**Липецкий государственный технический университет**

Факультет автоматизации и информатики

Кафедра Автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №3

по курсу "Операционная система Linux"

Управление процессами в Linux

Курбатова К.А.

|  |
| --- |
| Студент |
| Группа АИ-17-1 | Кургасов В.В. |  |
| Руководитель  Доцент |  |  |

Липецк 2019г.

Цель работы

Ознакомиться на практике с понятием процесса в операционной системе. Приобрести опыт и навыки управления процессами в операционной системе Linux.

Ход работы

Выполнение процесса на переднем плане и в фоновом режиме.

Для выполнения программы в режиме переднего плана достаточно просто набрать имя программы в командной строке и запустить ее на выполнение.

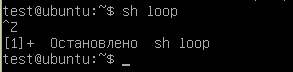


Рисунок 1 – Выполнение программы в режиме переднего плана.

Для запуска программы в качестве фонового процесса достаточно набрать в командной строке имя программы и в конце добавить знак амперсанта (&)

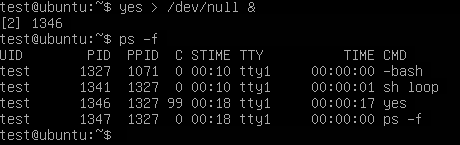


Рисунок 2 – Команда для запуска программы yes в фоновом режиме с подавлением вывода.

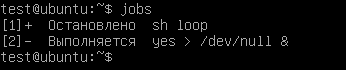


Рисунок 3 – Проверка состояния фоновых процессов с помощью команды jobs.

Остановка и возобновление процесса.

Существует еще один способ перевести процесс в фоновый режим. Для этого мы должны выполнить следующие действия:

1. Запустить процесс выполняться на переднем плане.
2. Приостановить выполнение процесса.
3. Продолжить процесс в фоновом режиме.

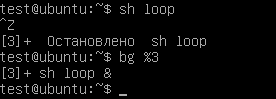


Рисунок 4 – Реализация алгоритма остановки и возобновления процесса.

Завершение работы процесса.

1. Если процесс интерактивный, как правило, в документации или прямо на экране написано, как корректно завершить программу.



Рисунок 5 – Запуск процесса.

2. В том случае, если вы не знаете, как завершить текущий процесс (не фоновый), можно воспользоваться клавиатурной комбинацией <Ctr> + <Break>. Попробуйте также комбинацию клавиш <Ctr> + <C>.



Рисунок 6 – Реализация второго варианта.

3.В том случае, если вам не удалось прекратить выполнение процесса вышеприведенными способами – например, программа зависла – для завершения процесса можно воспользоваться следующими командами: kill, killall.



Рисунок 7 – Реализация третьего варианта завершения процесса командой kill.

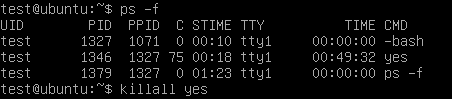
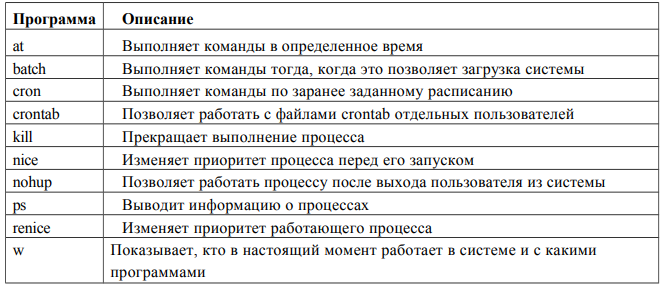


Рисунок 8 – Реализация третьего варианта завершения процесса командой killall.

Программы, используемые для управления процессами.

Таблица 1 – Программы управления процессами



Во вновь запущенном терминале процесс нельзя увидеть с помощью команды jobs, так как эта команда выводит список процессов текущего терминала, поэтому после подключения к терминалу необходимо использовать команду ps с ключом –A.

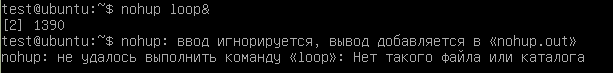


Рисунок 9 – Реализация команды nohup.

Простой запуск ps без параметров выдаст список программ, выполняемых на текущем терминале. Обычно этот список очень мал.

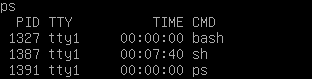


Рисунок 10 – Команда ps.

Первый столбец — PID (идентификационный номер процесса). Как уже упоминалось, каждый выполняющийся процесс в системе получает уникальный идентификатор, с помощью которого производится управление процессом. Каждому вновь запускаемому на выполнение процессу присваивается следующий свободный PID. Когда процесс завершается, его номер освобождается. Когда достигнут максимальный PID, следующий свободный номер будет взят из наименьшего освобожденного.

Следующий столбец — TTY — показывает, на каком терминале процесс выполняется.

Столбец TIME показывает, сколько процессорного времени выполняется процесс. Оно не является фактическим временем с момента запуска процесса, поскольку Linux — это многозадачная операционная система. Показывается время, реально потраченное процессором на выполнение процесса.

Столбец CMD показывает имя программы, опции командной строки не выводятся.

Для получения расширенного списка процессов, выполняемых в системе, используется следующая команда:



Рисунок 11 – команда ps –ax для расширенного списка.

Появился новый столбец — STAT, в котором отображается состояние (status) процесса.

Если вы хотите увидеть еще больше информации о выполняемых процессах, попробуйте выполнить команду:



Рисунок 12 – команда ps –aux для более расширенного списка.

Появились еще столбцы:

• USER — показывает, от имени какого пользователя был запущен данный процесс;

• %CPU, %MEM — показывают, сколько данный процесс занимает соответственно процессорного времени и объем используемой оперативной памяти;

• TIME — время запуска программы.

Команда top, с помощью которой можно получать информацию о запущенных в операционной системе процессах. Для использования достаточно просто запустить команду top на выполнение. Эта утилита выводит на экран список процессов в системе, отсортированных в порядке убывания значений использованного процессорного времени.

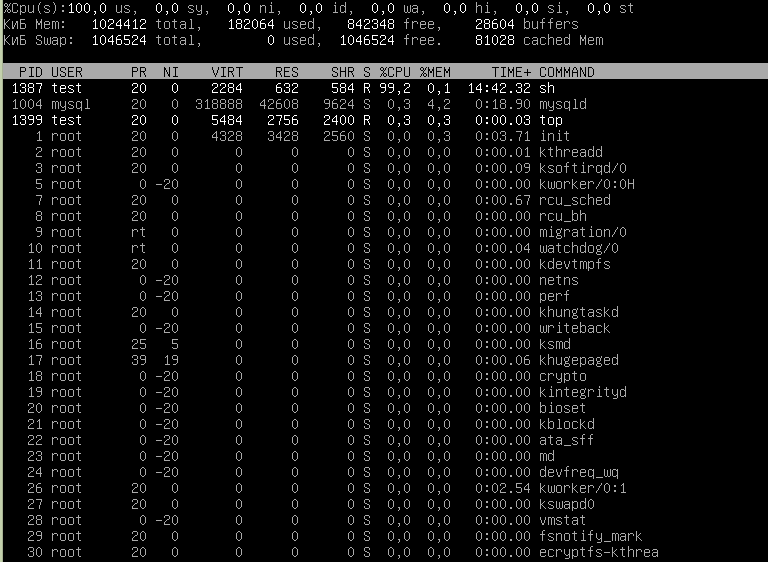


Рисунок 13 – Результат выполнения команды top.

Утилита top полностью управляется с клавиатуры. Вы можете получить справку, нажав клавишу h. Вот еще несколько полезных команд:

k — используется для отправки сигнала процессу;

u — используется для вывода процессов указанного пользователя;

i — используется для вывода только работающих процессов;

r — используется для изменения приоритета выбранного процесса.



Рисунок 14 – Убит процесс loop.



Рисунок 15- Полный список сигналов kill.

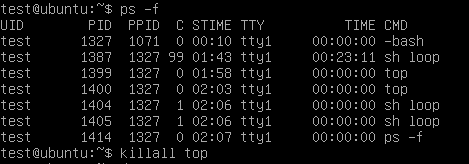


Рисунок 16 – killall удаляет процессы с одинаковым именем.

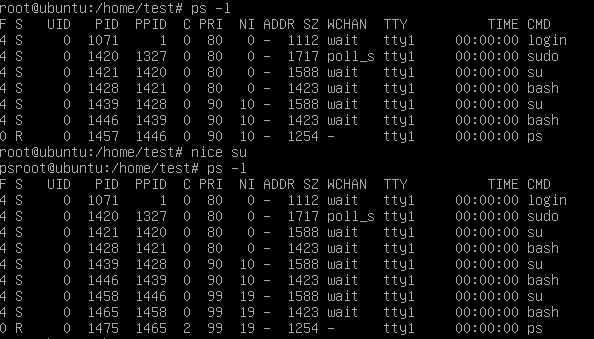


Рисунок 17 – повышение приоритета процесса пользователем root.

Программа renice, в отличие от программы nice, позволяет изменить приоритет уже работающего процесса.

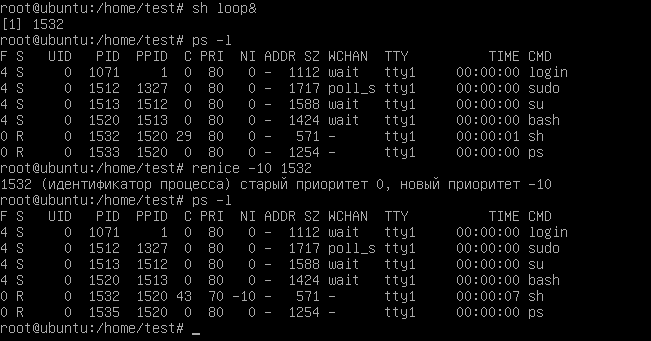


Рисунок 18 – Понижение приоритета процесса sh с помощью команды renice.

Для запуска одной или более команд в заранее определенное время использует ся 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.

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.

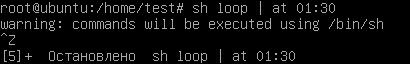


Рисунок 19 – реализация команды at.

Команда batch в принципе аналогична команде at. Более того, batch представляет собой псевдоним команды at -b. Для чего необходима эта команда? Представьте, вы хотите запустить резервное копирование вечером. Однако в это время система очень занята, и выполнение резервирования системы практически парализует ее работу. Для этого и существует команда batch — ее использование позволяет операционной системе самой решить, когда наступает подходящий момент для запуска задачи в то время, когда система не сильно загружена.

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.

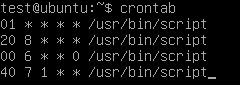


Рисунок 20 – Реализация команды crontab

Именованный канал или именованный конвейер — один из методов меж процессного взаимодействия, расширение поня-тия конвейера в UNIX и подобных ОС. Именованный канал позволяет различным процессам обмениваться данными, даже если программы, выполняющиеся в этих процессах, изначально не были написаны для взаимодействия с другими программами.



Рисунок 21 - Создание канала pipe, который принимает со второй консоли информацию и архивирует её gzip-ом.



Рисунок 22 - Вывод информации в канал pipe со второго терминала.

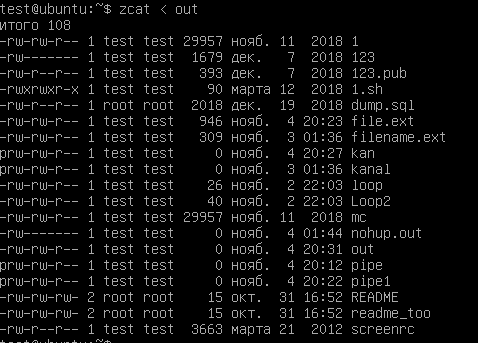


Рисунок 23 - Просмотр файла out, с помощью zcat, который читает сжатые файлы.

Вывод

Я ознакомился на практике с понятием процесса в операционной системе. Приобрел опыт и навыки управления процессами в операционной системе Linux.