Отчет по лабораторной работе №6

*дисциплина: Операционные системы*

Егорова Юлия Владимировна

Содержание

# 1 Цель работы

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

# 2 Выполнение лабораторной работы

1)Запиcываем в файл file.txt названия файлов, содержащихся в каталоге /etc. Дописываем в этот же файл названия файлов, содержащихся в домашнем каталоге.

Рис. 1: Запись в file.txt названия файлов из каталога /etc и домашнего каталога.

Рис. 1: Запись в file.txt названия файлов из каталога /etc и домашнего каталога.

2)Выводим имена всех файлов из file.txt, имеющих расширение .conf, после записываем их в новый текстовый файл conf.txt.

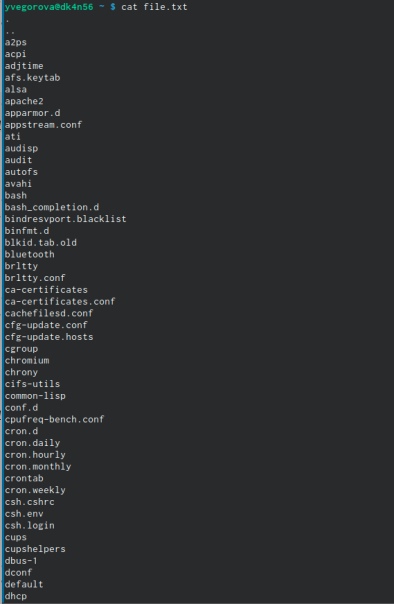


Рис. 2: Вывод файлов.

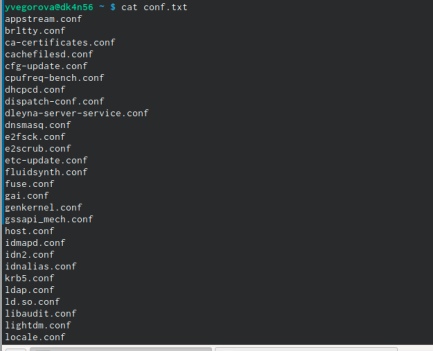


Рис. 3: Запись в новый текстовый файл.

3)Определяем, какие файлы в домашнем каталоге имеют имена, начинающиеся с символа “с” с помощью команды find.

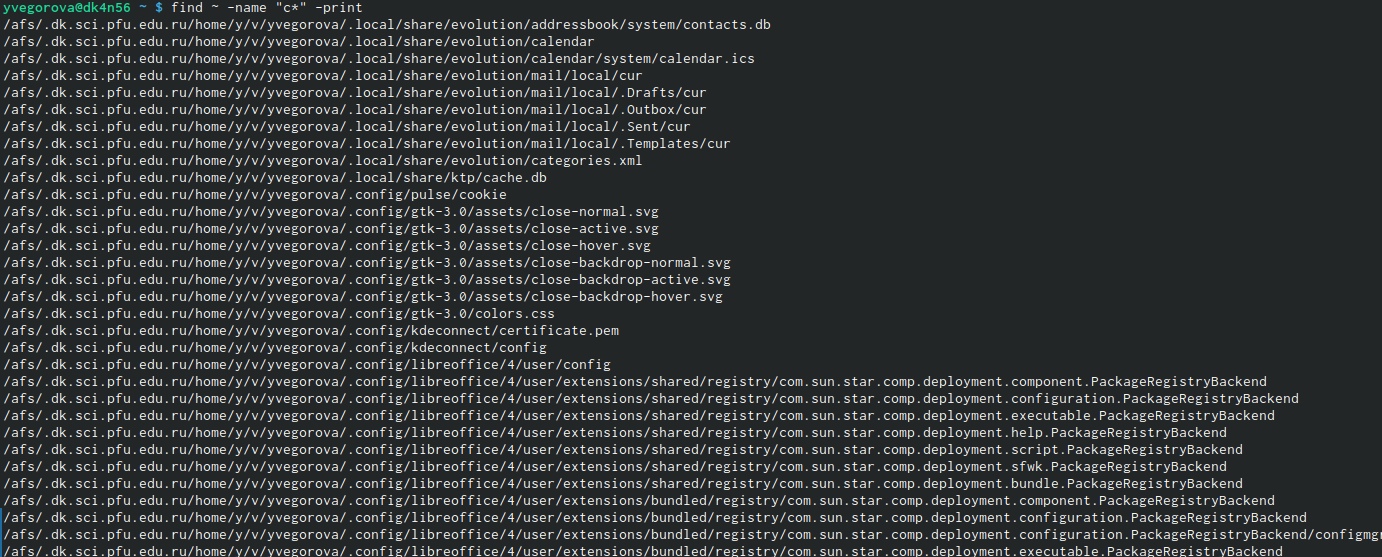


Рис. 4: Поиск файлов, начинающиеся с символа “с”.

4)Выводим на экран имена файлов, начинающиеся с символа h из каталога /etc.

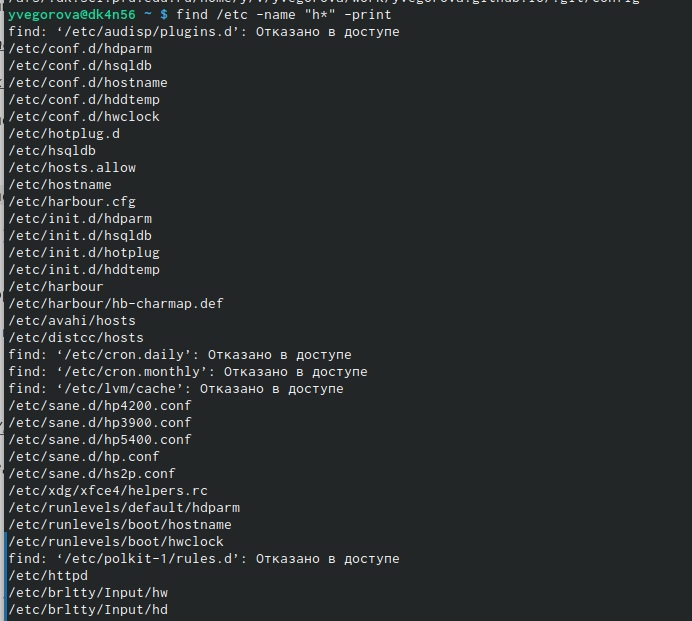


Рис. 5: Поиск файлов, начинающиеся с символа “h”

5)Запускаем в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log.

Рис. 6: Запуск процесса записи в файл файлов, начинающихся с log.

Рис. 6: Запуск процесса записи в файл файлов, начинающихся с log.

6)Удаляем файл ~/logfile.

Рис. 7: Удаление файла.

Рис. 7: Удаление файла.

7)Запускаем из консоли в фоновом режиме редактор gedit.

Рис. 8: Запуск в фоновом режиме редактора gedit.

Рис. 8: Запуск в фоновом режиме редактора gedit.

8)Определяем идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep.

Рис. 9: Определение идентификатора процесса gedit.

Рис. 9: Определение идентификатора процесса gedit.

9)Читаем справку команды kill c помощью команды man, после чего используем ее для завершения процесса gedit.

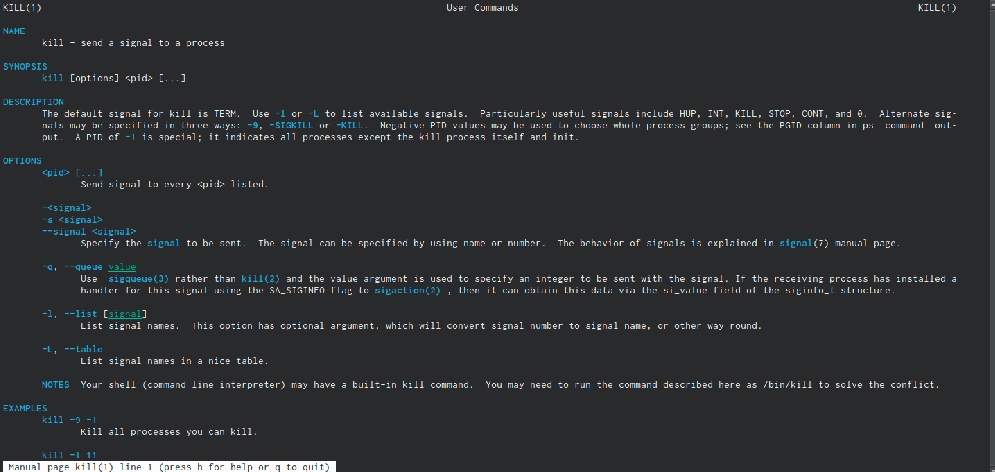


Рис. 10: Справка команды kill.

Рис. 11: Завершение процесса gedit.

Рис. 11: Завершение процесса gedit.

10)Выполняем команды df и du, предварительно получаем более подробную информацию об этих командах с помощью команды man.

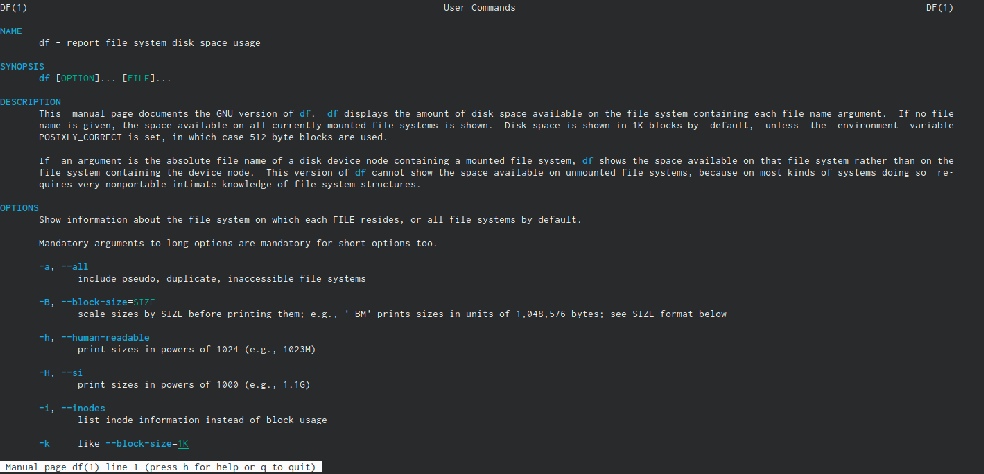


Рис. 12: Информация о команде df.

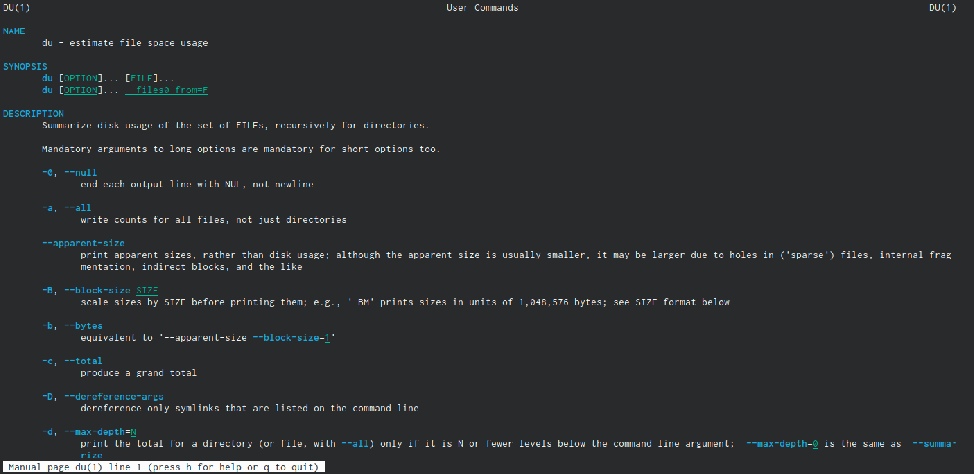


Рис. 13: Информация о команде du.

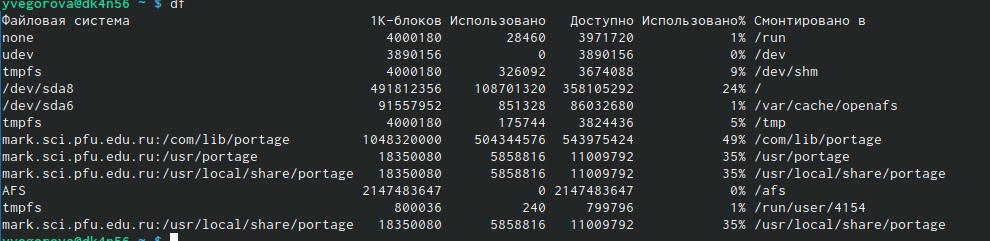


Рис. 14: Команда df.

Рис. 15: Команда du.

Рис. 15: Команда du.

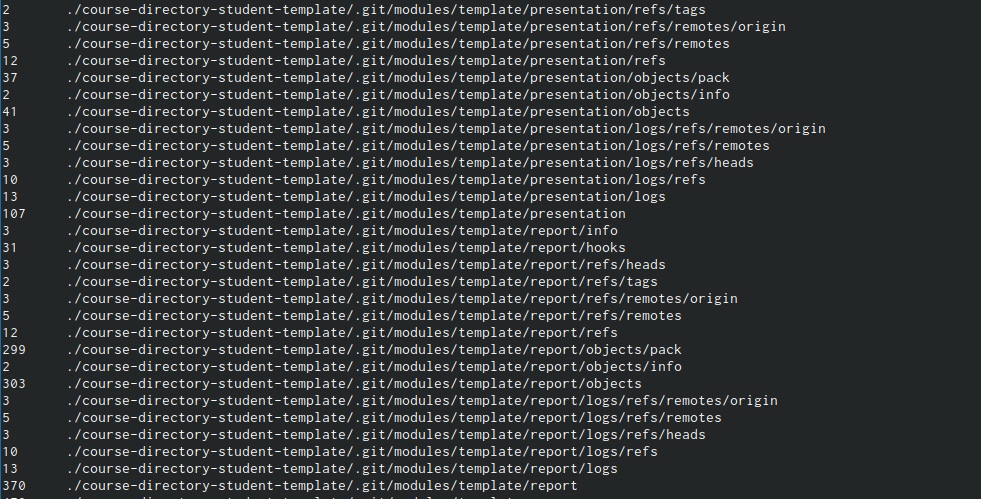


Рис. 16: Команда du.

11)Воспользовавшись справкой команды find, выводим имена всех директорий, имеющихся в домашнем каталоге.

Рис. 17: Вывод всех директорий, имеющихся в домашнем каталоге.

Рис. 17: Вывод всех директорий, имеющихся в домашнем каталоге.

# 3 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 (pipe) 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 (jobs). 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 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.14 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.

# 4 Вывод

В ходе данной лабораторной работы я ознакомилась с инструментами поиска файлов и фильтрации текстовых данных, приобрела практические навыки: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.