

进程管理

进程管理

进程

pstree 命令

静态查看进程信息

ps

pgrep

动态查看进程信息

top 终端提示符不显示

停止进程

xkill

进程优先级

指定优先级

调整优先级

前后台作业

进程管理课后作业

进程

- 进程： 运行在内存中程序实例，进程是程序运行的一种状态，是内存中的概念，进程与进程之间无法访问对方私有的内存区域。
- 线程： 程序运行的最小单元，一个进程可以派生出多个线程，同一个进程内的线程之间可以相互访问彼此内存区域，并且可以共享同一进程的共享内存区域。
- 进程编号： pid
- 父进程编号： ppid

pstree 命令

pstree 命令用以查看进程的结构

常用参数	参数说明
-a	显示每个程序的完整指令，包含路径，参数或是常驻服务的标示。
-c	不使用精简标示法。
-G	使用VT100终端机的列绘图字符。
-h	列出树状图时，特别标明执行的程序。
-H<程序识别码>	此参数的效果和指定"-h"参数类似，但特别标明指定的程序。
-l	采用长列格式显示树状图。
-n	用程序识别码排序。预设是以程序名称来排序。
-p	显示程序识别码。
-u	显示用户名称。
-U	使用UTF-8列绘图字符。
-V	显示版本信息。

```

[root@rhel6 /etc/skel]#pstree
init├─NetworkManager├─dhclient
|
└─{NetworkManager}
  ├abrt
  ├acpid
  ├anacron
  ├atd
  ├auditd──{auditd}
  ├automount──4*[{automount}]
  ├bonobo-activati──{bonobo-activat}
  ├certmonger
  ├console-kit-dae──63*[{console-kit-da}]
  ├crond
  ├cupsd
  ├2*[{dbus-daemon──{dbus-daemon}}]
  ├dbus-launch
  ├devkit-power-da
  ├gconfd-2
  ├gdm-binary├─gdm-simple-slav├─Xorg
  |
  |
  ├gdm-session-wor
  |
  |
  ├gnome-session├─at-spi-registry
  |
  |
  |
  ├gdm-simple-gree
  |
  |
  |
  ├gnome-power-man
  |
  |
  |
  ├metacity
  |
  |
  |
  ├polkit-gnome-au
  |
  |
  |
  └─{gnome-session}
    |
    |
    |
    └─{gdm-simple-sla}
      |
      |
      └─{gdm-binary}
        ├gnome-settings──{gnome-settings}
        └─gvfsd

```

```

├─hald├─hald-runner├─hald-addon-acpi
│
├─hald-addon-inpu
│
├─{hald}
├─master├─pickup|
├─qmgr
├─5*[mingetty]
├─modem-manager
├─polkitd
├─pulseaudio──{pulseaudio}
├─rhsmcertd
├─rpc.statd
├─rpcbind
├─rsyslogd──3*[{rsyslogd}]
├─rtkit-daemon──2*[{rtkit-daemon}]
├─sshd──sshd──bash──pstree
├─udev──2*[udev]
├─wpa_supplicant

```

静态查看进程信息

ps

- **ps** 查看当前标签页上的进程信息
- **ps aux** 系统进程快照
- **ps -ef** 所有进程 **f** 全格式罗列

pgrep

用来过滤进程号

```

[root@rhel6 ~]# pgrep -l ping <== 截取进程号 pid
27706 ping
[root@rhel6 ~]# pgrep -lU student <== 指定用户截取进程号 pid
27736 bash
27762 ping
[root@rhel6 ~]# pgrep -l -t pts/1 <== 指定登陆端口截取进程号 pid
2568 bash
27735 su
27736 bash
1.2.3 pidof
[root@rhel6 ~]# pidof Xorg <== 只显示进程的进程号 pid
15262

```

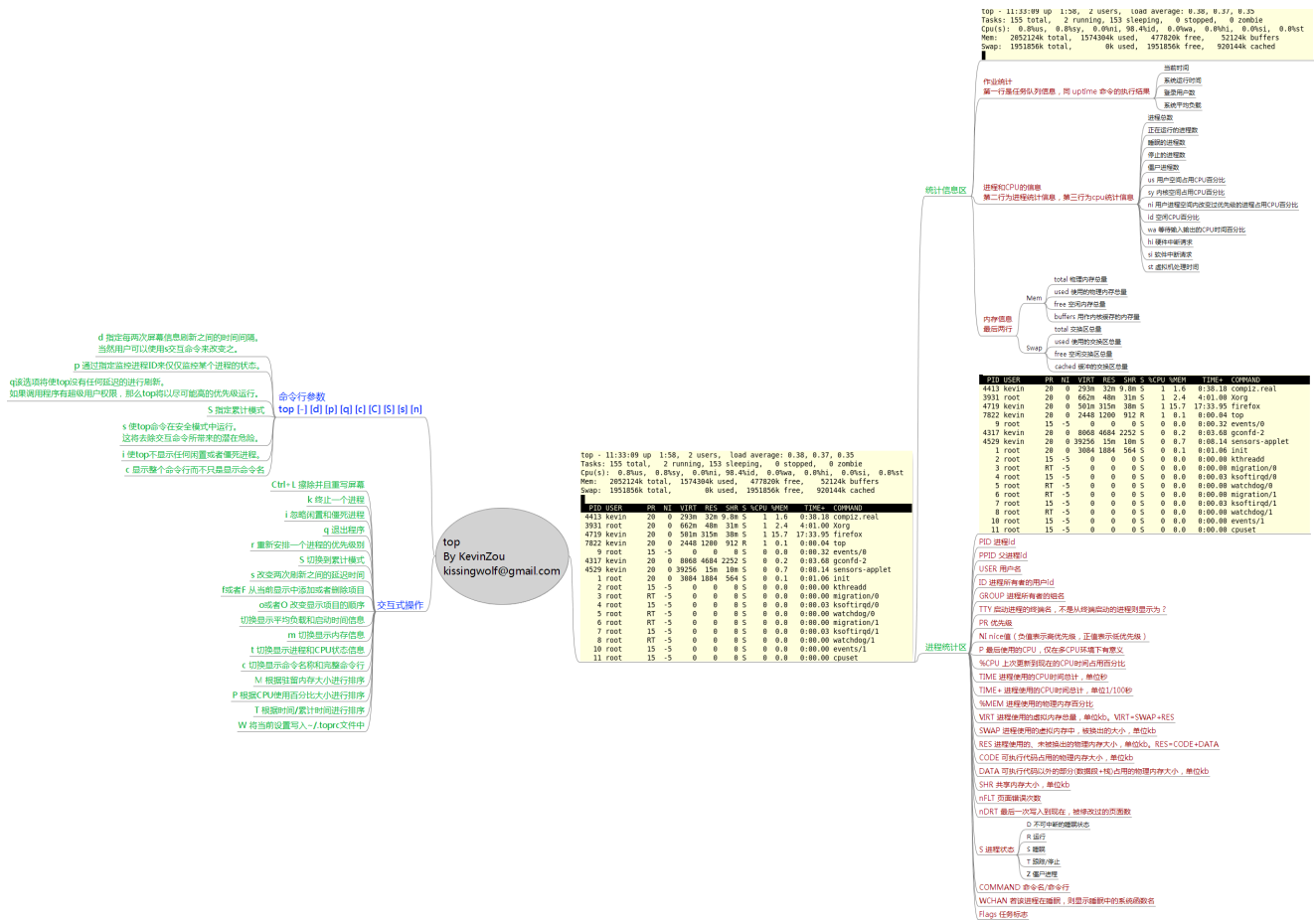
动态查看进程信息

top 终端提示符不显示

- **【d】** 修改默认刷新频率，默认 3s

- 【P】以 cup 占用百分比进行排序
- 【M】以内存的占用情况排序
- 【h】显示帮助信息
- 【<>】翻页
- 【k】杀掉进程 kill

top命令图解



停止进程

Linux中的 kill 命令用来停止指定的进程(terminate a process)的运行，是 Linux下进程管理的常用命令。

通常情况下停止一个前台进程可以使用 Ctrl+C 组合键，但是对于一个运行在后台进程需要用 kill命令来终止，我们就需要先使用 ps、pidof、pstree和top 等工具获取进程 PID，然后使用 kill 命令来杀掉该进程。

kill 命令是通过向进程发送指定的信号来结束相应进程的。在默认情况下,采用编号为15的 TERM 信号。TERM 信号将终止所有不能捕获该信号的进程。

对于那些可以捕获该信号的进程就要用编号为 9 的 kill 信号,强行“杀掉”该进程。

命令格式

kill [参数][进程号]

命令功能:

发送指定的信号到相应进程。不指定型号将发送 **SIGTERM (15)**终止指定进程。如果任无法终止该程序可用 “ - **KILL**” 参数，其发送的信号为 **SIGKILL(9)**，将强制结束进程,使用 **ps** 命令或者 **jobs** 命令可以查看进程号。**root** 用户将影响用户的进程,非 **root** 用户只能影响自己的进程。

命令参数：

参数	说明
-l	信号,若果不加信号的编号参数,则使用 “-l” 参数会列出全部的信号名称
-a	当处理当前进程时,不限制命令名和进程号的对应关系
-p	指定 kill 命令只打印相关进程的进程号,而不发送任何信号
-s	指定发送信号
-u	指定用户

xkill

哪里关不掉点哪里

主要用于停止图形化（GUI）程序

进程优先级

NI nice

数字表示：-20-19 数字越小，等级越高

```
ps -le | grep ping <== 查看 ping 的优先级
```

指定优先级

```
nice -n 3 ping 172.0.0.1
```

普通用户不能指定比 0 小的优先级，root 用户随意

普通用户只能做贡献，上帝随意设置优先级

调整优先级

```
renice -n [-20-19] [pid] <== 调整优先级
```

root 可以升高也可降低nice值，代表可以设置优先级更低可以设置优先级更高。

普通用户只能升高nice值，代表只能降低优先级。

```

[root@rhel6 ~]#nice -n 3 ping 172.25.0.10
[root@rhel6 ~]#nice -n 5 ping 172.25.0.10
[root@rhel6 ~]#ps -le|grep ping
 4 S      0 26613 26536  0 83   3 - 25812 skb_re pts/1    00:00:00 ping
 4 S      0 26614 26567  0 85   5 - 25812 skb_re pts/2    00:00:00 ping
[root@rhel6 ~]#renice -n -1 26613
26613: old priority 3, new priority -1
[root@rhel6 ~]#ps -le|grep ping
 4 S      0 26613 26536  0 79  -1 - 25812 skb_re pts/1    00:00:00 ping
 4 S      0 26614 26567  0 85   5 - 25812 skb_re pts/2    00:00:00 ping

```

前后台作业

- & 在创建进程的过程中，将前台作业放置到后台的方法，在命令后面加上 &
- jobs 查看后台进程
- fg 将后台作业调用到前台
- ctrl+z 将已经触发的前台作业调到后台，后台作业暂停
- bg 让后台作业执行

```

[root@rhel6 ~]#ping 172.25.0.11 >/dev/null &
[1] 26712
[root@rhel6 ~]#ping 172.25.0.10 >/dev/null &
[2] 26713
[#13#root@rhel6 ~]#jobs
[1]-  Running
ping 172.25.0.11 > /dev/null &
[2]+  Running
ping 172.25.0.10 > /dev/null &
[root@rhel6 ~]#fg 2
ping 172.25.0.10 > /dev/null
^Z
[2]+  Stopped
ping 172.25.0.10 > /dev/null
[root@rhel6 ~]#bg 2
[2]+  ping 172.25.0.10 > /dev/null &
[root@rhel6 ~]#ps -ef|grep ping
root 26712 26567  0 18:16 pts/2 00:00:00 ping 172.25.0.11
root 26713 26567  0 18:16 pts/2 00:00:00 ping 172.25.0.10
root 26717 26567  0 18:17 pts/2 00:00:00 grep ping
[root@rhel6 ~]#kill 26712
[root@rhel6 ~]#jobs
[1]-  Terminated
ping 172.25.0.11 > /dev/null
[2]+  Running
ping 172.25.0.10 > /dev/null &
[root@rhel6 ~]#kill 26713
[root@rhel6 ~]#jobs
[2]+  Terminated
ping 172.25.0.10 > /dev/null
[#21#root@rhel6 ~]#jobs

```

进程管理课后作业

进程管理

1. 关于父进程和子进程的实验:

打开一个终端登陆 `ssh root@rhel7-fN` 进入输入密码的状态,先不输入密码,查看进程;输入密码后再查看进程。

rhel7

```
[root@rhel7 ~]# ps -ef|grep bash
```

```
root      642      1  0 02:46 ?        00:00:00 /bin/bash /usr/sbin/ksmtuned
root     7844   7839  1 06:29 pts/0    00:00:00 -bash
```

```
[root@rhel7 ~]# ps -ef|grep ssh
```

```
root     1261      1  0 02:46 ?        00:00:00 /usr/sbin/sshd -D
root     7839   1261  0 06:29 ?        00:00:00 sshd: root@pts/0
root     7892   1261  0 06:29 ?        00:00:00 sshd: root [priv]
sshd     7893   7892  0 06:29 ?        00:00:00 sshd: root [net]
root     7906   7844  0 06:30 pts/0    00:00:00 grep --color=auto ssh
```

```
[root@rhel7 ~]# ps -ef|grep bash
```

```
root      642      1  0 02:46 ?        00:00:00 /bin/bash /usr/sbin/ksmtuned
root     7844   7839  0 06:29 pts/0    00:00:00 -bash
root     7915   7892  0 06:30 pts/1    00:00:00 -bash
root     7957   7844  0 06:30 pts/0    00:00:00 grep --color=auto bash
```

```
[root@rhel7 ~]# ps -ef|grep 7892
```

```
root     7892   1261  0 06:29 ?        00:00:00 sshd: root@pts/1
root     7915   7892  0 06:30 pts/1    00:00:00 -bash
```

```
[root@rhel7 ~]# ps -ef|grep 1261
```

```
root     1261      1  0 02:46 ?        00:00:00 /usr/sbin/sshd -D
root     7839   1261  0 06:29 ?        00:00:00 sshd: root@pts/0
root     7892   1261  0 06:29 ?        00:00:00 sshd: root@pts/1
```

```
[root@rhel7 ~]# ps -ef|grep 1|head -n 1
```

```
root      1      0  0 02:46 ?        00:00:03 /usr/lib/systemd/systemd --switched-root --system
--deserialize 20
```

rhel6

```
[root@rhel6 ~]#ps -ef|grep ssh
```

```
root     1421      1  0 15:56 ?        00:00:00 /usr/sbin/sshd
root    26564   1421  0 17:36 ?        00:00:00 sshd: root@pts/2
root    26750   1421  0 18:25 ?        00:00:00 sshd: root@pts/3
root    26829  26803  0 18:29 pts/3    00:00:00 ssh root@172.25.0.10
root    26848  26567  0 18:37 pts/2    00:00:00 grep ssh
```

```
[root@rhel6 ~]#ps -ef|grep ssh
```

```
root     1421      1  0 15:56 ?        00:00:00 /usr/sbin/sshd
root    26564   1421  0 17:36 ?        00:00:00 sshd: root@pts/2
root    26750   1421  0 18:25 ?        00:00:00 sshd: root@pts/3
root    26829  26803  0 18:29 pts/3    00:00:00 ssh root@172.25.0.10
root    26849   1421  0 18:37 ?        00:00:00 sshd: [accepted]
sshd    26850  26849  0 18:37 ?        00:00:00 sshd: [net]
root    26852  26567  0 18:37 pts/2    00:00:00 grep ssh
```

```
[root@rhel6 ~]#ps -ef|grep ssh
```

```
root     1421      1  0 15:56 ?        00:00:00 /usr/sbin/sshd
root    26564   1421  0 17:36 ?        00:00:00 sshd: root@pts/2
root    26750   1421  0 18:25 ?        00:00:00 sshd: root@pts/3
root    26829  26803  0 18:29 pts/3    00:00:00 ssh root@172.25.0.10
root    26849   1421  1 18:37 ?        00:00:00 sshd: root@pts/0
root    26874  26567  0 18:38 pts/2    00:00:00 grep ssh
```

```
[root@rhel6 ~]#ps -ef|grep 1421
```

```
root     1421      1  0 15:56 ?        00:00:00 /usr/sbin/sshd
root    26564   1421  0 17:36 ?        00:00:00 sshd: root@pts/2
root    26750   1421  0 18:25 ?        00:00:00 sshd: root@pts/3
```

```
root      26849  1421  0 18:37 ?          00:00:00 sshd: root@pts/0
root      26877 26567  0 18:38 pts/2      00:00:00 grep 1421
[root@rhel6 ~]#ps -ef|grep 1|head -n 1
root          1      0  0 15:55 ?          00:00:01 /sbin/init
```

1. 创建多个vi 进程，并使其运行在系统后台，设置vi 后台进程的优先级（nice）值，分别为1、5、15、17，并随后将其统一修改为18。
2. 观察top命令的显示信息，找出当前占用cpu、内存、I/O资源最多的进程号和进程名。