**实验2 进程控制与描述**

## 管理Linux系统进程

(实验估计时间：90分钟)

### 背景知识

在本实验中，我们将使用Linux命令工作，来标识和控制系统进程。Linux操作系统使用进程管理作业，进程由系统或者用户初始化。用户在Linux环境下执行的作业大多数都会启动一个进程。进程可以启动或者派生一个子进程，这样创建了一个进程层次结构或者进程树，类似于具有父子关系的文件系统层次结构。

可以使用ps (进程状态) 命令工作，监控系统进程，使用kill命令终止不想要的进程。

### 实验目的

1) 回顾系统进程的概念，加深对Linux / UNIX进程管理的理解，理解进程的描述、进程的状态，掌握进程、线程与程序的关系。

2) 回顾ps命令和选项。

3) 列出当前shell中的进程。

4) 列出运行在系统中的所有进程。

5) 根据命令名搜索特定的进程。

6) 确定一个进程，终止它。

7) 使用kill命令终止进程。

8) 根据用户查找和终止进程。

9) 根据命令名终止进程。

实验目的：

### 工具/准备工作

在开始本实验之前，请回顾教科书的相关内容。

您需要做以下准备：

1) 由指导老师分配的登录用户 ID (如user2) 和口令。

2) 一台运行Red Hat Linux操作系统的计算机。

### 实验内容与步骤

**步骤1**：登录进入GNOME。

以普通用户身份登录Linux系统。

**步骤2**：访问命令行。

单击红帽子，在“GNOME帮助”菜单中单击“系统工具”-“终端”命令，打开“终端”窗口。

**步骤3**：回顾系统进程概念。

每个运行的程序都会创建一个进程，进程分配到一个唯一的进程标识符 (PID) 。PID被系统用于标识和跟踪进程，直到进程结束。操作系统内核管理所有进程的初始化和终止。每一个进程都要求系统资源 (例如CPU时间和RAM空间) 在其中工作。当进程启动的时候，操作系统把系统资源分配给每个进程，当进程终止的时候，系统回收这些资源。在Linux系统启动的时候，首先启动的两个进程是sched (调度) 和init (初始化) ，它们管理着其他进程。Linux系统中有几种不同类型的进程：

守护进程：由Linux内核启动的进程，为了特定目的而存在。例如，lpsched守护进程存在只是为了处理打印作业。

父进程：派生其他进程的进程是父进程。一个叫做init的守护进程是第一个调用的进程。每一个进程，除了init之外，都有一个父进程。

子进程：由其他进程派生出来的进程叫做子进程。

孤儿进程：在进程返回输出之前，它的父进程结束了，这样的进程叫做孤儿进程。

僵死进程：子进程没有带着输出返回给父进程。这样的进程在系统中变成丢失的进程。

使用此前的信息，完成以下填空：

1) Linux系统中，几乎每一个启动的进程，都会由内核分配一个唯一的\_\_\_\_\_\_\_PID\_\_\_\_进程标识符，用于跟踪从进程启动到进程结束。

2) 当启动新进程的时候，内核也给它们分配系统资源，如\_\_cpu时间\_和\_RAM空间\_\_\_。

3) 永远不向父进程返回输出的进程叫做\_\_僵死进程\_\_\_\_。

4) 由父进程派生出来的进程叫做\_\_\_\_\_\_子进程\_\_\_\_\_\_进程。

5) \_\_\_\_父进程\_\_\_\_\_\_\_进程是一个派生另一个进程的进程。

6) 运行用于提供服务的Linux系统进程是\_\_\_守护进程\_\_\_\_。

7) 如果父进程在子进程之前结束，它创建了一个\_\_孤儿进程\_\_\_\_进程。

**步骤4**：回顾ps命令和信息。

ps命令用于列出当前系统中运行的进程。

如果一个进程耗时太长，或者看上去好像已经终止了，表现为一个终端窗口不再响应或挂起，通常需要使用ps查看系统中的进程。通过列出进程，可以查看初始化进程的命令或程序的名字，以及所有它派生出来的子进程。通过多次执行ps命令，查看进程的时间——进程占用的CPU时间，可以检查一个进程是否还在运行。如果时间不再增长，那么进程可能已经终止了。如果进程占用太长的时间或者已经停止，可以使用ps命令检查进程的进程ID (PID) ，然后杀死进程。

ps命令的输出将显示PID号和与其相关的命令或程序。PID号通常用于终止一个进程。ps命令 ( ps [-options] ) 的3个主要选项如下表所示。

 ps 或ps -x 查看系统中属于自己的process

ps -au 查看系统中所有使用者的process

ps -aux 查看系统中包含系统内部及所有使用者的process

ps -aux|grep apache 找出系统中运行的所有名称中带有"apache"串的process

**查看正在background中执行的process**

执行格式： jobs

表1 ps命令选项

|  |  |  |
| --- | --- | --- |
| ps选项 | 意义 | 功能 |
| ps | 无选项 | 显示当前shell或终端窗口中，当前用户的进程信息 |
| ps –e | 每一个 | 显示系统中每一个进程的信息 |
| ps –f | 完全 | 产生一个完全列表，给出每个进程的所有可用信息 |
| ps -u userid | 用户 | 显示特定用户的所有进程 |

基本的ps命令显示在当前shell中的进程信息，用户只能够查看在这个终端窗口中初始化的进程。输入ps命令，将结果填入表2中。

表2 实验记录

|  |  |  |  |
| --- | --- | --- | --- |
| PID | TTY | TIME | CMD |
| 4641 | Pts/0 | 00:00:00 | bash |
| 4673 | Pts/0 | 00:00:00 | ps |
|  |  |  |  |

从当前终端窗口中，练习使用给出的每个选项的ps命令。

输入 ps -f 命令，显示运行在系统中的某个进程的完全信息，填入表3中。

表3 实验记录

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| UID | PID | PPID | C | STIME | TTY | TIME | CMD |
| Hyj | 4641 | 4639 | 0 | 09:07 | pts/0 | 00:00:00 | bash |
| Hyj | 4674 | 4641 | 0 | 09:12 | pts/0 | 00:00:00 | ps -f |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

ps -f命令输出的各栏标题见表4所示。

表4 ps -f栏标题

|  |  |
| --- | --- |
| 值 | 描述 |
| UID | 初始化进程的用户ID |
| PID | 进程的进程标识号。PID可以用来杀死进程 |
| PPID | 进程的父进程标识号 |
| C | 进程的优先级 |
| STIME | 进程的启动时间 |
| TTY | 控制进程终端的终端类型 |
| TIME | 进程使用的CPU时间的数量 |
| CMD | 命令名守护进程 (执行的程序的名字) |

**步骤5**：列出系统中运行的所有进程。

输入 ps -ef 命令，显示运行在系统中的各个进程的完全信息。执行该命令，并与ps –f命令的输出结果对照，一致吗？有何不同？

\_不一致ps –ef显示的信息内容要多于ps -f\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

注意分析当前终端窗口中的输出结果：

a. 显示了多少个进程？

62\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

b. 进程ID的PID是什么？

\_\_\_\_进程的进程标示号 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

c. 启动进程的命令 (CMD) 是什么？

\_\_\_\_ sched \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

d. 请观察，什么命令的PID号是1？

\_\_\_\_\_init\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

e. 执行ps –ef >aaa命令，将ps命令的输出送到文本文件aaa。再次运行cat aaa | wc命令，计算进程的数目。其中，cat是显示文本文件命令。“|”是管道命令，就是将前一个命令的输出作为后一个命令的输入。wc 命令用来计算文本的行数，第一个数字显示的是行的数目，可以用来计算进程的数目。计算出进程数目并做记录。

\_\_\_\_\_\_62个\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

执行下列命令把输出结果输入到wc命令中：

ps -ef | wc -l

第一个数字显示的是行的数目，也是进程的数目：

\_\_63\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

执行 man ps命令，可以打开Linux用户命令手册，了解ps命令的用法，输入wq命令可退出用户手册的阅读。man命令可以执行吗？结果如何？

\_\_\_\_可以执行；打开了Linux用户命令手册\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**步骤6**：根据命令名搜索特定的进程。

为了终止一个进程，必须找到该进程c的PID。大多数运行的系统中，有上百个进程在运行，ps -ef的列表可能很长。如果知道启动进程的执行程序的名字，可以更快地找到PID。

文件中字符串的查找 grep

执行格式： grep string file

例： grep abc file1 查找并列出串abc所在的整行文字



通过把ps命令的输出结果输入到grep中，可以搜索想要终止的特定进程，确定正确的PID。

grep命令可以在其他命令的输出中搜索所有类型的特征字符串。-l (长的输出) 选项将显示和找到的PID相关的进程的名字；-e选项显示PID和初始化命令的名字。

a. 单击红帽子，在“GNOME帮助”菜单中单击“游戏”-“堆麻将”命令，打开“堆麻将”游戏。

b. 在当前的终端窗口中，执行命令：

ps –e | grep mahjongg

(“堆麻将”游戏的进程名是“mahjongg”) ，查找和“堆麻将”游戏调度守护进程相关的所有进程。

c. 显示了多少个相关进程？

\_\_\_\_\_\_ 2个 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

d. 显示进程中最小的进程ID号是多少？

\_\_\_\_5076\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

c. 在当前终端窗口中执行命令：

pgrep –l mahjongg

查找所有和“堆麻将”游戏调度守护进程相关的进程。ps和pgrep命令输出之间有什么不同？

\_ ps命令显示PID、TTY、TIME、CMD ；pgrep命令只显示PID和CMD \_\_\_

**步骤7**：确定一个要终止的进程。

当试图终止一个程序，或者释放一个挂起的终端窗口的时候，杀死和不响应的应用相关的PID可能还是不够的，一般需要杀死那个进程的父进程，在少数情况下，甚至要杀死父进程的父进程。为了沿着层次结构，从子进程追溯到派生它们的父进程，必须能够查找PID和PPID。

必须首先确定最低级不响应进程的PID。通常，可以试着去杀死那个进程。如果这样做不能够停止那个进程，可能需要杀死它的父进程。杀死父进程将杀死所有它派生的子进程。杀死一个父进程也比杀死几个子进程快得多。

1) 在GNOME当前终端窗口中，输入ps命令。

有多少进程正在运行？为什么只有这么少的进程？

\_有2个进程；因为ps只显示当前运行的进程

2) 正在运行的进程的名字是什么？它表示什么？

bash和ps \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3) 这个进程的PID是多少？

\_5083和5119\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4) 在bash shell中输入命令csh，打开一个C shell会话，当前系统的会话提示符是什么？

\_~\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5) 输入 ps -f 命令，显示在当前shell中运行的进程的所有信息。现在什么进程正在运行？

\_ bash、-sh、ps –f \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6) bash的PID是-sh (csh) 的PPID吗？

\_\_\_是，都为5083\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

7) sleep-----执行另一个命令之前等待的时间

sleep amount [units]

units默认为秒(s)，m表示分钟，h表示小时，d表示天。

输入命令sleep l000&，创建一个进程，把执行挂起1000秒 (大约15分钟) 。&表示在后台运行命令，返回shell提示符，这样您可以继续工作。

8) 再次输入ps -f命令。-sh (csh) 的PID是sleep命令的PPID吗？哪个PID是csh (C shell) 的子进程？

\_\_是，都为5122；5146和5147是csh的子进程\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

9) 输入命令 kill -9 <csh的PID> 退出C shell。然后再次键入ps -f命令。什么PID是sleep父命令的PPID？

\_\_\_init进程的PID： 1 是sleep父命令的PPID\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

sleep进程现在是什么类型的进程？

\_\_\_\_孤儿进程\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**步骤8**：使用kill命令终止一个进程。

信号用于终止、挂起和继续进程。有时候使用Ctrl + c可以终止一个不响应的进程。这样向进程发送一个中断 (INT) 信号，终止它及所有它派生的子进程。

kill命令提供了一个终止不想要的命令进程的直接方法。当想停止一个运行了很长时间的命令的时候，或者当需要终止一个不能够正常退出的进程的时候，它很有用。指定PID正常地杀死一个进程。

kill [-signal] process -id

为了使用kill命令终止一个进程，应该首先键入ps找出进程的PID，然后键入kill紧跟着PID。如果使用kill而没有指定信号，则信号15 (SIGTERM) 将被发送给特定PID号的进程。这叫做软杀死，通常终止了进程。如果可能，最好软杀死进程，这样它可以正确地关闭文件，温和地终止进程。

如果需要强制终止一个进程，可以使用带 -9选项的kill命令。这个选项叫做强制杀死，当shell对所有其他终止信号都不响应的时候，需要使用它来杀死shell。

kill -9 process -id

注意：对于shell之外的进程，把kill –9 (SIGKILL) 作为最后的选项，因为它是一种生硬的方法，不允许正确的进程终止。

a. 输入命令ps –f ，显示在当前shell中运行的进程的完整信息。现在什么进程正在运行？

bash、sleep 1000、ps -f \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

b. 因为sleep进程现在是一个孤儿进程，已经被init进程 (PID #l) 收养，输入命令，对sleep的PID执行软杀死。如果sleep已经终止 (超过15分钟) ，则重复步骤7中的8) 、9) 操作，然后软杀死sleep程序。再次输入ps -f命令。现在什么进程正在运行？

\_ bash、ps -f \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**步骤9**：按照用户查找和终止一个进程。fdsf

ps命令可以带 -u (用户) 选项使用，查找指定用户的进程。这个命令比ps-ef使用得更多，因为一般用户只能够管理属于自己的进程。我们可以通过用户登录名或者UID号来查找用户的进程。用户只能够终止自己的进程，但是超级用户可以终止系统中所有运行的进程。

ps –u login-ID或UID

1) 启动附件菜单中的任意一个应用程序（例如计算器应用程序）。

2) 使用带 -u选项的ps命令，查找以登录ID运行的所有进程。计算器应用程序的PID是什么？

5626

3) 使用软杀死终止计算器应用程序。看一看它是否还在桌面上运行？

计算器应用程序不再，但是堆麻将还在

4) 使用ps -u命令组合grep，查找“堆麻将”程序的PID号，而不是使用ps -u查找整个列表。您使用什么命令？

提示：猜猜“堆麻将”程序的命令名中可能有哪个字母，使用grep搜索那些字母。

ps –u student| grep mahj

5) 终止“堆麻将”程序。它是否还在桌面上运行？

否

6) 退出 C shell。

**步骤10**：删除所有在本实验中创建的文件和目录。

**步骤11**：关闭终端窗口，注销。

**附:其它命令**

（1）后台（background）执行process command的命令

执行格式： command & （在命令后加上 &）

例： gcc file1 & 在后台编译file1.c

注意：按下^Z，暂停正在执行的process。键入”bg”，将所暂停的process置入background中继续执行。

例： gcc file1 &

^Z

stopped

bg

（2）结束或终止在background中的进程 kill

执行格式： kill %n

例： kill %1 终止在background中的第一个job

kill %2 终止在background中的第二个job

（3）显示系统中程序的执行状态

例: top -q 不断地更新、显示系统程序的执行状态

第一行显示的项目依次为当前时间、系统启动时间、当前系统登录用户数目、平均负载。

第二行为进程情况，依次为进程总数、休眠进程数、运行进程数、僵死进程数、终止进程数。

第三行为CPU状态，依次为用户占用、系统占用、优先进程占用、闲置进程占用。

第四行为内存状态，依次为平均可用内存、已用内存、空闲内存、共享内存、缓存使用内存。

第五行为交换状态，依次为平均可用交换容量、已用容量、闲置容量、高速缓存容量。

PID 每个进程的ID。

PPID 每个进程的父进程ID。

UID 每个进程所有者的UID 。

USER 每个进程所有者的用户名。

PRI 每个进程的优先级别。

NI 该进程的优先级值。

SIZE 该进程的代码大小加上数据大小再加上堆栈空间大小的总数。单位是KB。

TSIZE 该进程的代码大小。对于内核进程这是一个很奇怪的值。

DSIZE 数据和堆栈的大小。

TRS 文本驻留大小。

D 被标记为“不干净”的页项目。

LIB 使用的库页的大小。对于ELF进程没有作用。

RSS 该进程占用的物理内存的总数量，单位是KB。

SHARE 该进程使用共享内存的数量。

STAT 该进程的状态。其中S代表休眠状态；D代表不可中断的休眠状态；R代表运行状态；Z代表僵死状态；T代表停止或跟踪状态。

TIME 该进程自启动以来所占用的总CPU时间。如果进入的是累计模式，那么该时间还包括这个进程子进程所占用的时间。且标题会变成CTIME。

%CPU 该进程自最近一次刷新以来所占用的CPU时间和总时间的百分比。

%MEM 该进程占用的物理内存占总内存的百分比。

COMMAND 该进程的命令名称，如果一行显示不下，则会进行截取。内存中的进程会有一个完整的命令行

按"ctrl+c"停止查看

（4）以树状图显示执行的程序 pstree

例: pstree -h 列出进程树并高亮标出当前执行的程序

(5)查看系统的当前进程信息

* + top

