Отчет по лабораторной работе №7

Дисциплина

Филиппова Анна Дмитриевна

Содержание

# 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.

# Выполнение лабораторной работы

1. Осуществляем вход в систему, используя свои логин и пароль.
2. Для того, чтобы записать в файл file.txt названия файлов, содержащихся в каталоге /etc, используем команду «ls –a /etc > file.txt». C помощью команды «ls -a ~ >> file.txt» дописываем в этот файл названия файлов, содержащихся в моем домашнем каталоге. Командой «cat file.txt» просматриваем файл, чтобы убедиться в правильности действий. (рис. 1) (рис. 2)

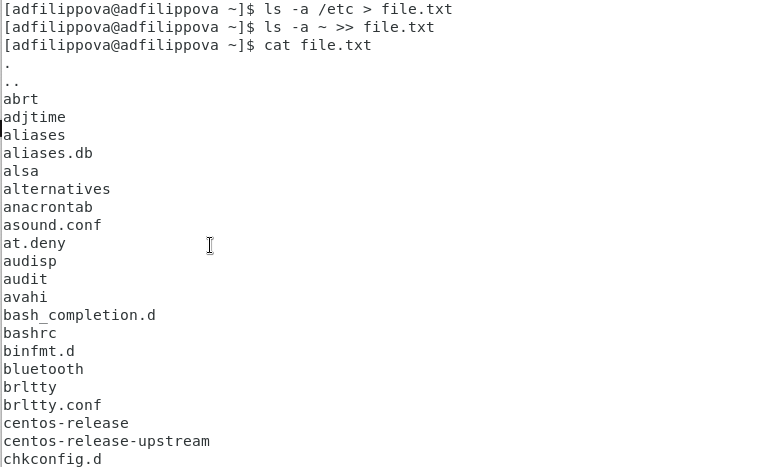


Figure 1: Запись в файл названий каталогов

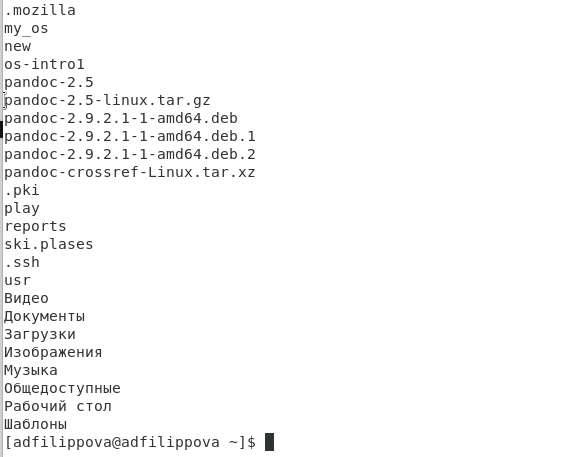


Figure 2: Содержимое файла

1. Выводим имена всех файлов из file.txt, имеющих расширение .conf и записываем их в новый текстовой файл conf.txt с помощью команды «grep -e ‘.conf$’ file.txt > conf.txt». Командой «cat conf.txt» проверяем правильность выполненных действий. (рис. 3)



Figure 3: Вывод файлов с определенным расширением и запись их в другой файл

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

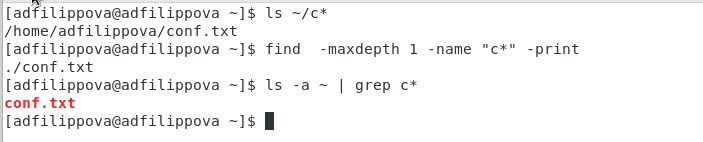


Figure 4: Команды для поиска определенных файлов

1. Воспользуемся командой «find /etc –maxdepth 1 –name “h\*” | less», чтобы вывести на экран (по странично) имена файлов из каталога /etc, начинающиеся с символа h. (рис. 5) (рис. 6)

Figure 5: Использование команды find

Figure 5: Использование команды find



Figure 6: Использование команды find

1. Запускаем в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log, используя команду «find / -name “log\*” > logfile &». Командой «cat logfile» проверяем выполненные действия. (рис. 7) (рис. 8)

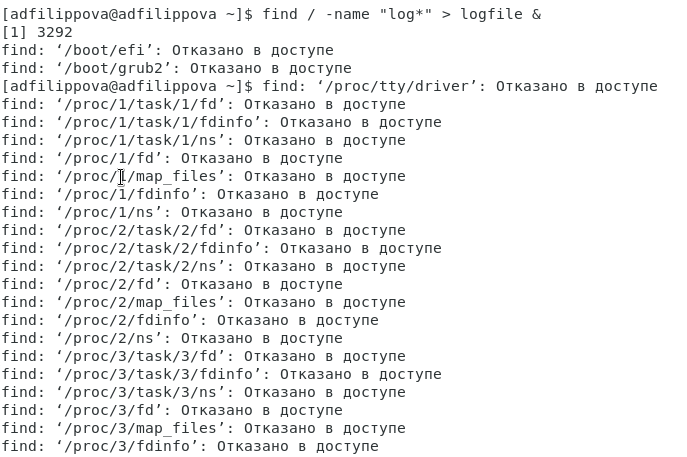


Figure 7: Запуск процесса в фоновом режиме

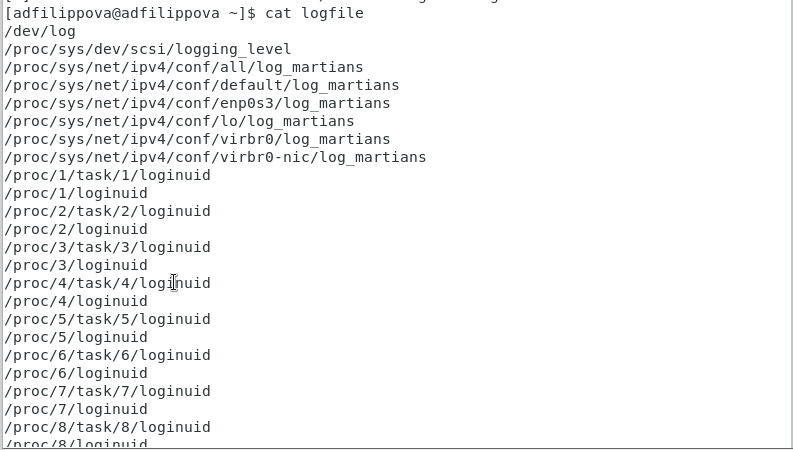


Figure 8: Проверка действий

1. Удаляем файл ~/logfile командой «rm logfile». (рис. 9)

Figure 9: Удаление файла

Figure 9: Удаление файла

1. Запускаем редактор gedit в фоновом режиме командой «gedit &», на экране появляется окно редактора. Чтобы определить идентификатор процесса gedit, используем команду «ps | grep -i “gedit”». Процесс имеет PID 3683. Узнать идентификатор процесса можно также, используя команду «pgrep gedit» или «pidof gedit». (рис. 10)

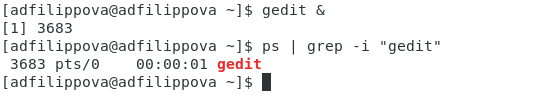


Figure 10: Запуск gedit

1. см. пункт 8
2. Прочитав информацию о команде kill с помощью команды «man kill», используем её для завершения процесса gedit (команда «kill 3683»). (рис. 11) (рис. 12) (рис. 13)

Figure 11: Команда man

Figure 11: Команда man

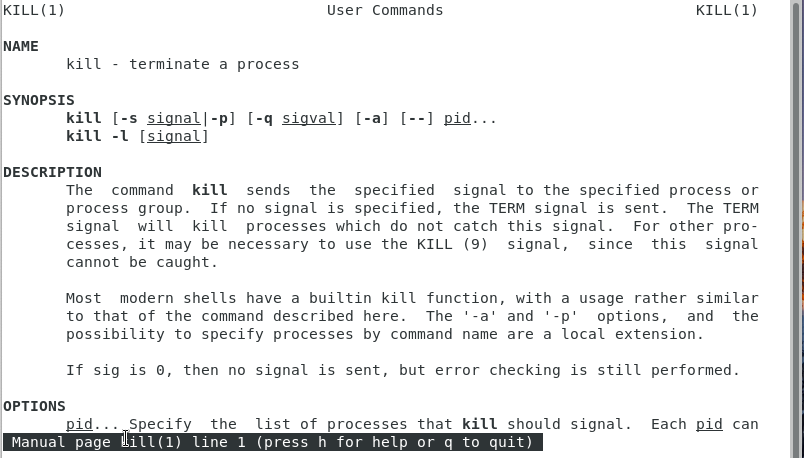


Figure 12: Справка о команде kill

Figure 13: Завершение процесса gedit

Figure 13: Завершение процесса gedit

1. C помощью команд «man df» и «man du» узнаем информацию по необходимым командам и далее использую их. df – утилита, показывающая список всех файловых систем по именам устройств, сообщает их размер, занятое и свободное пространство и точки монтирования. Синтаксис: df [опции] устройство 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. Синтаксис: du [опции] каталог\_или\_файл (рис. 14) (рис. 15) (рис. 16) (рис. 17) (рис. 18)

Figure 14: Команда man

Figure 14: Команда man

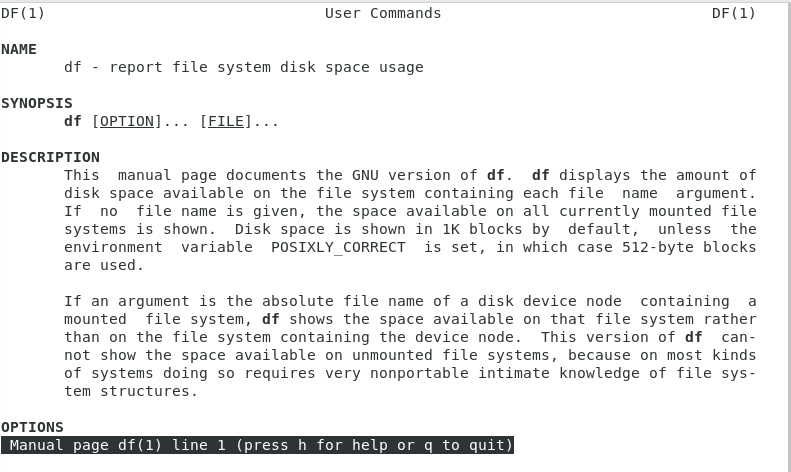


Figure 15: Справка df

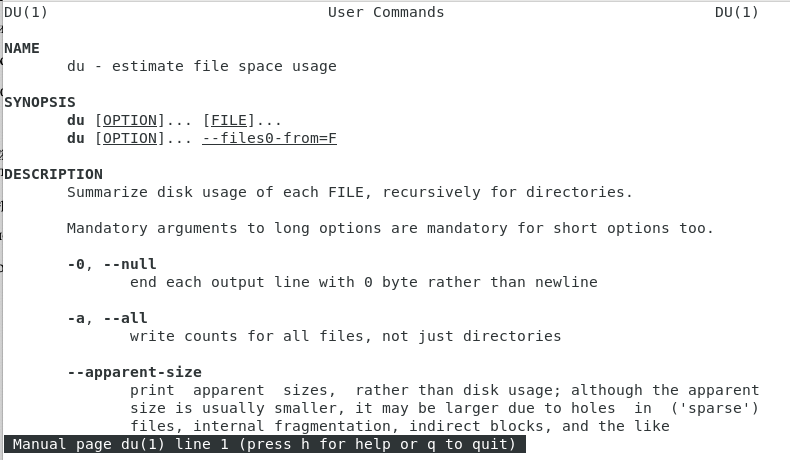


Figure 16: Справка du

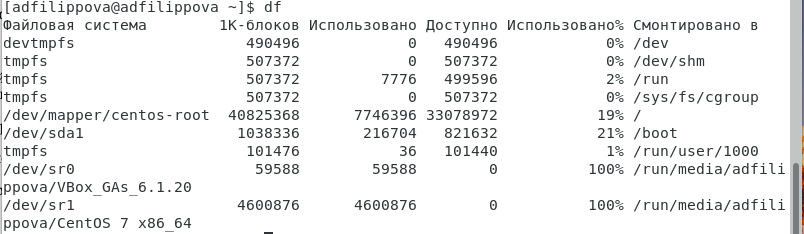


Figure 17: Использовние команды du



Figure 18: Использовние команды df

1. Выводим имена всех директорий, имеющихся в моем домашнем каталоге с помощью команды «find ~ -type d», предварительно получив информацию с помощью команды «man find». (рис. 19) (рис. 20) (рис. 21)

Figure 19: Команда man

Figure 19: Команда man

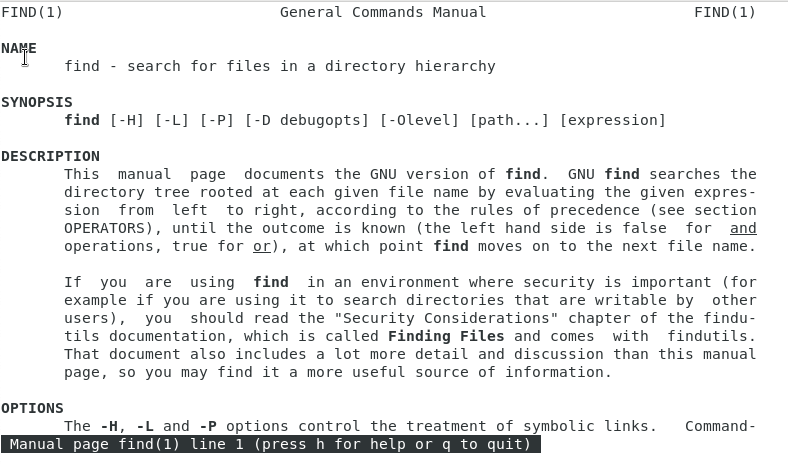


Figure 20: Справка о find

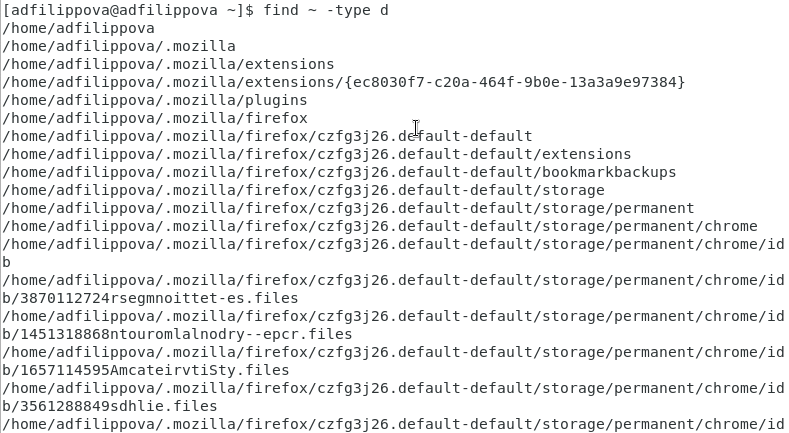


Figure 21: Выполнение команды

# Выводы

Я изучила инструменты поиска файлов и фильтрации текстовых данных, а также приобрела практические навыки: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.

# 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 (по умолчанию: консоль), файловый дескриптор 2. Большинство используемых в консоли команд и программ записывают результаты своей работы в стандартный поток вывода stdout.
2. “>” Перенаправление вывода в файл “>>” Перенаправление вывода в файл и открытие файла в режиме добавления (данные добавляются в конец файла).
3. Конвейер служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей. Синтаксис следующий: команда 1 | команда 2 (это означает, что вывод команды 1 передастся на ввод команде 2).
4. 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.
5. pid: идентификатор процесса (PID) процесса (process ID), к которому вызывают метод gid: идентификатор группы UNIX, в котором работает программа.
6. 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. Ими можно управлять с помощью команды jobs, которая выводит список запущенных в данный момент задач.
7. 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 с top, то htop показывает абсолютно все процессы в системе, время их непрерывного использования, загрузку процессоров и расход оперативной памяти.
8. 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 -type f - искать только файлы -type d - 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 каталог
9. Файл по его содержимому можно найти с помощью команды grep: «grep-r “слово/выражение, которое нужно найти”».
10. Утилита df, позволяет проанализировать свободное пространство на всех подключенных к системе разделах.
11. При выполнении команды du (без указания папки и опции) можно получить все файлы и папки текущей директории с их размерами. Для домашнего каталога: du ~/
12. 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%20–%20уникальный%20идентификатор%20процесса). 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 (в канале) 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 его имя.

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.