**实验三：进程管理（一）**

返回[目录](#_目录)

**实验目的：**

1. 学习进程的基本概念，理解进程的基本属性
2. 掌握进程操作的主要命令
3. 掌握通过/proc文件系统获取进程信息
4. 理解Linux系统的进程树

**实验时间**

6学时

**预备知识：**

1. 有关进程的主要命令：
   1. ps：查看系统运行进程

ps常用选项（最常用ps -aux）：

|  |  |
| --- | --- |
| 选项 | 说明 |
| x | 显示机器上的所有进程，不以终端来区分 |
| a | 显示终端上的所有进程，不以用户来区分 |
| w | 提供详细的宽范围输出 |
| u | 显示面向用户的格式 |
| f | 以进程树格式列出进程 |

ps运行结果的各字段含义：

|  |  |
| --- | --- |
| 字段 | 说明 |
| UID | 用户ID |
| PID | 进程ID |
| PPID | 父进程ID |
| TTY | 控制终端ID |
| PRI | 优先级编号（编号越低，分配给此进程的计算机时间越多） |
| NI(nice) | 影响动态优先级调整 |
| STAT | 当前的进程状态 |
| TIME | 使用的CPU时间 |
| COMMAND | 命令的名称 |

进程状态(STAT)可为以下状态之一：

|  |  |
| --- | --- |
| 状态代码 | 说明 |
| R（可运行） | 可运行的进程 |
| S（休眠） | 正在等待外部事件的进程（如数据到达）。 |
| D（不可中断的休眠） | 类似“S”（此时不能终止此进程） |
| T（已跟踪或已停止） | 进程已被暂停 |
| X （不用） | 进程已死 |
| Z（僵进程） | 进程已自行终止，但尚未释放 |

* 1. pstree：以树结构显示进程列表

pstree常用选项：

|  |  |
| --- | --- |
| 选项 | 说明 |
| -p | 显示进程PID |
| -u | 显示用户ID |

1.3 top: 实时显示正在运行的进程

top运行结果的各字段含义：

|  |  |
| --- | --- |
| 字段 | 说明 |
| PID | 进程ID |
| USER | 用户名 |
| PR | 优先级 |
| NI | Nice值 |
| VIRT | 进程使用的虚拟内存总量（单位为KB） |
| RES | 进程持有的物理内存总量（单位为KB） |
| SHR | 共享内存大小（单位为KB） |
| S | 进程状态 |
| %CPU | CPU使用率 |
| %MEM | 内存使用率(RES) |
| TIME | CPU时间 |
| COMMMAND | 命令名称/行 |

top常用命令：

|  |  |
| --- | --- |
| 字段 | 说明 |
| ？ | 帮助 |
| h | 帮助 |
| r | 将一个新的 nice 值分配给运行中的进程 |
| k | 向某个运行中的进程发送终止信号（与 kill 或 killall相同） |
| N | 按进程 ID 排序 |
| P | 按 CPU 负载排序 |

1.4 nice和renice

nice:为进程指派静态优先级

E.g. $nice –n +5 sleep 99

renice:更改运行进程的nice值

E.g. $renice 5 1012

注意： 1.调度程序使用nice级别来确定为运行的进程提供服务的频率。

2.nice的范围：-20~20；nice值越低，进城优先级越高。

3.nice默认为10。

4.只有root用户可以负的nice值启动进程；只有root用户可以降低运行进程的nice值。

1.5 前台与后台进程

$command & 以后台方式启动进程

fg 将进程切换到前台

bg 将进程切换到后台运行

jobs 列出作业控制的内容

ctrl+Z 挂起进程

1.6 kill和killall 向进程发送信号

kill 用PID向进程发送信号

E.g. $kill –SIGTERM 1703

killall 用进程的命令名称向进程发送信号

常用信号：

|  |  |  |
| --- | --- | --- |
| 编号 | 名称 | 说明 |
| 1 | SIGHUP | 重装载配置文件 |
| 2 | SIGINT | 通过键盘（Ctrl+C）中断 |
| 9 | SIGKILL | 停止进程。 |
| 15 | SIGTERM | 立即结束进程（以受控的方法终止进程，因此可以进行清理） |
| 18 | SIGCONT | 继续由STOP停止的进程 |
| 19 | STOP | 停止进程 |

注意：kill –l 或man 7 signal可获取信号的完整列表。

2. 利用/proc文件系统查看进程信息

系统中运行的每个进程在/proc目录下有个目录，目录名称为进程PID。

/proc/$PID/下包含进程的详细信息：

|  |  |
| --- | --- |
| 文件 | 说明 |
| cmdline | 进程的命令行 |
| environ | 进程的环境信息 |
| fd目录 | 为每个打开的文件描述符提供一个入口 |
| mem | 可通过其访问进程的内存映像 |
| stat | 包含进程的大多数信息 |
| status | 包含用户可读的进程信息 |
| cwd | 指向进程的当前工作目录 |
| exe | 指向正在被执行的文件 |
| maps | 内存映射区信息 |
| root | 进程的根目录 |
| statm | 进程对内存的使用情况 |

**实验内容:**

1. 列出你的系统中当前运行的所有进程，并解释bash进程的各项信息的含义（不包括RSS和SHR）。
2. 分别从至少三个虚拟终端登录，以树状形式列出你的系统中当前运行的所有进程及其PID。找出你当前运行进程的所有祖先进程，并说明其各自的作用。分析Linux系统中的进程的家族关系。
3. 试验有几种方法可以列出系统中的所有运行进程？
4. 执行下列命令和操作：

$sleep180

^Z

$jobs

$sleep 240&

$sleep 300&

$sleep 330&

$sleep 630&

$jobs

$fg 4

^C

$bg 1

$kill 5

解释命令的作用和执行结果。

1. 利用/proc文件系统，选择一个进程，如-bash，列出并解释/proc/$PID/cmdline, /proc/$PID/stat, /proc/$PID/status文件的内容。
2. 分别以root和普通用户身份运行一个进程，使用nice、renice和top设置或者改变其优先级，并验证是否成功。