，如果超过这个数字，TIME\_WAIT套接字将立刻被清除并打印警告信息。

默认为180000，改为5000。对于Apache、Nginx等服务器，上几行的参数可以很好地减少TIME\_WAIT套接字数量，但是对于 Squid，效果却不大。此项参数可以控制TIME\_WAIT套接字的最大数量，避免Squid服务器被大量的TIME\_WAIT套接字拖死。

### iptables

#### 三条链

INPUT链 – 处理来自外部的数据。

OUTPUT链 – 处理向外发送的数据。

FORWARD链 – 将数据转发到本机的其他网卡设备上。

数据流向场景

访问本机：在INPUT链上做过滤

本机访问外部：在OUTPUT链上做过滤

通过本机访问其他主机:在FORWARD链上做过滤

开启所有可以访问权限

iptables -P INPUT ACCEPT

关闭用DROP

INPUT 是输入

OUTPUT 是输出

#### 参考

http://www.360doc.com/content/14/1028/11/11991\_420541924.shtml