Отчет по лабораторной работе №7

Дисциплина: Операционные системы

Калистратова Ксения Евгеньевна

Содержание

[Цель работы 1](#_Toc72171205)

[Задачи 1](#_Toc72171206)

[Выполнение лабораторной работы 1](#_Toc72171207)

[Контрольные вопросы 9](#_Toc72171208)

[Выводы 13](#_Toc72171209)

# Цель работы

Ознакомление с инструментами поиска файлов и фильтрации текстовых данных. Приобретение практических навыков: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.

# Задачи

1. Изучить потоки ввода и вывода.
2. Изучить конвейер.
3. Изучить команду поиска файлов.
4. Изучить команду, позволяющую найти указанную строку символов.
5. Изучить команды по проверке использования диска.
6. В ходе работы использовать эти команды и интерпретировать их вывод.
7. Выполнить отчет.

# Выполнение лабораторной работы

1. Осуществляю вход в систему, используя свои логин и пароль.
2. Для того, чтобы записатьв файл file.txt названия файлов, содержащихся в каталоге /etc, использую команду «ls–a/etc> file.txt». Далее с помощью команды «ls-a~ >> file.txt» дописываюв этот же файл названия файлов, содержащихся в моем домашнем каталоге. Командой «catfile.txt» просматриваю файл, чтобы убедиться в правильности действий. (рис. 1) (рис. 2)

Изображение выглядит как текст

Автоматически созданное описание

Figure 1: Файл file.txt

Изображение выглядит как текст

Автоматически созданное описание

Figure 2: Файл file.txt

1. Вывожуимена всех файлов из file.txt, имеющих расширение .conf и записываюих в новый текстовой файл conf.txt с помощью команды «grep-e‘.conf$’file.txt> conf.txt». Командой «catconf.txt»проверяю правильность выполненных действий. (рис. 3)

Изображение выглядит как текст, монитор, снимок экрана, экран

Автоматически созданное описание

Figure 3: Расширение .conf

1. Определить, какие файлы в моем домашнем каталоге имеют имена, начинающиеся с символа c, можно несколькими командами: «find~ -maxdepth1 -name“c*” -print»(опция maxdepth1 необходима для того, чтобы файлы находились только в домашнем каталоге(не в его подкаталогах)),«ls ~/c*» и «ls –a~ | grepc\*». (рис. 4)

Изображение выглядит как текст

Автоматически созданное описание

Figure 4: Файлы, начинающиеся с символа c

1. Чтобы вывести на экран (по странично) имена файлов из каталога /etc, начинающиеся с символа h, воспользуемся командой «find/etc– maxdepth 1 –name“h\*” | less».

(рис. 5)

Изображение выглядит как текст

Автоматически созданное описание

Figure 5: Файлы, начнающиеся с символа h

(рис. 6)

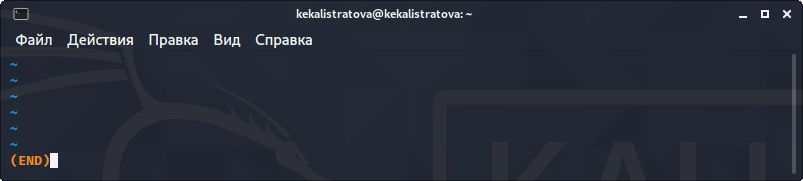


Figure 6: Файлы, начнающиеся с символа h

1. Запускаюв фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log, используя команду «find/ -name “log\*” > logfile &». (рис. 7)

Изображение выглядит как текст

Автоматически созданное описание

Figure 7: Файлы, имена которых начинаются с log.

Командой «cat logfile» проверяю выполненные действия. (рис. 8) (рис. 9)

Изображение выглядит как текст

Автоматически созданное описание

Figure 8: cat logfile

1. Далее удаляю файл ~/logfile командой «rm logfile».(рис. 10)

Изображение выглядит как текст

Автоматически созданное описание

Figure 10: Удаление файла

1. Запускаю редактор gedit в фоновом режиме командой «gedit &». После этого на экране появляется окно редактора. (рис. 11)

Изображение выглядит как текст

Автоматически созданное описание

Figure 11: Редактор degit

1. Чтобы определить идентификатор процесса gedit, использую команду «ps| grep-i “gedit”». Из рисунка видно, что наш процесс имеет PID 7018. Узнать идентификатор процесса можно также, используя команду «pgrep gedit» или «pidof gedit». (рис. 12)

Изображение выглядит как текст

Автоматически созданное описание

Figure 12: Идентификатор процессаgedit

1. Прочитав информацию о команде kill с помощью команды «man kill», используюеё для завершения процесса gedit(команда «kill 11457»). (рис. 13) (рис. 14) (рис. 15)

Figure 13: man kill

Изображение выглядит как текст

Автоматически созданное описание

Figure 14: Справка команды kill

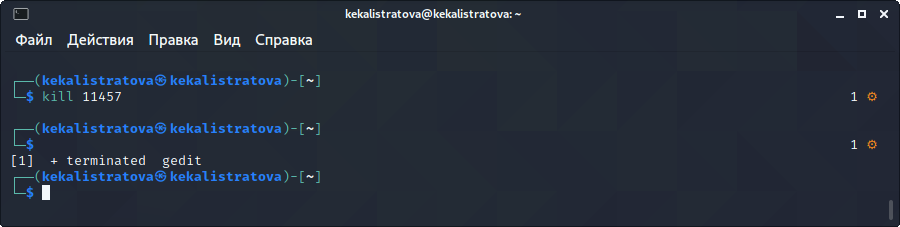


Figure 15: Завершение процесса gedit

1. C помощью команд «man df»и «man du» узнаю информацию по необходимым командам и далее использую их. (рис. 16)

Изображение выглядит как текст

Автоматически созданное описание

Figure 16: Команда man df

df – утилита,показывающаясписок всех файловых систем по именам устройств, сообщает их размер, занятое и свободное пространство и точки монтирования. (рис. 17)

Изображение выглядит как текст

Автоматически созданное описание

Figure 17: Команда man

Синтаксис: df[опции]устройство. (рис. 18)

Изображение выглядит как текст, черный, табличка, снимок экрана

Автоматически созданное описание

Figure 18: Команда df

du – утилита, предназначенная для вывода информации об объеме дискового пространства, занятого файлами и директориями. Она принимает путь к элементу файловой системы и выводит информацию о количестве байт дискового пространства или блоков диска, задействованных для его хранения. (рис. 19)

Изображение выглядит как текст

Автоматически созданное описание

Figure 19: Команда man du

Синтаксис: du [опции] каталог\_или\_файл. (рис. 20)

Изображение выглядит как текст, черный, табличка, снимок экрана

Автоматически созданное описание

Figure 20: Команда du

1. Вывожу имена всех директорий, имеющихся в моем домашнем каталоге с помощью команды «find ~ -type d», предварительно получив информацию с помощью команды «man find»/. (рис. 21) (рис. 22)

Изображение выглядит как текст

Автоматически созданное описание

Figure 21: Справка команды find

Изображение выглядит как текст

Автоматически созданное описание

Figure 22: Команда find

# Контрольные вопросы

1. В системе по умолчанию открыто три специальных потока:

–stdin − стандартный поток ввода (по умолчанию: клавиатура), файловый дескриптор 0;

–stdout − стандартный поток вывода (по умолчанию: консоль), файловый дескриптор 1;

-stderr − стандартный поток вывод сообщений об ошибках (поумолчанию: консоль), файловый дескриптор 2.

Большинство используемых в консоли команд и программ записывают результаты своей работы в стандартный поток вывода stdout.

1. ‘>’ Перенаправление вывода в файл

‘>>’ Перенаправление вывода в файл и открытие файла в режиме добавления (данные добавляются в конец файла)/

1. Конвейер (pipe) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей.

Синтаксис следующий:

команда1|команда2 (это означает, что вывод команды 1 передастся на ввод команде 2)

1. Процесс рассматривается операционной системой как заявка на потребление всех видов ресурсов, кроме одного − процессорного времени. Этот последний важнейший ресурс распределяется операционной системой между другими единицами работы − потоками, которые и получили свое название благодаря тому, что они представляют собой последовательности (потоки выполнения) команд.

Процесс − это выполнение программы. Он считается активной сущностью и реализует действия, указанные в программе.

Программа представляет собой статический набор команд, а процесс это набор ресурсов и данных, использующихся при выполнении программы.

1. pid: идентификатор процесса (PID) процесса (processID), к которому вызывают метод

gid: идентификатор группы UNIX, в котором работает программа.

1. Любую выполняющуюся в консоли команду или внешнюю программу можно запустить в фоновом режиме. Для этого следует в конце имени команды указать знак амперсанда &.

Запущенные фоном программы называются задачами (jobs). Ими можно управлять с помощью команды jobs, которая выводит список запущенных в данный момент задач.

1. top − это консольная программа, которая показывает список работающих процессов в системе. Программа в реальном времени отсортирует запущенные процессы по их нагрузке на процессор.

htop − это продвинутый консольный мониторинг процессов. Утилита выводит постоянно меняющийся список системных процессов, который сортируется в зависимости от нагрузки на ЦПУ. Если делать сравнение сtop, то htop показывает абсолютно все процессы в системе, время их непрерывного использования, загрузку процессоров и расход оперативной памяти.

1. find − это команда для поиска файлов и каталогов на основе специальных условий. Ее можно использовать в различных обстоятельствах, например, для поиска файлов по разрешениям, владельцам, группам, типу, размеру и другим подобным критериям.

Команда find имеет такой синтаксис:

find[папка][параметры] критерий шаблон [действие]

Папка − каталог в котором будем искать

Параметры − дополнительные параметры, например, глубина поиска, и т д.

Критерий − по какому критерию будем искать: имя, дата создания, права, владелец и т д.

Шаблон – непосредственно значение по которому будем отбирать файлы.

Основные параметры:

-P никогда не открывать символические ссылки

-L - получает информацию о файлах по символическим ссылкам. Важно для дальнейшей обработки, чтобы обрабатывалась не ссылка, а сам файл.

-maxdepth - максимальная глубина поиска по подкаталогам,для поиска только в текущем каталоге установите 1.

-depth - искать сначала в текущем каталоге, а потом в подкаталогах

-mount искать файлы только в этой файловой системе.

-version - показать версию утилиты find

-print - выводить полные имена файлов

-typef - искать только файлы

-typed - поиск папки в Linux

Основные критерии:

-name - поиск файлов по имени

-perm - поиск файлов в Linux по режиму доступа

-user - поиск файлов по владельцу

-group - поиск по группе

-mtime - поиск по времени модификации файла

-atime - поиск файлов по дате последнего чтения

-nogroup - поиск файлов, не принадлежащих ни одной группе

-nouser - поиск файлов без владельцев

-newer - найти файлы новее чем указанный

-size - поиск файлов в Linux по их размеру

Примеры:

find~ -type d поиск директорий в домашнем каталоге

find~ -type f -name ".\*" поиск скрытых файлов в домашнем каталоге

1. Файл по его содержимому можно найти с помощью команды grep: «grep -r" слово/выражение, которое нужно найти"».
2. Утилита df, позволяет проанализировать свободное пространство на всех подключенных к системе разделах.
3. При выполнении команды du (без указания папки и опции) можно получить все файлы и папки текущей директории с их размерами. Для домашнего каталога: du ~/
4. Основные сигналы (каждый сигнал имеет свой номер), которые используются для завершения процесса:

* SIGINT–самый безобидный сигнал завершения, означает Interrupt. Он отправляется процессу, запущенному из терминала с помощью сочетания клавиш Ctrl+C. Процесс правильно завершает все свои действия и возвращает управление;
* SIGQUIT–это еще один сигнал, который отправляется с помощью сочетания клавиш, программе, запущенной в терминале. Он сообщает ей что нужно завершиться и программа может выполнить корректное завершение или проигнорировать сигнал. В отличие от предыдущего, она генерирует дамп памяти. Сочетание клавиш Ctrl+/;
* SIGHUP–сообщает процессу, что соединение с управляющим терминалом разорвано, отправляется, в основном, системой при разрыве соединения с интернетом;
* SIGTERM–немедленно завершает процесс, но обрабатывается программой, поэтому позволяет ей завершить дочерние процессы и освободить все ресурсы;
* SIGKILL–тоже немедленно завершает процесс, но, в отличие от предыдущего варианта, он не передается самому процессу, а обрабатывается ядром. Поэтому ресурсы и дочерние процессы остаются запущенными.

Также для передачи сигналов процессам в Linux используется утилита kill, её синтаксис: kill [-сигнал] [pid\_процесса] (PID – уникальный идентификатор процесса). Сигнал представляет собой один из выше перечисленных сигналов для завершения процесса.

Перед тем, как выполнить остановку процесса, нужно определить его PID. Для этого используют команды ps и grep. Команда ps предназначена для вывода списка активных процессов в системе и информации о них. Команда grep запускается одновременно с ps (вканале) и будет выполнять поиск по результатам команды ps.

Утилита pkill – это оболочка для kill, она ведет себя точно так же, и имеет тот же синтаксис, только в качестве идентификатора процесса ей нужно передать ег оимя.

killall работает аналогично двум предыдущим утилитам. Она тоже принимает имя процесса в качестве параметра и ищет его PID в директории /proc. Но эта утилита обнаружит все процессы с таким именем и завершит их.

# Выводы

В ходе выполнения данной лабораторной работы я изучила инструментыпоиска файлови фильтрации текстовых данных, а также приобрела практические навыки: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.