**Липецкий государственный технический университет**

Отчет по Лабораторной работе № 3
по дисциплине «Операционная система Linux»
на тему «Управление процессами в Linux»


| | | |
|---|---|---|
| Студент | | Елфимова Д.А. |
| | подпись, дата | фамилия, инициалы |
| Руководитель | | |
| доцент, к.п.н. | | Кургасов В.В. |
| учёная степень, учёное звание | подпись, дата | фамилия, инициалы |


Липецк 2019 г.

**Задание**

1. Повторить команды cat, head, tail, more, less, grep, find
2. Разобраться с понятиями конвейер, перенаправление ввода-вывода.
3. Ознакомится с информацией из рекомендованных источников и других про конвейеризации.
4. Повторить назначение прав доступа. Команды chmod, chown
5. Ознакомиться с информацией по теме процессы, посмотреть и опробовать примеры наиболее распространенных команд, изучить возможность запуска процессов в supervisor
6. Изучить возможность автоматического запуска программ по расписанию.

## Оглавление

## 1. Команды cat, head, tail, more, less, grep, find

Утилита cat является одним из наиболее универсальных инструментов операционной системы. Все, что она делает - это копирует данные из стандартного потока ввода в стандартный поток вывода. Данная утилита в комбинации с командной оболочкой позволяет реализовать мощные и разнообразные механизмы обработки данных. Может использоваться в качестве следующих задач:

— Объединение файлов

```
$ echo один > part1
$ echo два > part2
$ echo три > part3
$ cat part1 part2 part3
```

— Создание файлов

— Копирование файлов

```
$ cat winter.txt
Сегодня очень холодно!
$ cat winter.txt > cold.txt
$ cat cold.txt
Сегодня очень холодно!
```

Утилита head также может использоваться для вывода первых n строк файла.

```
$head -4 /home/test/
```

Кроме того, утилита head может использоваться для вывода первых n байт файла.

```
$head -c4 /home/test/
```

По аналогии с утилитой head, утилита tail может использоваться для вывода последних строк файла.

Утилита more может оказаться полезной в случае возникновения необходимости вывода содержимого файлов, которое не умещается на

экране. Утилита more позволяет ознакомится с содержимым файла, разделенным на страницы. После открытия файла с использованием данной утилиты следует использовать клавишу "пробел"для перехода к следующей странице или клавишу q для выхода из режима просмотра содержимого файла. Некоторые пользователи предпочитают использовать утилиту less вместо утилиты more.

find — утилита поиска файлов по имени и другим свойствам.

grep — утилита командной строки, используется для поиска и фильтрации текста в файлах, на основе шаблона, который может быть регулярным выражением.

Подробнее по последним двум утилитам лучше смотреть в справке, так как возможностей их использования великое множество.

### 2. Перенаправление ввода-вывода

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 программе.

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

### 3. Конвейеризация

Простым примером перенаправления является pipe (конвейер). Обозначается он символом «|». Используется pipe следующим образом:

команда 1 | команда 2 | команда 3 ...

При таком вызове все данные, которые при обычном запуске команды 1 выводились бы на экран будут поступать на стандартный ввод команды 2, как будто бы мы вводим эти данные с клавиатуры. Конвейер можно понимать как канал, в который один процесс может только писать, а другой — только читать из него. Подробнее о каналах в Л.р. 2.

### 4. Назначение прав доступа. Команды chmod, chown

Информация о правах доступа (см Л.р. 1):

$---$ - нет прав;

$--x$ - разрешено только выполнение файла, как программы, но не изменение и не чтение;

$-w-$ - разрешена только запись и изменение файла;

$-wx$ - разрешено изменение и выполнение, но в случае с каталогом, вы не можете посмотреть его содержимое;

$r--$ - права только на чтение;

$r-x$ - только чтение и выполнение, без права на запись;

$rw-$ - права на чтение и запись, но без выполнения;

$rwx$ - все права;

$--s$ - установлен SUID или SGID бит, первый отображается в поле для владельца, второй для группы;

$--t$ - установлен sticky-bit, а значит пользователи не могут удалить этот файл.

В данном случае владельцем файла является пользователь root и группа root. Права доступа представлены цепочкой символов lrwxrwxrwx. Эти символы можно условно разделить на 4 группы. Первая группа, состоящая из единственного символа, определяет тип файла. Этот символ может принимать такие значения:

- — - обычный файл;
- — d = - каталог;
- — b = - файл блочного устройства;
- — c = - файл символьного устройства;
- — s = - доменное гнездо (socket);

— p = - именованный канал (pipe);
— l = - символическая ссылка (link).

Далее следуют три группы по три символа, которые и определяют права доступа к файлу соответственно для владельца файла, для группы пользователей, которая сопоставлена данному файлу, и для всех остальных пользователей системы. Каждый номер в команде chmod представляет собой права для одного из типов пользователей (владелец, группа и другие). Возьмем номер 7. Теперь, используя объяснение числовых значений (r=4,w=2,x=1), единственный вариант для получения числа 7 – сложение чисел 4, 2 и 1, получаем 4+2+1=7. Проще говоря, это означает ВСЕ права доступа (чтение, запись, выполнение – rwx). Первый номер устанавливает права доступа для владельца файла. Второй номер устанавливает права доступа для группы владельца. Третий номер дает группе другие. Третья часть в команде – это имя файла для которого мы изменяем права доступа.

Команда chown применяется, если необходимо поменять владельца/группу каталога или файла.

## 5. Процессы, примеры наиболее распространенных команд, возможность запуска процессов в supervisor

1. Дисплей команды top
   Команда показывает информацию о заданиях, памяти, ЦПУ.
   top
2. Сортировка с -O.
   Сортировка поля по алфавиту поля (SHIFT + O) может производиться, например, с помощью PID (идентификатор процесса) по букве «A».
3. Дисплей процесса конкретного пользователя Использование топ команды с U для отображения конкретных деталей процесса пользователя.
   top -u tecmint
4. Выделите Процесс запуска в верхней части Опция Нажмите ’z’ при запуске верхней команды будет отображать процесс выполнения в цвете, который может помочь вам легко определить процесс выполнения.

5. Показывает Абсолютный Путь Процессов Нажмите 'c' вариант в запуске верхней команды, он будет отображать абсолютный путь запуска процесса.
6. Изменение задержки или установить 'Screen Refresh Interval' в верхней части По умолчанию интервал обновления экрана составляет 3,0 секунды, то же самое можно изменить нажатием 'D' вариант в запуске верхней команды и изменить его по желанию.
7. Убить процесс запуска с аргументом 'k' Вы можете убить процесс после нахождения PID процесса, нажав опцию 'k' при запуске верхней команды.
8. Сортировка по использованию процессора Нажмите (Shift-P) для сортировки процессов в порядке использования процессора.
9. Renice процесс Вы можете использовать опцию 'r', чтобы изменить приоритет процесса, также называемый Renice.
10. Сохранение результатов команды top
    top -n 1 -b > top-output.txt
11. Получение помощи по команде top Нажмите 'h' для ее получения.

Supervisor — это система клиент/сервер, при помощи которой пользователь (администратор) может контролировать подключенные процессы в системах типа UNIX. Инструмент создает процессы в виде подпроцессов от своего имени, поэтому имеет полный контроль над ними. Благодаря этой системе всеми доступными процессами можно управлять через браузер.

Supervisor — простой и достаточно мощный инструмент для контроля процессов. С правильной настройкой он способен обеспечить бесперебойную работу веб-сервиса.

### 6. Автоматический запуск программ по расписанию

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. Коман-

да at требует, по меньшей мере, двух параметров — время выполнения программы и запускаемую программу с ее параметрами запуска. Приведенный ниже пример запустит команду ls на выполнение в 1 час 5 минут. Каждая команда записывается на отдельной строке, т. е. завершается нажатием клавиши <Enter>, а по окончании ввода всех команд — <Ctrl>+<D> . at 1:05 ls > file Помимо времени, в команде at может быть также определена и дата запуска программы на выполнение.

### batch

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. Формат команды batch представляет собой просто список команд для выполнения, следующих в строках за командой; заканчивается список комбинацией клавиш <Ctrl>+<D>. Можно также поместить список команд в файл и перенаправить его на стандартный ввод команды batch.

### cron и crontab

cron — 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.

Каждая строка в файле crontab содержит шаблон времени и команду. Команда выполняется тогда, когда текущее время соответствует приведенному шаблону. Шаблон времени состоит из пяти частей, разделенных пробелами или символами табуляции, и имеет вид: минуты часы день_ месяца месяц день_ недели Эти поля обязательно должны присутствовать в файле.

Для того чтобы программа cron игнорировала какое-то поле шаблона времени, поставьте в нем символ звездочки (*). Например, шаблон 10 01 01 * * говорит о том, что команда должна быть запущена в десять минут второго каждого первого числа любого (*) месяца, каким бы днем недели оно ни было.

## Заключение

В ходе данной лабораторной работы были изучены или повторно рассмотрены некоторые команды ОС Linux, было проведено ознакомление и анализ рекомендованной литературы, а также информации о процессах и запуске программ по расписанию.