**Липецкий государственный технический университет**

Факультет автоматизации и информатики

Кафедра Автоматизированных систем управления

Отчет по лабораторной работе № 3

«Управление процессами ОС Ubuntu»

по курсу «ОС Linux»

Грунау Г. Ю.

Студент

Группа АИ-18

Кургасов В. В.

Руководитель

Липецк 2020 г.

Содержание

[Выполнение работы 3](#_Toc56458413)

[1. Повторить команды cat, head, tail, more, less, grep, find 3](#_Toc56458414)

[2. Разобраться с понятиями конвейер, перенаправление ввода-вывода. 6](#_Toc56458415)

[3. Ознакомиться с информацией из рекомендованных источников и других про конвейеризации 8](#_Toc56458416)

[4. Повторить назначение прав доступа. Команды chmod, chown 9](#_Toc56458417)

[5. Ознакомиться с информацией по теме процессы, посмотреть и опробовать примеры наиболее распространенных команд, изучить возможность запуска процессов в supervisor 11](#_Toc56458418)

[6. Запуск процессов по расписанию. 18](#_Toc56458419)

[Вывод 19](#_Toc56458420)

# Выполнение работы

1. Повторить команды cat, head, tail, more, less, grep, find

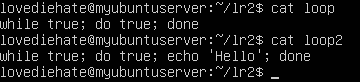


Рисунок - Использование команды cat

cat — одна из наиболее часто используемых команд в Linux. Она считывает данные из файлов и выводит их содержимое. Это самый простой способ отображения содержимого файла в командной строке.

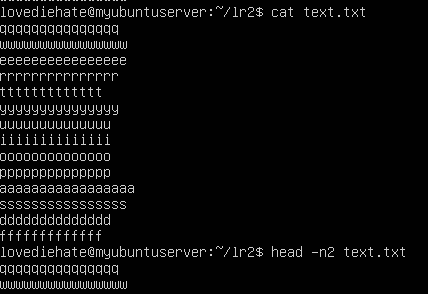


Рисунок - Использование команды head



Рисунок - Использование команды tail

Введена команда ps -ef | more

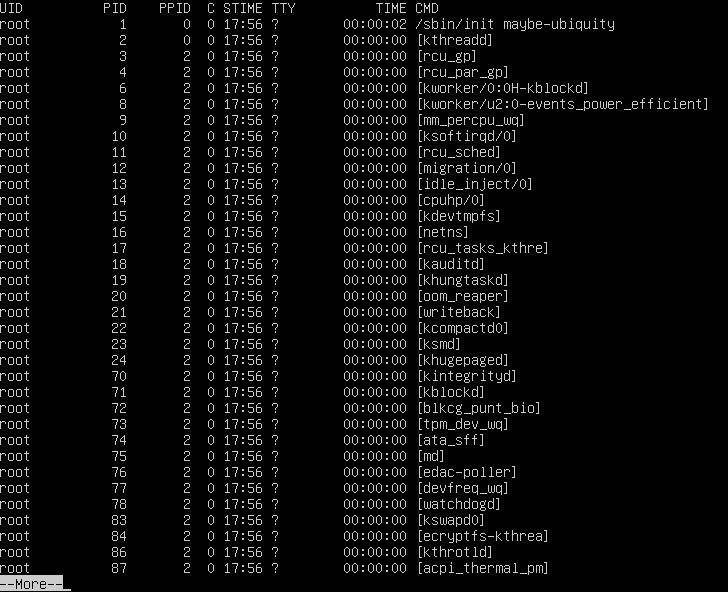


Рисунок - Использование more

Команда more позволяет выводить изображение в терминале на одной странице одновременно

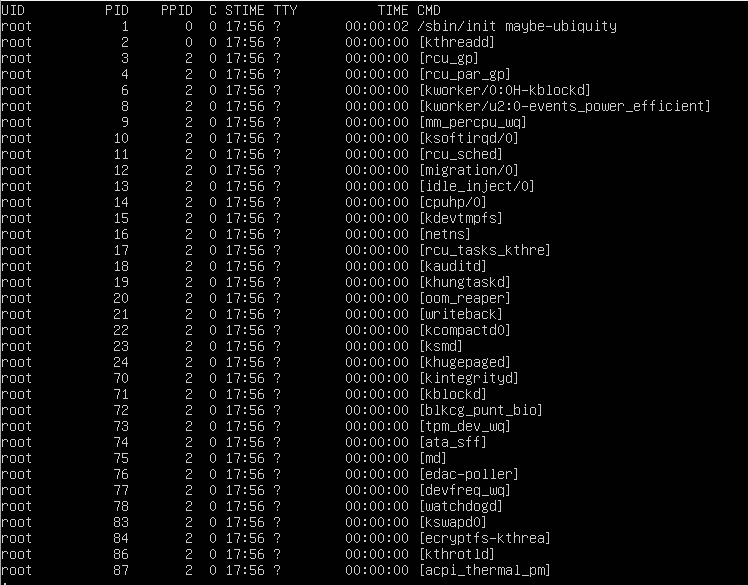


Рисунок - Команда less

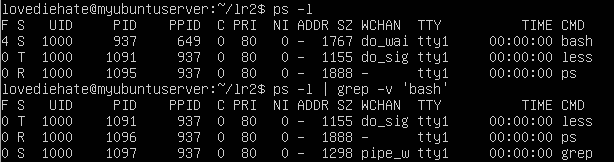
less – существенно более развитая команда для пролистывания текста. При чтении данных со стандартного ввода она создает буфер, который позволяет листать текст как вперед, так и назад, а также искать как по направлению к концу, так и по направлению к началу текста

Рисунок - Команда grep

Grep это утилита командной строки Linux, который даёт пользователям возможность вести поиск строки

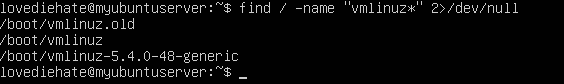


Рисунок - Команда find

Команда find представляет собой универсальный инструмент поиска: она позволяет искать файлы и каталоги, просматривать все каталоги в системе или только текущий каталог.

1. Разобраться с понятиями конвейер, перенаправление ввода-вывода.

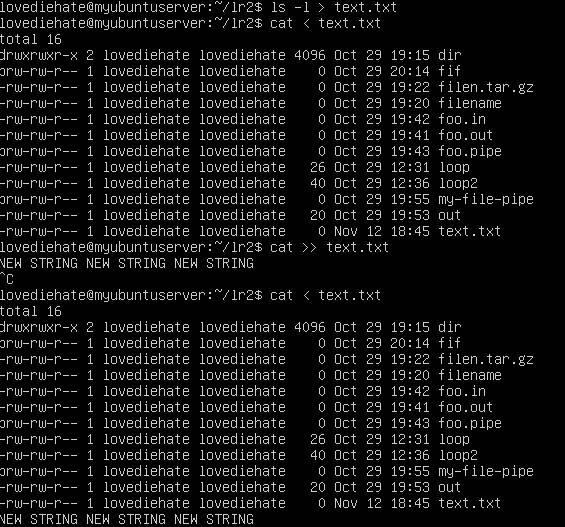


Рисунок - Перенаправление in out

Символ «|» на Рисунке 6 — это и есть конвейер. Его можно понимать как канал, в который один процесс может только писать, а другой — только читать из него. Выборка и помещение информации в такой канал происходит в порядке FIFO.

В работе с командной строкой Linux есть понятия стандартных устройств ввода, вывода и вывода ошибок.

stdin – стандартное устройство ввода. Имеет файловый указатель №0. Автоматически открывается всеми процессами.

stdout – стандартное устройство вывода. Имеет файловый указатель №1. Автоматически открывается всеми процессами.

stderr – стандартный поток ошибок (специальное устройство вывода для сообщений об ошибках. Имеет файловый указатель №2. Автоматически открывается всеми процессами.

По умолчанию практически все команды Linux используют для ввода информации stdin , а для вывода stdout и stderr,если их параметрами не указано обратное.

Операторы перенаправления способны изменять направление вывода и ввода информации. Так оператор:

> - перенаправляет стандартный поток в файл (другой поток). При этом если файл существует, то он перезаписывается, если не существует – создается.

>> - перенаправляет стандартный поток в файл. При этом если файл существует, то информация добавляется в конец, если не существует – файл создается.

< - перенаправляет содержимое указанного файла на стандартный ввод программы.

>& - перенаправляет стандартные потоки вывода и ошибок друг в друга.

1. Ознакомиться с информацией из рекомендованных источников и других про конвейеризации

Конвейеры — это возможность нескольких программ работать совместно, когда выход одной программы непосредственно идет на вход другой без использования промежуточных временных файлов. Синтаксис: команда1 | команда2, выполняет команду1 используя её поток вывода как поток ввода при выполнении команды2, что равносильно использованию двух перенаправлений и временного файла:

команда1 > ВременныйФайл

команда2 < ВременныйФайл

rm ВременныйФайл

Хороший пример командных конвейеров — это объединение echo с другой командой для получения интерактивности в неинтерактивных средах, к примеру:

echo -e "ИмяПользователя\nПароль" | ftp localhost

Конвейер (англ. pipeline) в терминологии операционных систем семейства Unix — некоторое множество процессов, для которых выполнено следующее перенаправление ввода-вывода: то, что выводит на поток стандартного вывода предыдущий процесс, попадает в поток стандартного ввода следующего процесса. Запуск конвейера реализован с помощью системного вызова pipe().

1. Повторить назначение прав доступа. Команды chmod, chown

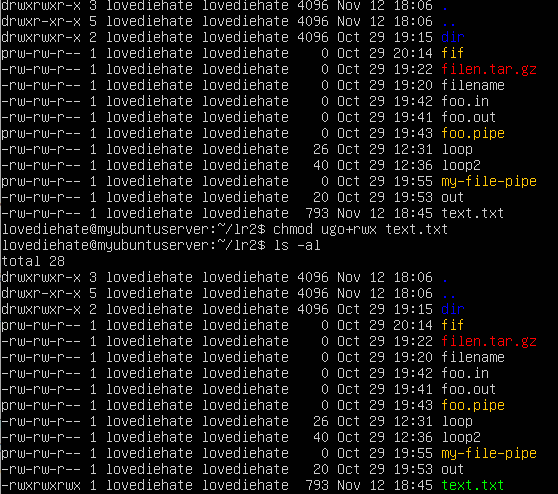


Рисунок - Назначение прав

Команда chmod (Change MODe – сменить режим) – изменяет права доступа к файлу. Для использования этой команды также необходимо иметь права владельца файла или права root . Синтаксис команды таков:

chmod mode filename , где

filename – имя файла, у которого изменяются права доступа;

mode – права доступа, устанавливаемые на файл. Права доступа можно записать в 2 вариантах – символьном и абсолютном.

Команда chown (CHange OWNer – сменить владельца) – позволяет сменить владельца файла. Для использования этой команды необходимо либо иметь права владельца текущего файла или права root . Синтаксис команды прост:

chown username:groupname filename , где

username – имя пользователя – нового владельца файла;

groupname – имя группы – нового владельца файла;

filename – имя файла, у которого сменяется владелец.

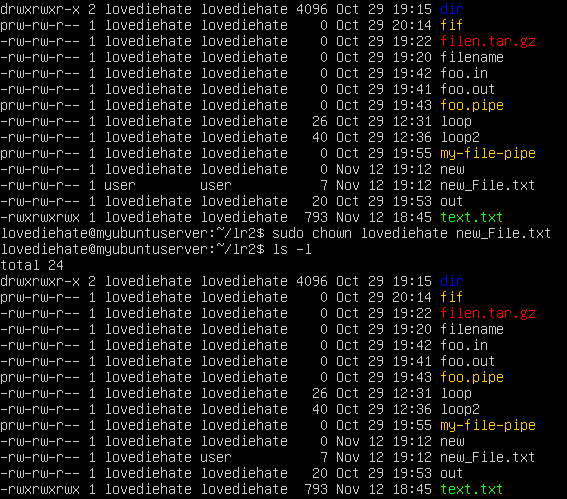


Рисунок - Назначение владельца

1. Ознакомиться с информацией(5) по теме процессы, посмотреть и опробовать примеры наиболее распространенных(6) команд, изучить возможность запуска процессов в supervisor

Процесс – понятие совокупности программного кода и данных, загруженных в память ЭВМ.

Процесс – это не запущенная программа (приложение) или команда, так как приложение может создавать несколько процессов одновременно. Код процесса не обязательно должен выполняться в текущий момент времени, так как процесс может находиться в состоянии спящего. В этом случае выполнение кода такого процесса приостановлено. Существует всего 3 состояния, в которых может находиться процесс:

Работающий процесс – в данный момент код этого процесса выполняется.

Спящий процесс – в данный момент код процесса не выполняется в ожидании какого-либо события (нажатия клавиши на клавиатуре, поступление данных из сети и т.д.)

Процесс-зомби – сам процесс уже не существует, его код и данные выгружены из оперативной памяти, но запись в таблице процессов остается по тем или иным причинам.

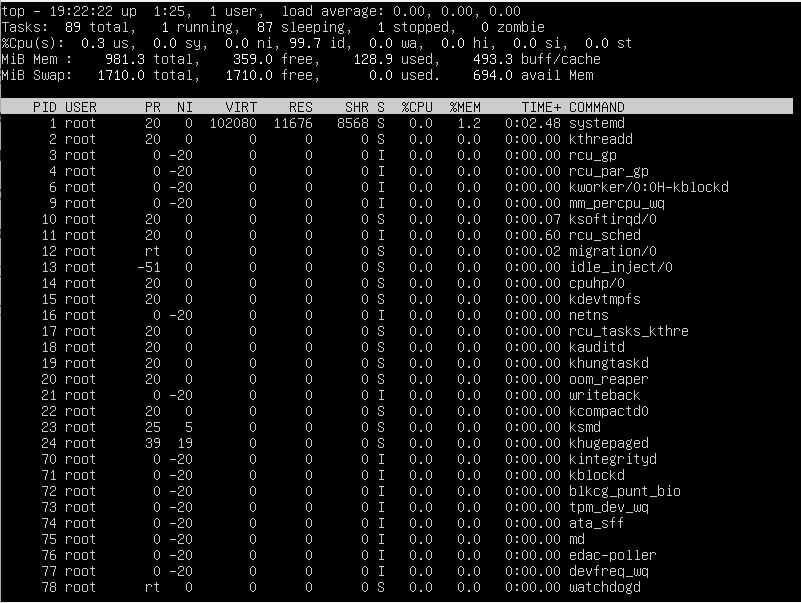
# top

Рисунок 11 - Команда top

# top –u lovediehate

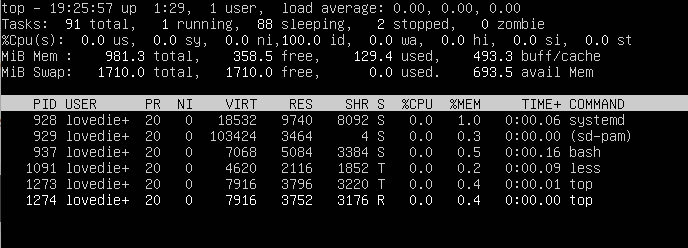


Рисунок – Команда top для пользователя

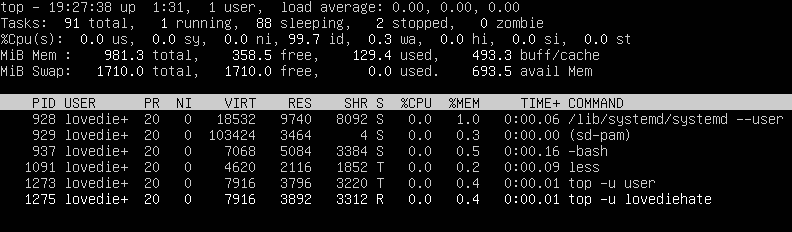


Рисунок - Нажатие 'c'

Теперь виден абсолютный путь к программам.

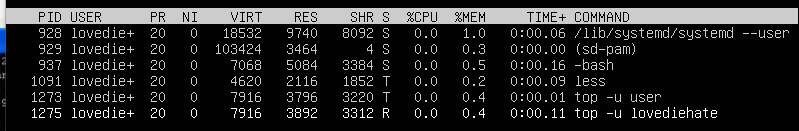


Рисунок – Нажатие 'd'

Изменен интервал обновления снимка.

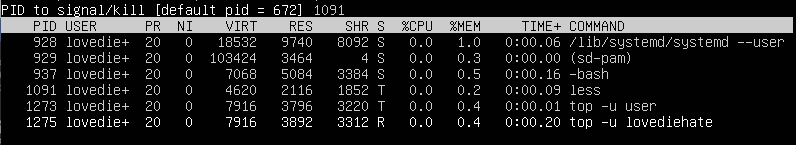


Рисунок - Убийство процесса кнопкой ‘k’

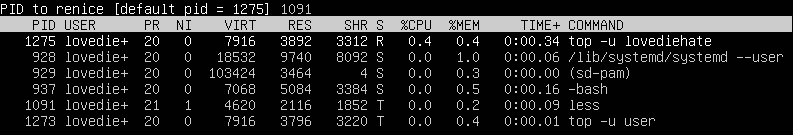


Рисунок – Изменение приоритета процесса с ‘r’

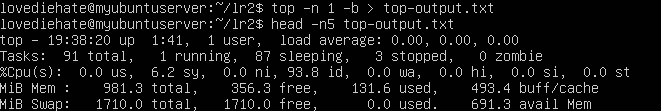


Рисунок – Перенаправление вывода в файл

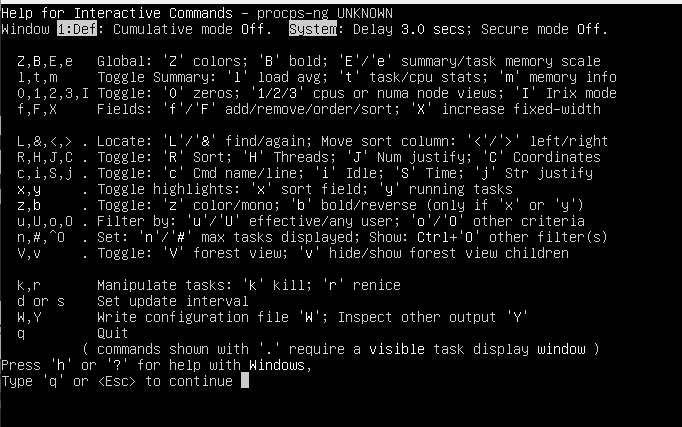


Рисунок – Справка на ‘h’

Supervisor – это менеджер процессов, который существенно упрощает управление долго работающими программами, предоставляя простой и понятный интерфейс.

Для установки нужно ввести команду sudo apt-get install supervisor

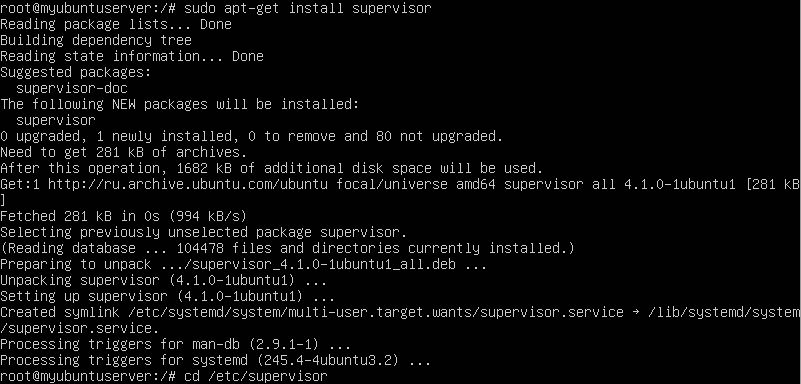


Рисунок – Установка supervisor

Допустим, у нас есть некий скрипт Somebody.

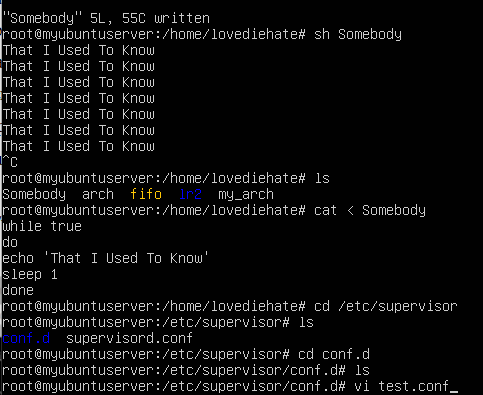


Рисунок – Скрипт

Новые программы добавляются в Supervisor посредством конфигурационных файлов, которые передают переменные среды и сообщают, какой из исполнительных файлов нужно запустить в определенный момент, как обрабатывать вывод.

Создаем тестовый файл конфигураций в /etc/supervisor/conf.d

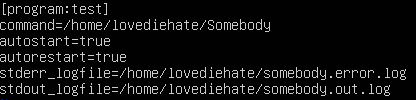


Рисунок – Сожержимое конфига

Ниже приведено описание каждой строки и некоторые тонкие настройки, которые могут пригодиться в дальнейшем.

[program:test]

command=/home/lovediehate/Somebody

Конфигурация начинается с определения программы и полного пути к ней.

autostart=true

autorestart=true

Эти строки определяют базовое автоматическое поведение скрипта в определенных условиях. Опция autostart запускает программу при загрузке системы; значение false заставит включать программу вручную. Опция autorestart определяет, как Supervisor будет управлять программой в случае ее отключения, и имеет три опции:

false – Supervisor никогда не будет перезапускать программу после завершения ее работы;

true – Supervisor будет всегда перезапускать программу после завершения работы;

unexpected – Supervisor будет перезапускать программу только в случае, если она завершила работу из-за возникновения неожиданного кода ошибки (любой стандартный код, кроме 0 и 2).

stderr\_logfile=/home/lovediehate/somebody.err.log

stdout\_logfile=/home/lovediehate/somebody.out.log

Последние две строки определяют местонахождение двух основных лог-файлов программы. В соответствии с именами опций, stdout и stderr задают расположение файлов stdout\_logfile и stderr\_logfile.

Запустим команды supervisorprctl reread && supervisorprctl update, чтобы Supervisor считал новые настройки и они вступили в силу. Затем проверим содержимое лога вывода. Как видно на скрине, Supervisor запустил процесс вывода строки.

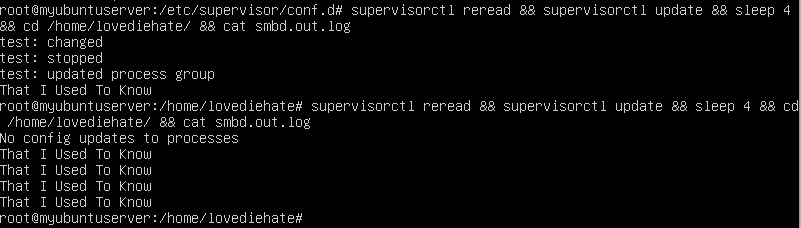


Рисунок – Результат

Для проверки статуса процессов используем команду «supervisorctl». Основные команды: start, stop, restart.



Рисунок – Остановка процесса

1. Запуск процессов по расписанию.

Crontab — это команда, используемая для установки, удаления или вывода файла конфигурации cron, используемого для управления демоном cron. Cron используется для планирования задач, которые будут выполняться периодически.

Описания регулярных действий, запускаемых утилитой– это так называемая crontab-таблица, которая имеет строго определенный формат. Она состоит из 6 колонок, разделённых табуляторами или пробелами, первые 5 из которых определяют время запуска действия: «minute(s) hour(s) day(s) month(s) weekday(s) command(s)». Сначала задаётся колонка минут, затем часов, дней, месяцев и дней недели. Для задания шага значений используется символ «/». Последняя колонка интерпретируется как команда запуска, то есть само действие.

Для редактирования файла расписания необходимо использовать команду crontab –e, а для удаления crontab –r.

Строка \*/1 \* \* \* \* echo `date` >> /home/lovediehate/cronlog.txt означает, что мы каждую минуту будем выводить текущую дату в файл cronlog.txt.



Рисунок – Ввод crontab –e

Как мы видим, дата успешно вывелась в текстовом файле.



Рисунок – Результат в файле

# Вывод

В результате выполнения лабораторной работы я получил знания по работе с процессами в ОС Linux Ubuntu. Научился пользоваться перенаправлением ввода-вывода, «Supervisor», планировщиком задач. Выполнил основные команды просмотра файлов и изучил существующие у них параметры.