GNU/Linux
系统资源管理命令

系统资源管理命令 -du

命令 :du

功能 : 查看目录 / 文件使用磁盘空间的大小

语法格式： du［ 参数 ］[文件或文件夹]

参数

-a 所有文件包含子目录

-b 输出的结果以 bytes 为单位

-c 只计算目录的总用量

系统资源管理命令 -du

参数：

-h 以 1024 进制进行单位换算

-s 只显示目录使用空间的总和

--inodes 查看所用 i- 节点数量

示例

1. 查看 /home 目录使用的总和，并进行单位换算

du -sh /home

系统资源管理命令 -df

命令 :df
功能 : 查看硬盘分区空间使用情况

语法格式格式： df [选项] [分区]

参数
-h 以 1024 进制进行单位换算
-H 以 1000 进制进行单位换算
-t 指定显示分区的类型
-i 显示分区的 inode 使用情况

系统资源管理命令 -free

命令 :free
功能 : 查看内存所使情况

命令格式：free　　[选项]　　[分区]
-m　　以 1024KB=1M 计算
-g　　以 1024MB=1G 计算

系统资源管理命令 - 进程与线程

Linux 属于多用户、多任务的操作系统，其特性在于：

1. 多用户是：

指多个用户在同一时间使用计算机系统；

2. 多任务是：

指 Linux 可以同时执行几个任务，它可以在还未执行完一个任务时又执行另一项任务。

系统资源管理命令 - 进程与线程

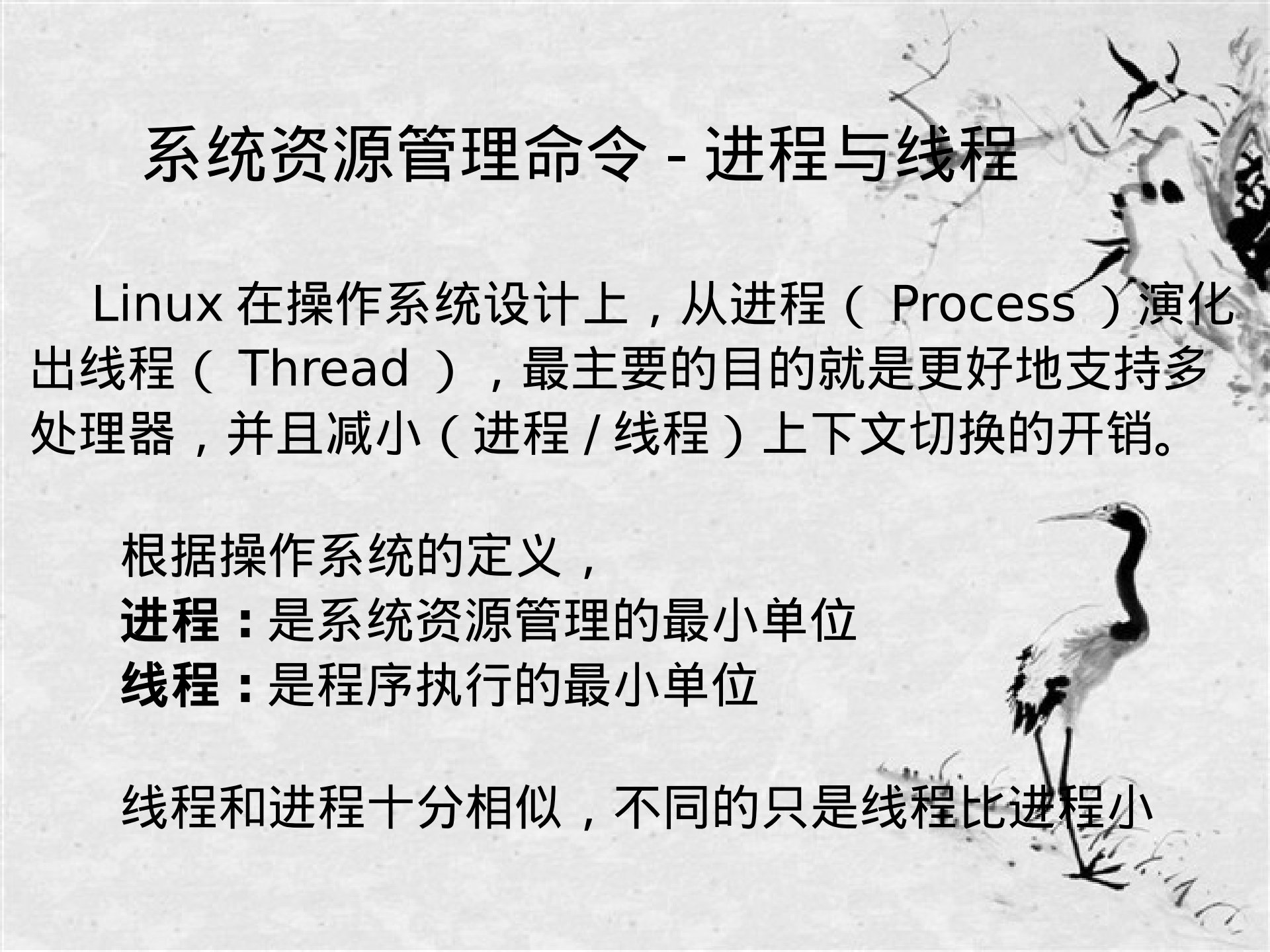
Linux 在操作系统设计上，从进程（ Process ）演化出线程（ Thread ），最主要的目的就是更好地支持多处理器，并且减小（进程 / 线程）上下文切换的开销。

根据操作系统的定义，

进程：是系统资源管理的最小单位

线程：是程序执行的最小单位

线程和进程十分相似，不同的只是线程比进程小



系统资源管理命令 - 进程与线程

线程：

1. 采用了多个线程可共享资源的设计思想。它们的操作大部分都是在同一地址空间进行的。

2. 从一个线程切换到另一线程所花费的代价比进程低。

3. 进程本身的信息在内存中占用的空间比线程大。

系统资源管理命令 - 进程与线程

4. 线程更能允分地利用内存

5. 线程可以看作是在进程内部执行的指定序列。

6. 线程和进程的最大区别在于线程完全共享相同的地址空间，运行在同一地址上。

系统资源管理命令 - 进程与线程

7．多个程序在同一时间请求，CPU 根据”先进先出”原则执行线程。而其他的线程则在线程队列中等待。

系统资源管理命令 - 进程与线程

进程定义：

在自身虚拟地址空间运行的一个单独的程序。

进程说明：

1. 程序在未被执行的时候，被当做静态指令的集合。

2. 程序被执行后称之为"作业"或"任务"

系统资源管理命令 - 进程与线程

进程说明：

3. 每个任务至少拥有一个进程作为此任务在系统中的代表。

4. 操作系统利用分时管理的方法使所有的"任务"进程"共同分享系统资源。主动权属于操作系统。

5. 当某个线程死掉，则可以通过对其管理的进程进行操作，从而可以将此进程所代表的任务结束掉。

系统资源管理命令 - 进程与任务

进程和任务的区别：

一个正在执行的进程成为一个任务
一个任务可以包含多个进程

因此对任务的控制即是对正在运行的进程进行控制

系统资源管理命令 - 进程与线程

Linux 系统下进程的管理：

1. Linux 系统中的所有进程都是相互关联的

2. 除了初始化进程外，多个进程都有一个父进程。

3. 新的进程实际是通过复制得来得。

系统资源管理命令 - 进程与线程

Linux 系统最高进程 :

Linux 系统启动后即产生了第一个进程。即 systemd 进程。此进程的 PID 号 (进程号) 为 1.

所有子进程都是通过 (父进程)systemd 进程衍生 (fork) 的。

如 :

```
systemd(init)---
                |
                |---shell
                          |
                          |------top
```

系统资源管理命令 -systemd

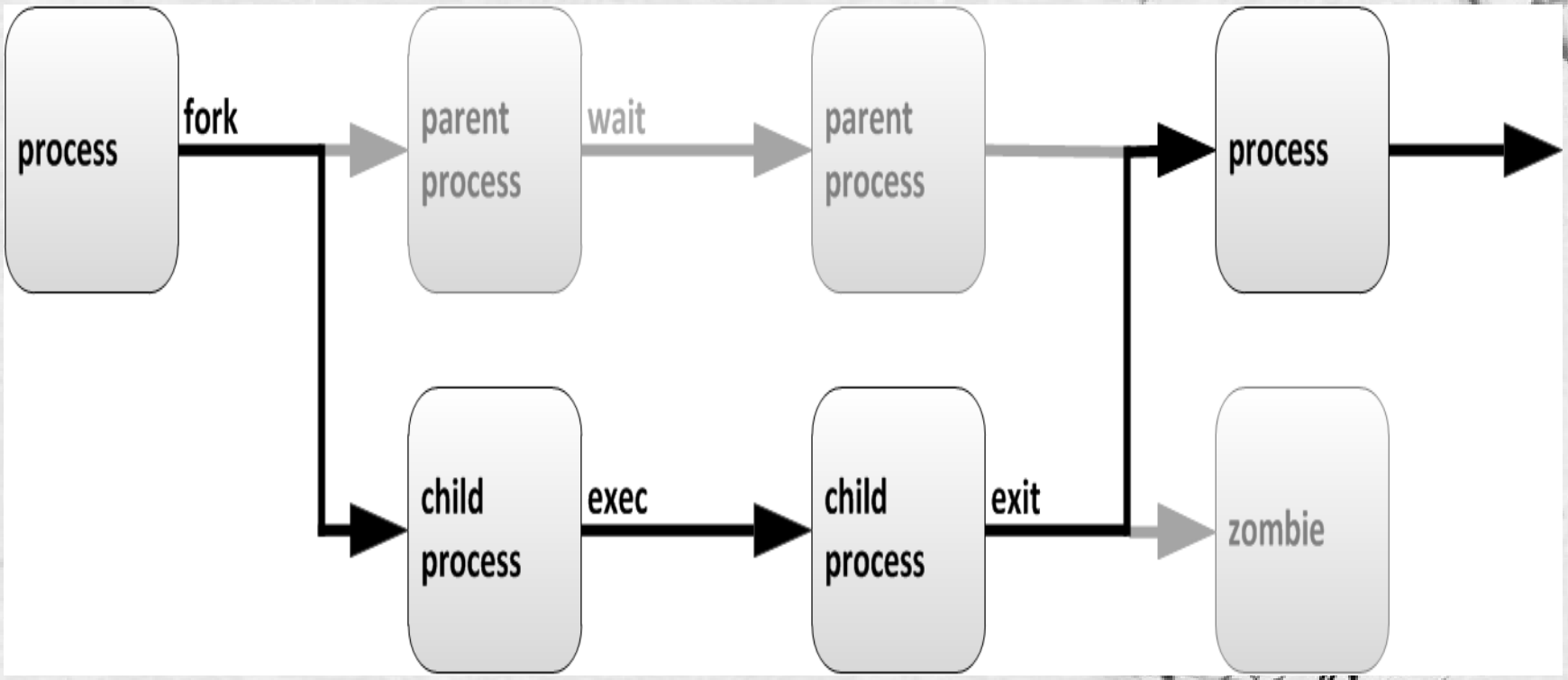
systemd 进程的作用：

1．扮演终结父进程的角色。因为 systemd 进程永远不会被终止，所以系统总是可以确信它的存在，并在必要的时候以它为参照。

2．如果某个进程在它衍生出来的全部子进程结束之前被终止，就会出现必须以 systemd 为参照。此时那些失去了父进程的子进程就都会以 systemd 作为它们的父进程。

系统资源管理命令

进程声明周期示意图



系统资源管理命令 -ps

命令 :ps

功能 : 显示当前系统内进程信息及状态等。

语法格式 :ps [options]

系统资源管理命令 -ps

选项：

a　显示当前终端下所有用户的进程

x　选择所有不在当前终端下的进程。

u　查看进程的 UID 或账户名

w　列加宽，可显示更多的信息，可重复使用

系统资源管理命令 -ps

选项：

e　　选择所有的进程。

f　　列示完整的列表

l　　显示进程的所属者，进程号和父进程号。

系统资源管理命令 -ps

ps aux 输出结果：

USER: 程序是以哪个用户名的名义运行。

PID: 进程号 ,PID 号范围为 1-32768, 至最高值 , 则循环。

%CPU: 进程 的 CPU 使用率

%MEM: 进程的 MEM 使用率

系统资源管理命令 -ps

ps aux 输出结果：

VSZ: 进程所使用的虚拟内存大小 (Virtual Size)

RSS: 进程使用的驻留集大小或实际内存的大小 (Kbytes)

TTY: 进程在哪个 TTY 执行的

STAT: 进程的状态

系统资源管理命令 -ps

STAT 进程状态：

D　不可中断的休眠（一般为 I/O), 须直到有中断发生

R　运行状态（正在运行队列中）

S　休眠状态

I　空闲状态

T　终止，收到终止指令后停止运行

Z　僵尸进程

P　等待交换页

W　没有足够的内存页可分配

系统资源管理命令 -ps

STAT 进程状态：

X　退出状态，集成即将被销毁 .(此意味着进程彻底被释放)

<　高优先级

N　低优先级

L　内存锁页

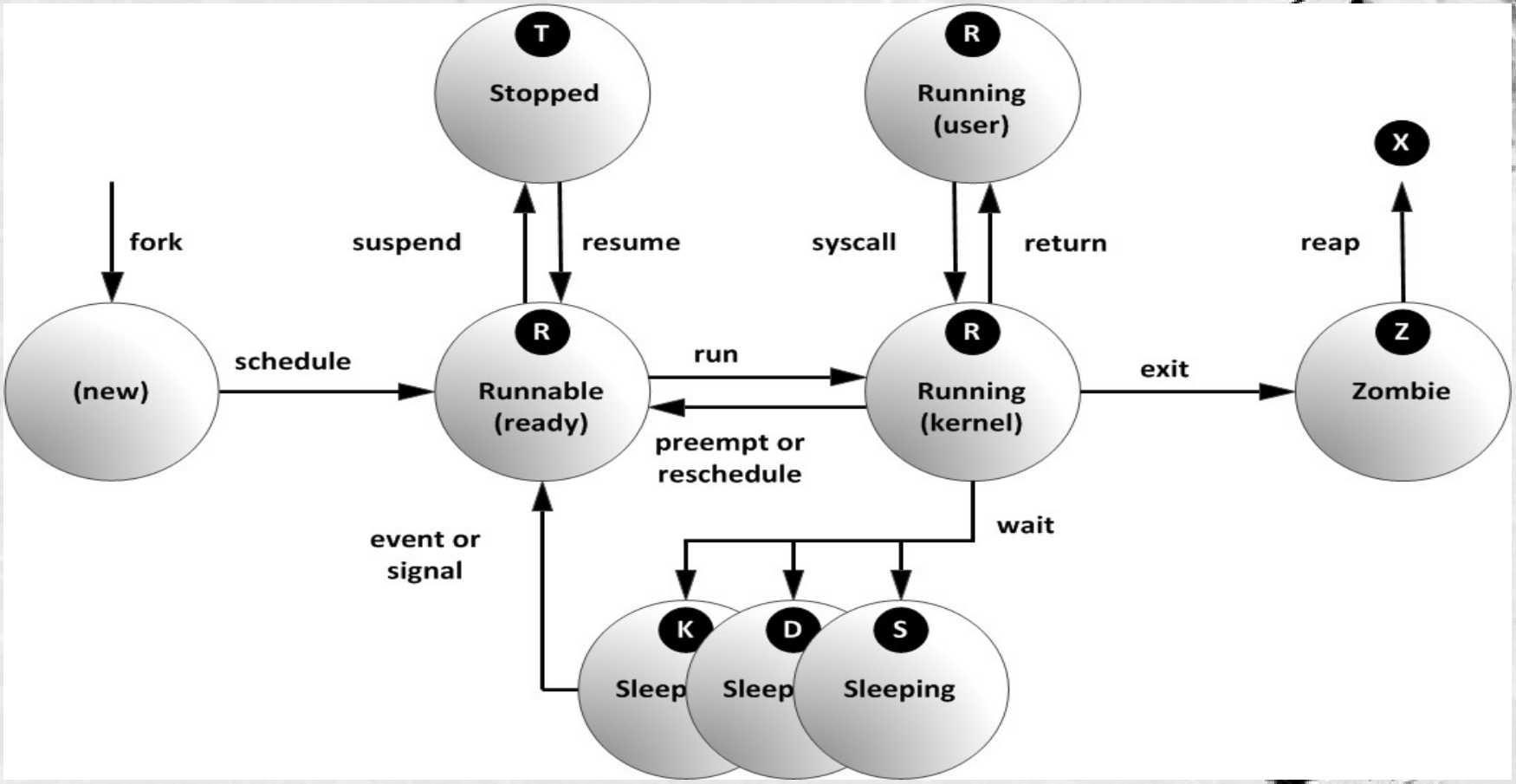
s　进程的管理者 (一般表示其下有子进程)

l　多进程

+　位于后台的进程组

系统资源管理命令

进程状态



系统资源管理命令 -ps

ps aux 输出结果：

START: 进程启动时间及日期

TIME: 进程使用的总 CPU 时间

COMMAND: 正在执行的命令及参数

系统资源管理命令 -ps

ps elf 输出结果：

F: 标记 .

1. 进程 forked(交叉 ），但未执行 exec 调用

4. 用特权用户权限

S: 进程状态，以 1 个字符进行表示

PID: 执行此进程的 PID

PPID: 此进程的父进程号

系统资源管理命令 -ps

输出结果：

C: 进程 CPU 的使用率，为整数

PRI: 进程优先级编号

NI: 优先级

ADDR: 进程所在内存地址

SZ: 实际占用的物理内存大小

系统资源管理命令 -ps

输出结果：

WCHAN: 使用 kernel 函数的进程处于休眠状态

STIME: 进程启动的时间

TTY: 进程在哪个终端执行

TIME: 进程执行所花费的时间

CMD: 执行的命令及参数

系统资源管理命令 -ps

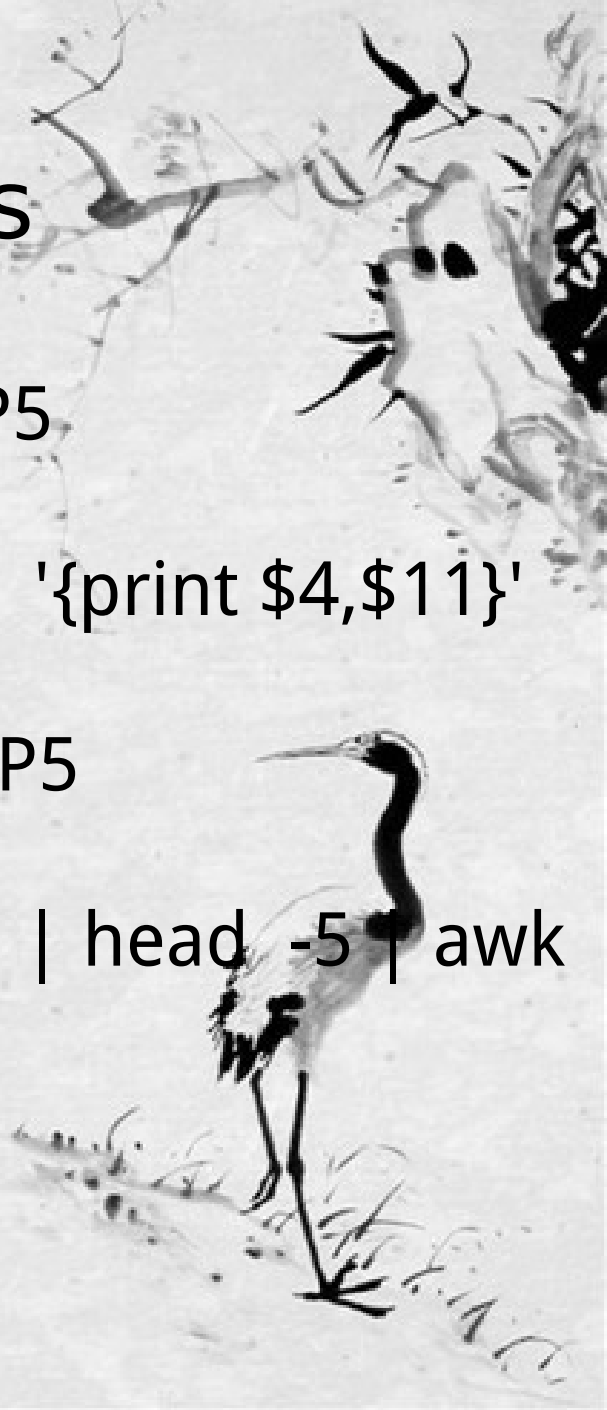
示例 :

1. 查看系统占用内存最高的进程的 TOP5

```
#ps aux | sort -rn -k4 | head  -5 | awk  '{print $4,$11}'
```

2. 查看系统占用 CPU 最高的进程的 TOP5

```
#ps aux | sort -rn -k3 | sed  '/%CPU/d' | head  -5 | awk
  '{print $3,$11}'
```



系统资源管理命令 -ps

示例：

3. 生成新的报表

```
#ps -o pid,pcpu,nice,comm
```

4. 生成新的进程报表

```
#ps -axef -o comm,pid,nice,pcpu
```

//* 更多队列信息可参看 man 1 ps

系统资源管理命令 -pstree

命令 :pstree

功能 : 进程以树型结构显示

语法格式 :

pstree [options]

参数

-u 在进程名旁显示进程所属的用户名。

-p 在进程名旁显示 PID 。

-a 显示命令的详细信息。

系统资源管理命令 -pstree

示例

1. 查看进程树
   #pstree

2. 查看进程树并显示进程 ID 及属主
   #pstree -pu

3. 查看账户 snow 所产生的进程及其进程 ID
   #pstree -p snow

系统资源管理命令 -top

命令 :top
功能 : 实时刷新当前系统情况
特点 :top 有许多内置命令。

内置命令 1:

h	帮助。
q	退出。
space	立刻刷新。
s	设置刷新时间，单位为秒。
k	杀掉一个进程。
r	定义一个进程的优先级。

系统资源管理命令 -top

top 结果输出：

第 1 行：
当前系统时间

uptime 时间

当前登入系统的账户总数

当前系统 1、 5、 15 分钟的系统负载值 (即任务队列的平均长度), 数值一般超过 5 即负载过大。

系统资源管理命令 -top

top 结果输出：

第 2 行 Tasks(任务 / 进程):

当前进程总数

运行状态的进程总数

休眠状态的进程总数

僵尸状态的进程总数

系统资源管理命令 -top

top 结果输出：

第 3 行 CPU 状态：

us: 用户空间占用 CPU%

sy:kernel 空间占用 CPU%

ni: 改变过优先级的进程占 CPU%

id: 空闲 CPU%

系统资源管理命令 -top

top 结果输出：

wa:IO 等待占用 CPU%

hi: 硬中断占用 CPU%

si: 软中断占用 CPU%

st:Xen Hypervisor 服务分配给虚拟机上的任务占用 CPU%

系统资源管理命令 -top

top 结果输出：

第 4 行 : 内存 (单位 kb)

物理内存总数

使用的内存总数

空闲内存总数

缓存总数

系统资源管理命令 -top

top 结果输出：

第 5 行 :swap(单位 kb)

swap 总数

使用 swap 的内存总数

swap 空闲内存总数

缓冲交换区总数

系统资源管理命令 -top

top 结果输出：

第 7 行 : 各进程的状态监控

PID: 进程 ID

USER: 进程使用者

PR: 进程优先级

NI: 优先级值

系统资源管理命令 -top

top 结果输出：

VIRT: 使用的虚拟内存总量 (kb)VIRT=SWAP+RES

RES: 进程使用的，没有被置换出来的物理内存 (kb)

SHR: 共享内存大小 (kb)

S: 状态 (参看 ps)

%CPU: 进程自上次更新后到本次更新所占用的 CPU%

系统资源管理命令 -top

top 结果输出：

%MEM: 进程自上次更新后到本次更新所占用的 MEM%

TIME+: 进程使用的 CPU 时间总计，单位 1/100 秒

COMMAND: 进程生成的命令及参数

系统资源管理命令 -top

top 内置命令 2 ：

1: 按”一”键可以监控每个 CPU 及每个 CPU 各个核心情况

b: 开启 / 关闭高亮显示

x: 开启 / 关闭排序列的高亮显示 (先执行 b, 在执行此)

x 子操作 ：

(1)shift+>: 高亮向右排序列

(2)shift+<: 高亮向左排序列

系统资源管理命令 -top

top 内置命令 2 ：

M: 根据 MEM 使用率大小进程排序

P: 根据 CPU 使用率进行排序

T: 根据时间 / 累计时间进行排序

W: 将当前设置写入 ~/.toprc 配置文件

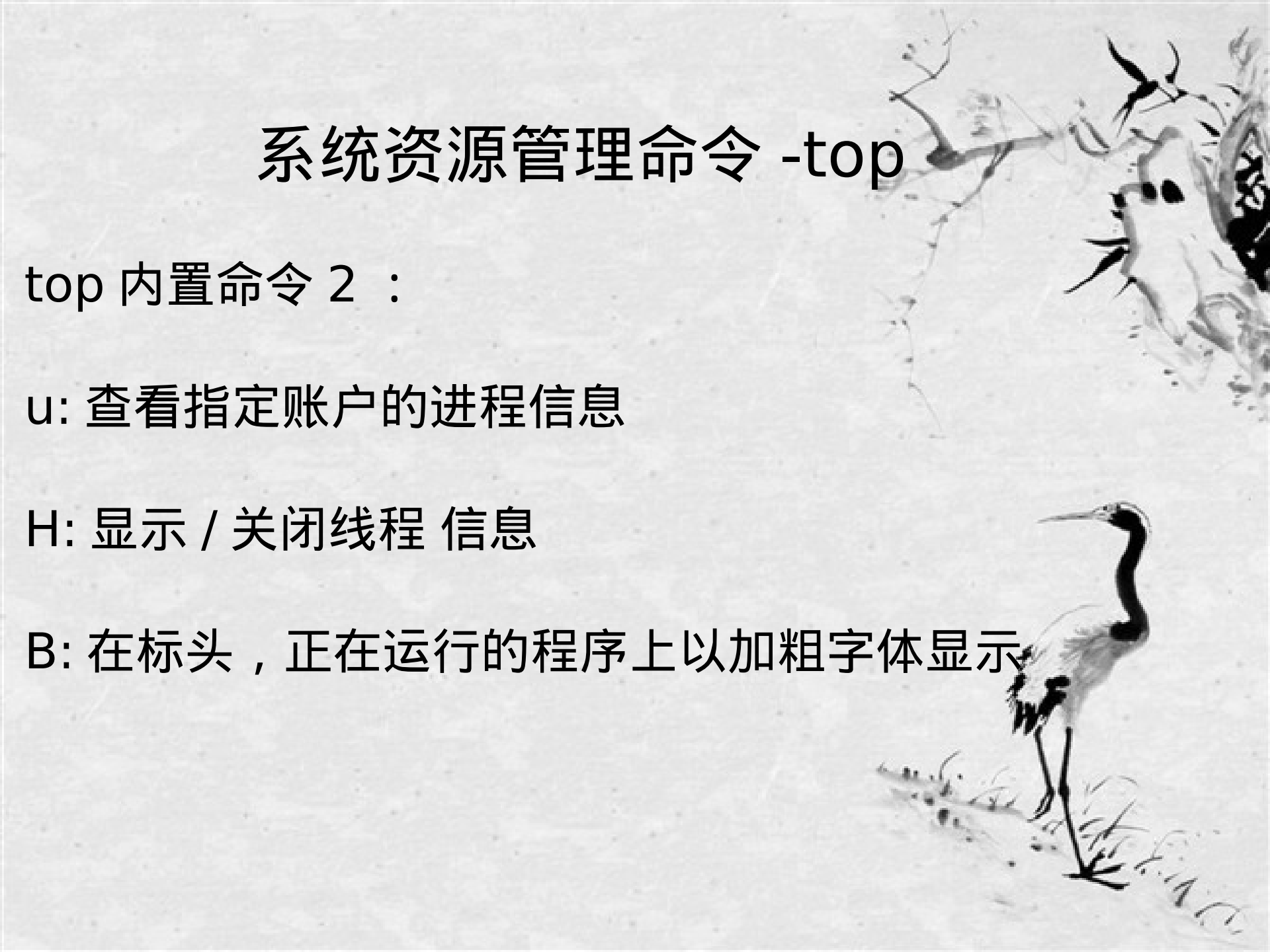
系统资源管理命令 -top

top 内置命令 2 ：

u: 查看指定账户的进程信息

H: 显示 / 关闭线程 信息

B: 在标头，正在运行的程序上以加粗字体显示



系统资源管理命令 -lsof

命令 :lsof

功能 : 显示进程所打开的文件

语法格式 :lsof [选项]

系统资源管理命令 -lsof

lsof 命令可以列出被进程所打开的文件的信息。被打开的文件可以是：

1. 普通的文件
2. 目录
3. 网络文件系统的文件
4. 字符设备文件

系统资源管理命令 -lsof

lsof 命令可以列出被进程所打开的文件的信息。被打开的文件可以是：

5.(函数) 共享库

6. 管道，命名管道

7. 符号链接

8. 底层的 socket 字流，网络 socket ， unix 域名 socket

系统资源管理命令 -lsof

lsof 命令可以列出被进程所打开的文件的信息。被打开的文件可以是：

9. 在里面，大部分的东西都是被当做文件的... .. 还有其他很多

系统资源管理命令 -lsof

lsof 输出各列信息的意义如下：

COMMAND ：进程的名称 PID ：进程标识符

USER ：进程所有者

FD ：文件描述符，应用程序通过文件描述符识别该文件。如 cwd、 txt 等 TYPE ：文件类型，如 DIR、 REG 等

系统资源管理命令 -lsof

lsof 输出各列信息的意义如下：

DEVICE ：指定磁盘的名称

SIZE ：文件的大小

NODE ：索引节点（文件在磁盘上的标识）

NAME ：打开文件的确切名称

系统资源管理命令 -lsof

示例：

1. 列出所打开的文件
   #lsof | less

2. 查看哪个进程在使用指定文件
   #lsof /filepath/file_name

3. 递归查看某个目录的文件信息
   #lsof +D /filepath/filepath2/

系统资源管理命令 -lsof

示例：

4. 查看指定目录的所有文件
   #lsof | grep etc

5. 列出指定用户打开的文件信息
   #lsof -u snow

6. 列出某个程序所打开的文件
   #lsof -c cron

系统资源管理命令 -lsof

示例：

7. 列出某个用户使用某个程序所打开的文件
   #lsof -u root -c cron

8. 列出除了某个用户外被打开的文件
   #lsof -u ^root

9. 列出某个 PID 所打开的文件
   #lsof -p 123

系统资源管理命令 -lsof

示例：

10. 列出多个 PID 所打开的文件
    #lsof -p 123,234,111

11. 列出除了某个 PID 外，其他 PID 所打开的文件
    #lsof -p ^123

12. 列出多个程序多打开的文件
    #lsof -c cron -c at

系统资源管理命令 -lsof

示例：

13. 列出某个用户组所打开的文件

```
#lsof -g  0
```

14. 显示哪个进程在使用指定 sudo 的可执行文件

```
#lsof `which sudo`
```

15. 显示哪个进程在使用光驱

```
#lsof /dev/cdrom
```

系统资源管理命令 -uptime

命令 :uptime

功能 : 显示系统不间断运行时间及 1,5,15 分钟负载

语法格式 :uptime [选项]

系统资源管理命令 -kill

命令 :kill

功能 : 对进程进行控制

语法结构 :

Kill [信号] PID

系统资源管理命令 -kill

特点：

1. kill 不仅仅可以杀掉进程，而且还可以对进程实现暂停，继续。

2. kill 的操作对象是 PID 。

3. kill 对进程的操作实际是对 PID 发送信号。

系统资源管理命令 -kill

查看信号：

命令：

#kill -l

总计 64 个

(1) 前 32 个为非实时信号（不可靠信号，即不支持队列，在使用时信号可能被丢失）

(2) 后 32 个为实时信号（可靠信号，支持队列）

系统资源管理命令 -kill

了解信号 (部分):

SIGHUP 1 一般程序收到此信号会退出，有些程序能够用此信号来 reload 。

SIGINT 2 键盘终端 (等于快捷键 ^c) 用于通知前台进程组终止进程。

SIGQUIT 3 类似 SIGINT, 但由 QUIT 字符 (等于快捷键 ^/) 来控制 , 进程在因收到 SIGQUIT 退出时会产生 core 文件，在这个意义上类似于一个程序错误信号。

系统资源管理命令 -kill

了解信号 (部分):

SIGKILL	9	强制终止进程。
SIGTERM	15	要求程序自己正常退出。这也是默认值。
SIGCONT	18	让已停止的进程继续执行。

系统资源管理命令 -kill

了解信号 (部分):

SIGSTOP　　19　　让正在执行的进程暂停。

SIGSTP　　20　　停止进程的运行 (可通过 susp 字符或 ^z 快捷键发送此信号

系统资源管理命令 -kill

示例：

1. 强制杀死 PID:3245

```
[root@dgtraing root]# kill  -9  3245
```

2. 对 PID:3266 发送 SIGTERM 信号

```
[root@dgtraing root]# kill  -SIGTERM 3266
```

系统资源管理命令 -killall

命令 :killall
功能 : 控制同名程序的所有进程

语法格式
killall [信号] 进程名

[root@dgtraing root]# killall bash

系统资源管理命令 - 前后台

前后台调度

1. 程序在后台运行 1

#commands & ←-& 符号为 commands 在后台运行

2. 程序在后台运行 2

#bg commands ←-bg 命令为 commands 在后台运行

系统资源管理命令 - 前后台

进程的前后台调度

3. 查看后台任务

```
#jobs
```

4. 根据 jobs 命令的后台任务序号调回前台

```
#fg  1  ←-fg 为前台命令
```

5. 如果程序被挂起 (^z), 也可以让任务恢复并在后台运行

```
#jobs
#bg 1
```

系统资源管理命令 - 前后台

后台进程关闭方法。

(1) jobs 查看工作号

(2) 调用 kill %jobs 队列序号，即可关闭进程。

如：

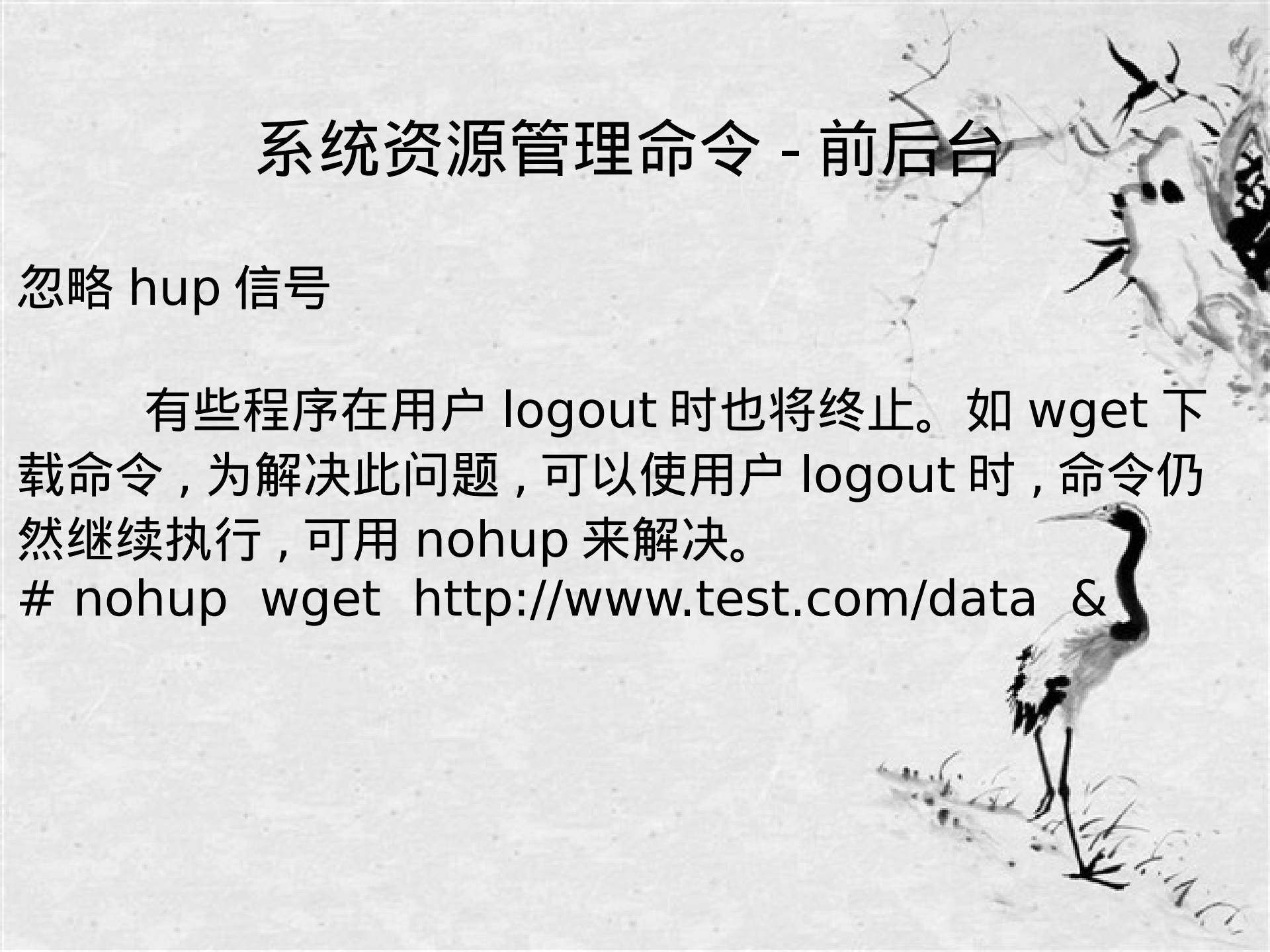
```
# jobs
   [1]+     Running          ./test.sh &
# kill  %1
```

系统资源管理命令 - 前后台

忽略 hup 信号

有些程序在用户 logout 时也将终止。如 wget 下载命令，为解决此问题，可以使用户 logout 时，命令仍然继续执行，可用 nohup 来解决。

```
# nohup  wget  http://www.test.com/data  &
```



系统资源管理命令 - 进程优先级

优先级

1. Linux 可以动态修改进程的优先级，以确保某个进程都可以得到更多 / 更少的运行资源。

2. 在 Linux 中进程的优先级从最低 19 到最高 -20

3. 优先级具有继承性。子进程会从父进程处获得优先级指数。

系统资源管理命令 - 进程优先级

命令 :nice
功能 : 在程序启动时直接赋予相关进程的优先级

语法格式 :
nice [优先级] 程序名

查看当前 shell 优先级

nice
0
0 表示当前 shell 的优先级。

系统资源管理命令 - 进程优先级

示例：

1. 对 append 脚本设置最低优先级

```
# nice  -19  ./append  &
```

2. 对 append 脚本设置最高优先级

```
[root@dgtraing root]# nice  --20  ./append  &
```

注意：最高的优先级指数是 -20 ，前面”-”为参数符

系统资源管理命令 - 进程优先级

命令 :renice

功能 : 对现有进程重新赋予优先级

语法格式 :

renice ［ 优先级 ］ PID

系统资源管理命令 - 进程优先级

示例：

1. 更改 PID:2794 优先级为最低优先级
renice 19 2794

2. 更改 PID:2794 优先级为最高优先级
renice -20 2794