**Липецкий государственный технический университет**

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №3

по дисциплине «ДПО Интаро»

Управление процессами в Linux

|  |  |  |
| --- | --- | --- |
| Студент |  | Поленников А.А. |
| Группа АИ-17 |  | . |
| Руководитель |  |  |
|  |  | Кургасов В.В. |

Липецк 2019 г.

Цель работы

Ознакомится с средствами управления процессами в операционной системе Linux.

Остановка и возобновление процесса

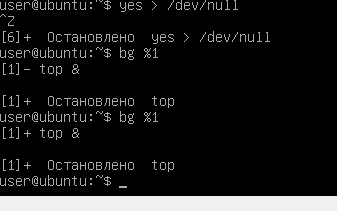


Рисунок 1 – остановка и возобновление, перевод из фонового на передний план

Для того чтобы перевести выполнение этого процесса в фоновый режим, необходимо выполнить следующую команду: bg

Причем необязательно делать это сразу после остановки процесса, главное правильно указать номер остановленного процесса.

Для того чтобы вернуть процесс из фонового режима выполнения на передний план, достаточно выполнить следующую команду: fg

Завершение работы процесса



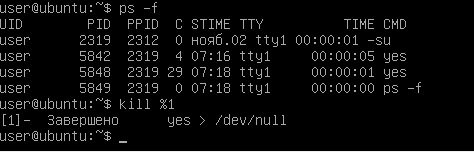
Рисунок 2 – завершение процесса

Рисунок 3 – завершение процесса с помощью kill

Команда **kill** может получать в качестве аргумента, как номер процесса, так и идентификационный номер (PID) процесса

С помощью команды **killall** можно прекратить выполнение нескольких процессов сразу, имеющих одно и то же имя. Например, команда killall mc прекратит работу всех программ **mc**, запущенных от имени данного пользователя.

Программы, используемые для управления процессами

nohup

Эта утилита позволяет организовать фоновый процесс, продолжающий свою работу даже тогда, когда пользователь отключился от терминала, в отличие от команды &, которая этого не позволяет. Для организации такого фонового процесса необходимо выполнить команду в форме: nohup выполняемая\_фоновая\_команда &

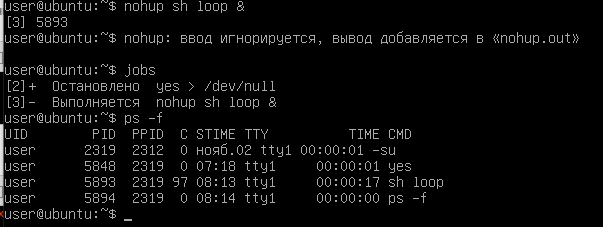


Рисунок 4 – утилита nohup

ps

Программа предназначена для получения информации о существующих в операционной системе процессах.

Опции команды ps

а Показать процессы всех пользователей

с Имя команды из переменной среды

е Показать окружение

f Показать процессы и подпроцессы

h Вывод без заголовка

j Формат заданий l "Длинный" формат вывода

m Вывод информации о памяти

n Числовой вывод информации

r Только работающие процессы

s Формат сигналов

S Добавить время использования процессора порожденными процессами

txx Только процессы, связанные с терминалом хх

u Формат вывода с указанием пользователя

v Формат виртуальной памяти

w Вывод без обрезки информации для размещения в одной строке

x Показать процессы без контролирующего терминала

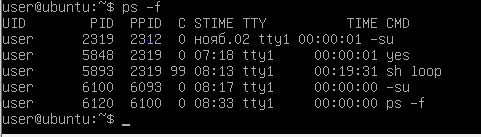


Рисунок 5 – команда ps

top

Еще одна утилита, с помощью которой можно получать информацию о запущенных в операционной системе процессах.

Утилита top после запуска периодически обновляет информацию о состоянии процессов в операционной системе, что позволяет нам динамически получать информацию о загрузке системы.

Утилита top полностью управляется с клавиатуры. Вы можете получить справку, нажав клавишу h.

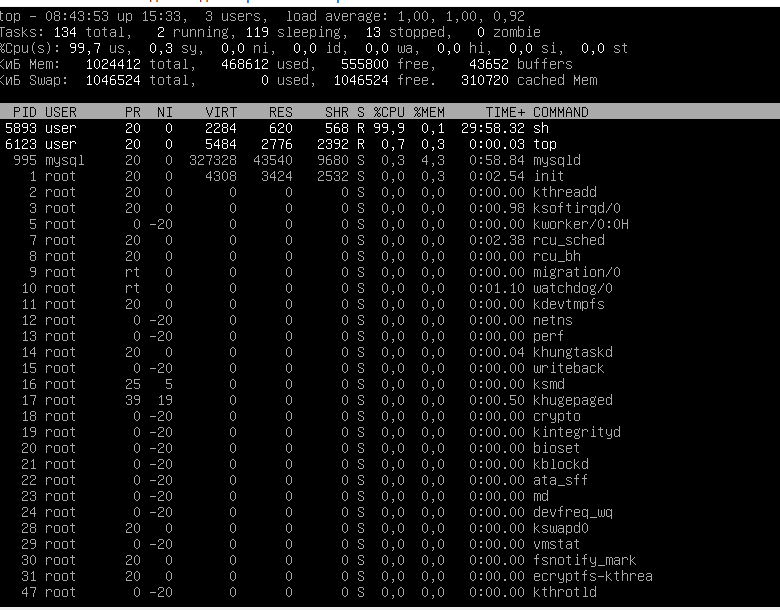


Рисунок 6 – команда top

kill

Программа kill (в переводе с английского — убить) предназначена для посылки соответствующих сигналов указанному нами процессу. Как правило, это бывает тогда, когда некоторые процессы начинают вести себя неадекватно. Наиболее часто программа применяется, чтобы прекратить выполнение процессов.

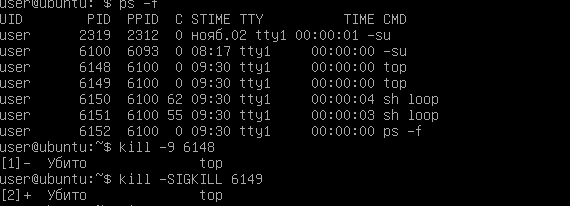


Рисунок 7 – команда kill

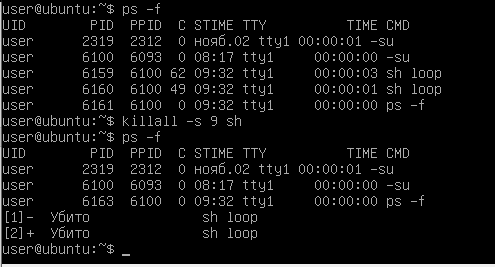


Рисунок 8 – команда killall

nice

В операционной системе Linux у каждого процесса есть свой приоритет исполнения. Поскольку операционная система многозадачная, то для выполнения каждого процесса выделяется определенное количество времени в соответствии с приоритетом.

Программа nice позволяет запустить команду с предопределенным приоритетом выполнения, который задается в командной строке. C помощью утилиты nice можно понизить приоритет какой-либо задачи, таким образом, предоставляя другим процессам больше процессорного времени. Повысить приоритет той или иной задачи имеет право только пользователь root. Синтаксис использования nice следующий: nice -number command



Рисунок 9 – использвание nice



Рисунок 10 – просмотр процесса в top

renice

Программа renice, в отличие от программы nice, позволяет изменить приоритет уже работающего процесса. Формат запуска программы следующий: renice -number PID

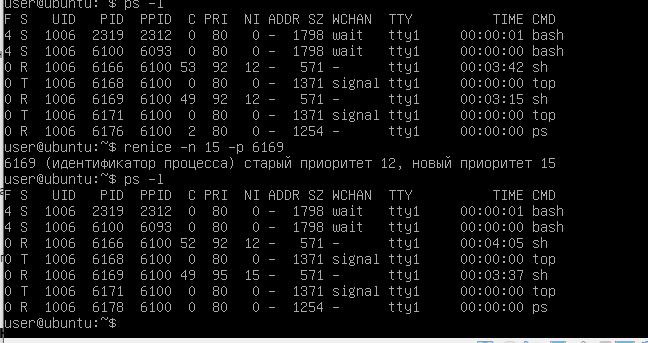


Рисунок 11 – процесс renice

at

Одна из основных задач автоматизации администрирования операционной системы — выполнение программ в заданное время или с заданной периодичностью.

Для запуска одной или более команд в заранее определенное время используется команда at. В этой команде вы можете определить время и дату запуска той или иной команды. Команда at требует, по меньшей мере, двух параметров — время выполнения программы и запускаемую программу с ее параметрами запуска.

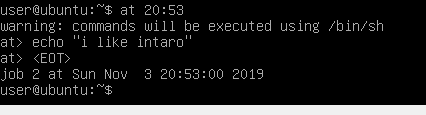


Рисунок 12 – команда at

batch

Команда batch в принципе аналогична команде at. Более того, batch представляет собой псевдоним команды at -b. Команда batch — ее использование позволяет операционной системе самой решить, когда наступает подходящий момент для запуска задачи в то время, когда система не сильно загружена.

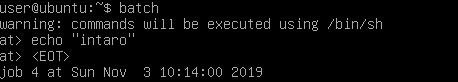


Рисунок 13 – команда batch

cron и crontab

сron — это программа, выполняющая задания по расписанию, но, в отличие от команды at, она позволяет выполнять задания неоднократно. Вы определяете времена и даты, когда должна запускаться та или иная программа. Времена и даты могут определяться в минутах, часах, днях месяца, месяцах года и днях недели.

Программа cron запускается один раз при загрузке системы. При запуске cron проверяет очередь заданий at и задания пользователей в файлах crontab. Если для запуска не было найдено заданий — следующую проверку cron произведет через минуту.

Для создания списка задач для программы cron используется команда crontab. Для каждого пользователя с помощью этой команды создается его собственный crontab-файл со списком заданий, имеющий то же имя, что и имя пользователя.

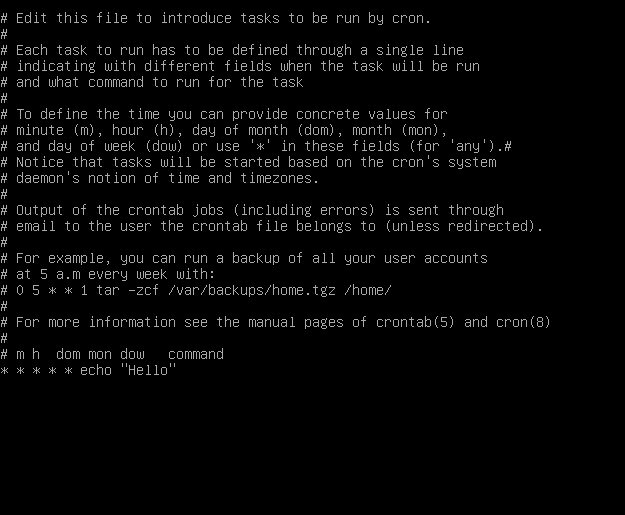


Рисунок 14 – команда crontab

Общение между процессами

Процессы могут общаться между собой при помощи каналов, сокетов, разделяемой памяти и т. д. Рассмотрим только один вариант: каналы.

Именованные каналы (FIFO: First In First Out). Данный вид канала создаётся с помощью mknod или mkfifo, и два различных процесса могут обратиться к нему по имени.



Рисунок 14 - первый терминал

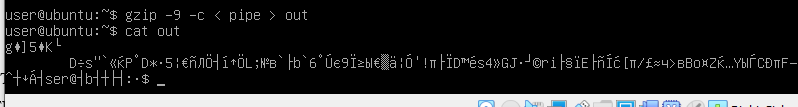


Рисунок 15 – второй терминал

Был создан канал pipe по которому передавались данные между терминалами.

Вывод

В ходе лабораторной работы мы ознакомились с средствами управления процессами в операционной системе Linux.