Linux操作系统编程

常用UNIX/Linux命令
-进程控制类命令

- 文件目录类命令
- **进程控制类命令**
- 用户及权限管理类命令

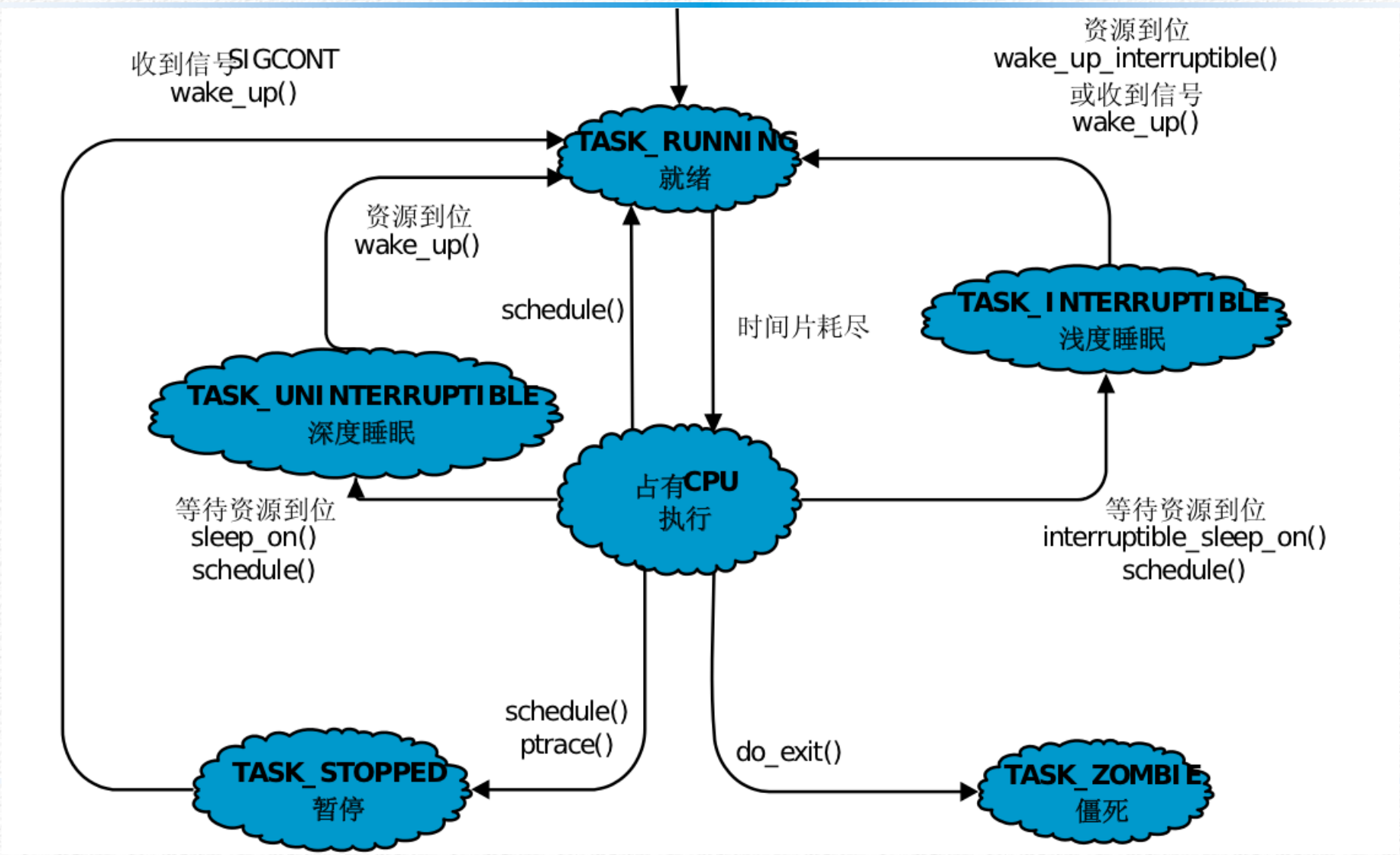
进程控制类命令

- 查看系统中的进程命令：ps　top
- 控制系统中的进程命令：kill　killall　nice　renice
- 进程后台运行命令 &
- 进程的挂起和恢复

程序和进程的区别

- **程序**是一个包含可执行代码的文件，它放在磁盘等介质上。
- 当程序被操作系统装载到内存并分配给它一定资源后，此时可称为**进程。**
- **程序**是**静态**概念，**进程**是**动态**概念。

Unix/Linux中的进程状态

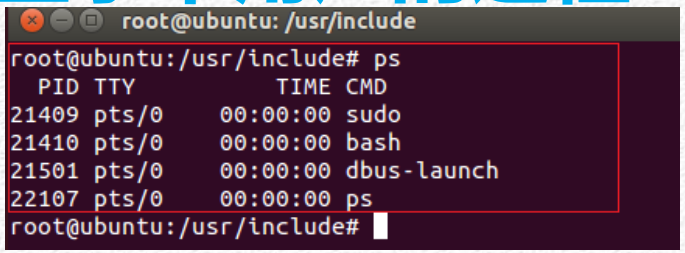


查看系统中的进程命令之ps命令

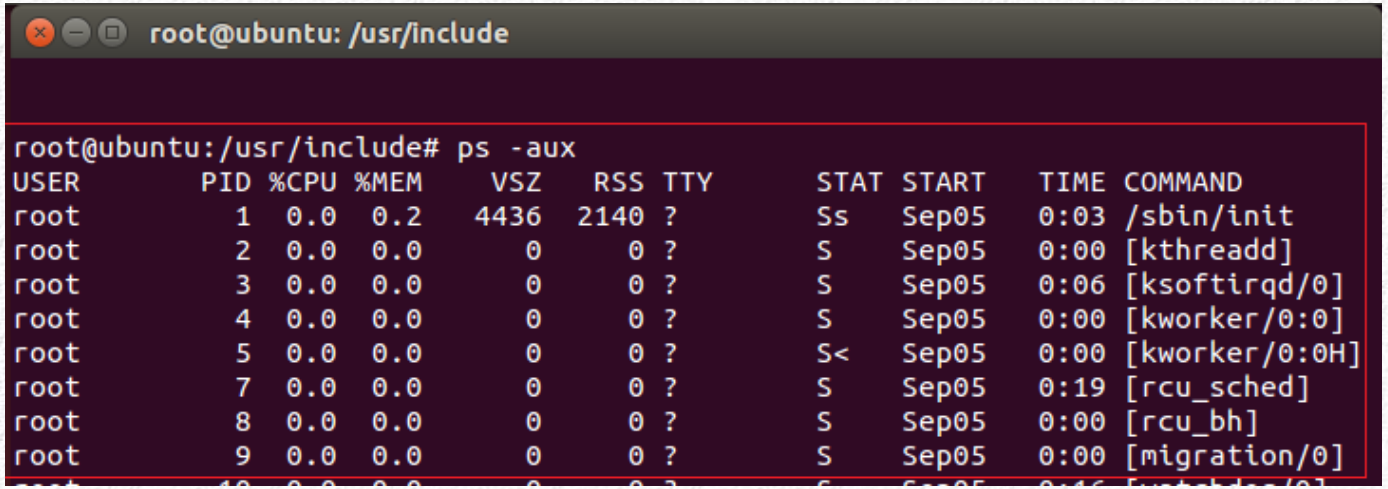
- 功能：ps命令是用来显示系统瞬间的进程信息，它可以显示出在用户输入ps命令时系统的进程及进程的相关信息。
- 格式：ps [参数]
  - l 长格式输出
  - u 按用户名和启动时间的顺序来显示进程
  - j 用任务格式来显示进程
  - f 用树形格式来显示进程
  - a 显示所有用户的所有进程（包括其它用户）
  - x 显示无控制终端的进程
  - r 显示运行中的进程

查看系统中的进程命令之ps命令

- 显示本用户的进程



- 查看系统和每位用户全部进程



[pp@host pp]$ ps –aux | grep pp （查找pp用户的进程）

监视系统任务的top命令

- **功能：**动态监视系统任务的工具，输出结果是连续的
- **格式：**top [参数]

-b 以批量模式运行，但不能接受命令行输入
-c 显示命令行，而不仅仅是命令名
-d N 显示两次刷新时间的间隔，比如 -d 5，表示两次刷新间隔为5秒
-i 禁止显示空闲进程或僵尸进程
-n NUM 显示更新次数，然后退出。比如 -n 5，表示top更新5次数据就退出
-p PID 仅监视指定进程的ID；PID是一个数值
-q 不经任何延时就刷新
-s 安全模式运行，禁用一些效互指令
-S 累积模式，输出每个进程的总的CPU时间

监视系统任务的top命令

```
  5:59am  up 2 min,  1 user,  load average: 0.32, 0.19, 0.07
33 processes: 32 sleeping, 1 running, 0 zombie, 0 stopped
CPU states:  0.0% user,  0.5% system,  0.0% nice, 99.4% idle
Mem:    126112K av,   36812K used,   89300K free,       0K shrd,    6996K buff
Swap:   265032K av,       0K used,  265032K free                   17876K cached

  PID USER     PRI  NI  SIZE  RSS SHARE STAT %CPU %MEM   TIME COMMAND
  616 root      15   0   428  428   380 S     0.1  0.3   0:00 gpm
  742 root      15   0  1012 1012   836 R     0.1  0.8   0:00 top
    1 root      15   0   480  480   428 S     0.0  0.3   0:04 init
    2 root      15   0     0    0     0 SW    0.0  0.0   0:00 keventd
    3 root      15   0     0    0     0 SW    0.0  0.0   0:00 kapmd
    4 root      34  19     0    0     0 SWN   0.0  0.0   0:00 ksoftirqd_CPU0
    5 root      15   0     0    0     0 SW    0.0  0.0   0:00 kswapd
    6 root      25   0     0    0     0 SW    0.0  0.0   0:00 bdflush
    7 root      15   0     0    0     0 SW    0.0  0.0   0:00 kupdated
    8 root      25   0     0    0     0 SW    0.0  0.0   0:00 mdrecoveryd
   16 root      15   0     0    0     0 SW    0.0  0.0   0:00 kjournald
  111 root      16   0     0    0     0 SW    0.0  0.0   0:00 kjournald
  112 root      15   0     0    0     0 SW    0.0  0.0   0:00 kjournald
  113 root      15   0     0    0     0 SW    0.0  0.0   0:00 kjournald
  114 root      15   0     0    0     0 SW    0.0  0.0   0:00 kjournald
  437 root      15   0   536  536   456 S     0.0  0.4   0:00 syslogd
  441 root      15   0   428  428   376 S     0.0  0.3   0:00 klogd
```

每5秒钟刷新一次，动态显示

按下U键：输入用户名　　查看用户进程

按下K键：输入PID　　删除进程

控制系统中的进程命令之kill

- 功能：该命令用于向某个进程（通过PID标识）传送一个信号，它通常与ps和jobs命令一起使用
- kill命令的格式是：kill –signal PID，常用的signal参数如下：
  - 1：SIGHUP，启动被终止的进程
  - 2：SIGINT，相当于输入ctrl+c，中断一个程序的进行
  - 9：SIGKILL，强制中断一个进程的进行
  - 15：SIGTERM，以正常的结束进程方式来终止进程
  - 17：SIGSTOP，相当于输入ctrl+z，暂停一个进程的进行
- 使用范例

  以正常的结束进程方式来终止第一个后台工作进程　　kill -SIGTERM %1

  重新启动进程ID为PID的进程　　kill -SIGHUP PID

killall命令

- killall命令使用进程的名称来杀死进程，使用此指令可以杀死一组同名进程
- 使用kill命令可以杀死指定进程PID的进程，如果要根据进程名称找到需要杀死的进程，还需要在之前使用ps等命令再配合grep来查找进程，而killall把这两个过程合二为一
- 用法：killall [参数] <正在运行的进程名>
  - -e：对长名称进行精确匹配； -I：忽略大小写的不同； -p：杀死进程所属的进程组； -i：交互式杀死进程，杀死进程前需要进行确认； -l：打印所有已知信号列表； -q：如果没有进程被杀死。则不输出任何信息； -r：使用正规表达式匹配要杀死的进程名称； -s：用指定的进程号代替默认信号“SIGTERM”； -u：杀死指定用户的进程。
- 使用范例：

```
[root@localhost test]# killall game
```

nice命令

- **功能：** nice 命令允许在默认优先级的基础上进行增大或减小的方式来运行命令
- **格式：** nice [参数] <command [arguments...]>
  - command 是系统中任意可执行文件的名称
  - -n, --adjustment 指定程序运行优先级的调整值
  - 优先级的调整值范围为-20 ~ 19，其中数值越小优先级越高，数值越大优先级越低
  - 若 nice命令未指定优先级的调整值，则以缺省值10来调整程序运行优先级，既在命令通常运行优先级基础之上增加10
- **使用范例：**

[root@host root]# nice -n -5 myprogram&

在后台以通常运行优先级-5的优先级运行myprogram

renice命令

- **功能：** 改变一个正在运行的进程的nice值
- **格式：** renice [参数] <pid>

  -n:指定程序运行优先级的调整值
- **使用范例：**

  [root@host root]# renice -5 777

  将正在运行的PID为777的进程nice值改为-5

在后台运行程序的＆命令

[root@host root]# cp –r /usr/* test &

将/usr 目录下的所有子目录及文件

复制到/root/test目录下的工作放到后台运行

进程的挂起和恢复

- 进程的中止（挂起）和终止
  - 挂起（Ctrl+Z）
  - 终止（Ctrl+C）
- 进程的恢复
  - 恢复到前台继续运行（fg）fg [n]
  - 恢复到后台继续运行（bg）bg [n]
- 查看被挂起的进程（jobs）