Лабораторная работа №6

Поиск файлов. Перенаправление ввода-вывода. Просмотр запущенных процессов

Гнатюк Анастасия Станиславовна

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.

# Задание

1. Осуществите вход в систему, используя соответствующее имя пользователя.
   1. Запишите в файл file.txt названия файлов, содержащихся в каталоге /etc. Допишите в этот же файл названия файлов, содержащихся в вашем домашнем каталоге.
      1. Выведите имена всех файлов из file.txt, имеющих расширение .conf, после чего запишите их в новый текстовой файл conf.txt.
         1. Определите, какие файлы в вашем домашнем каталоге имеют имена, начинавшиеся с символа c? Предложите несколько вариантов, как это сделать.
            1. Выведите на экран (по странично) имена файлов из каталога /etc, начинающиеся с символа h.

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

Удалите файл ~/logfile.

Запустите из консоли в фоновом режиме редактор gedit.

Определите идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep. Как ещё можно определить идентификатор процесса?

Прочтите справку (man) команды kill, после чего используйте её для завершения процесса gedit.

Выполните команды df и du, предварительно получив более подробную информацию об этих командах, с помощью команды man.

Воспользовавшись справкой команды find, выведите имена всех директорий, имеющихся в вашем домашнем каталоге.

# Теоретическое введение

Перенаправление ввода-вывода

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. Например, команда ls выводит в стандартный поток вывода (консоль) список файлов в текущей директории. Потоки вывода и ввода можно перенаправлять на другие файлы или устройства. Проще всего это делаетсяс помощью символов >, >>, <, <<.

**Конвейер**

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

1 команда 1 | команда 2

Конвейеры можно группировать в цепочки и выводить с помощью перенаправления в файл, например:

*1 ls -la |sort > sortilg\_list*

Чаще всего скрипты на Bash используются в качестве автоматизации каких-то рутинных операций в консоли, отсюда иногда возникает необходимость в обработке stdout одной команды и передача на stdin другой команде, при этом результат выполнения команды должен обработан.

**Поиск файла**

Команда find используется для поиска и отображения на экран имён файлов, соответствующих заданной строке символов.

Формат команды:

*1 find путь [-опции]*

Путь определяет каталог, начиная с которого по всем подкаталогам будет вестись поиск.

**Фильтрация текста**

Найти в текстовом файле указанную строку символов позволяет команда grep.

Формат команды:

*1 grep строка имя\_файла*

Кроме того, команда grep способна обрабатывать стандартный вывод других команд (любой текст). Для этого следует использовать конвейер, связав вывод команды с вводом grep.

**Проверка использования диска**

Команда df показывает размер каждого смонтированного раздела диска.

Формат команды:

*1 df [-опции] [файловая\_система]*

**Управление задачами**

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

*1 gedit &*

Будет запущен текстовой редактор gedit в фоновом режиме. Консоль при этом не будет заблокирована.

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

Для завершения задачи необходимо выполнить команду

*1 kill %номер задачи*

**Управление процессами**

Любой команде, выполняемой в системе, присваивается идентификатор процесса (process ID). Получить информацию о процессе и управлять им, пользуясь идентификатором процесса, можно из любого окна командного интерпретатора.

Получение информации о процессах

Команда ps используется для получения информации о процессах.

Формат команды:

*1 ps [-опции]*

Для получения информации о процессах, управляемых вами и запущенных v(работающих или остановленных) на вашем терминале, используйте опцию aux. Для запуска команды в фоновом режиме необходимо в конце командной строки указать знак & (амперсанд).

Пример работы, требующей много машинного времени для выполнения, и которую целесообразно запустить в фоновом режиме:

*1 find /var/log -name "*.log" -print > l.log &\*

# Выполнение лабораторной работы

1. Я заранее осуществила вход в систему.
2. С помощью команды ls -a мы берём все файлы, находящиееся в каталоге etc и переносим с помощью >> в файл file.txt.

Рис.1: Перенос файлов из каталога /etc в file.txt

Рис.1: Перенос файлов из каталога /etc в file.txt

Рис.2: Перенос файлов из домашненго каталога в file.txt

Рис.2: Перенос файлов из домашненго каталога в file.txt

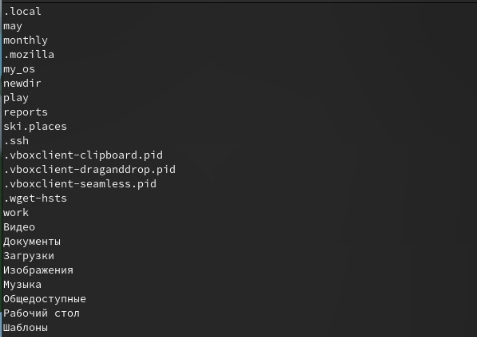


Рис.3: Файлы домашней директории в file.txt

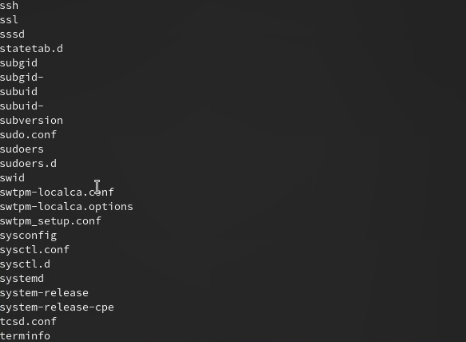


Рис.4: Файлы /etc в file.txt

1. С помощью команды grep мы находим в файле file.txt все файлы, которые в названии имеют conf. Затем создаём файл с помощью команды touch conf.txt, куда мы и переносим все файлы с расширением .conf из file.txt.

Рис.5

Рис.5

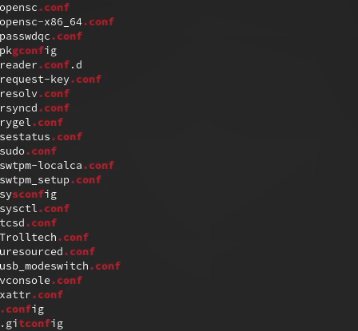


Рис.6

Рис.7

Рис.7

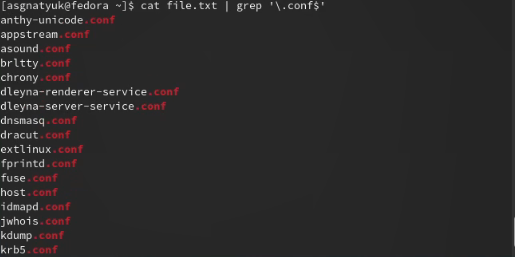


Рис.8

Рис.9

Рис.9

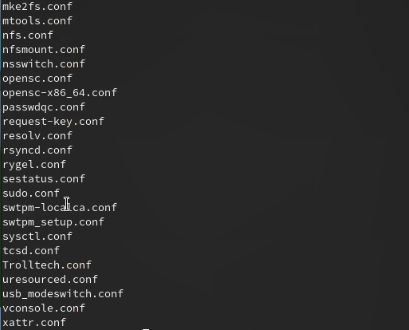


Рис.10

1. С помощью команд ls -a | grep c\* и file c\* выводим все файлы, названия которых начинаются с символа с.

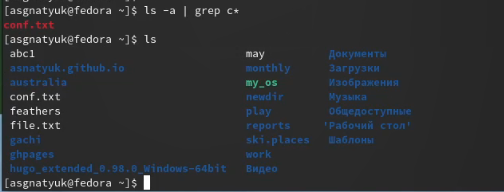


Рис.11: Первый способ

Рис.12: Второй способ

Рис.12: Второй способ

1. С помощью команды find /etc -name "h\*" -print выводим все файлы из каталога /etc, которые начинаются на h.

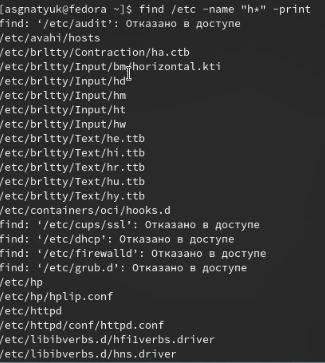


Рис.13

1. С помощью команды find / -name "log\*" -print > logfile переносим все файлы из домашней директории, которые начинаются с log, в logfile.

Рис.14:

Рис.14:

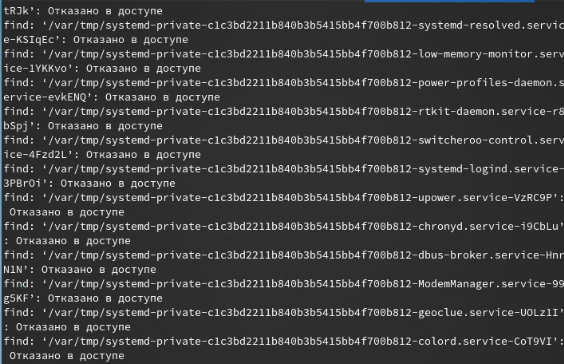


Рис.15:

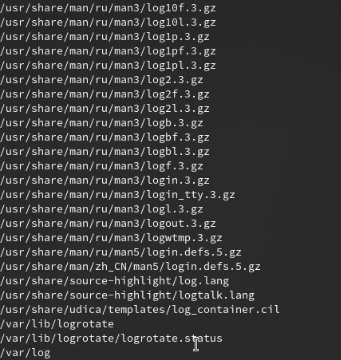


Рис.17: С помощью команды cat проверяем содержимое файла logfile

1. С помощью команды rm удаляем файл logfile.

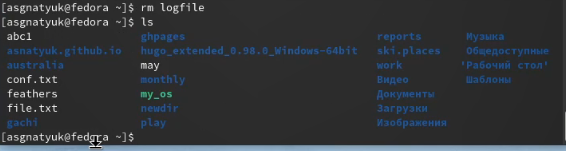


Рис.16

1. С помощью & запускаем gedit в фоновом режиме.

Рис.18

Рис.18

1. С помощью команды ps изнаём идентификатор процесса gedit -> 5571.

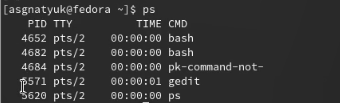


Рис.19

1. С помощью команды man мы изучаем информацию о команде kill.

Рис.20

Рис.20

1. С помощью команды kill прекращаем работу gedit.

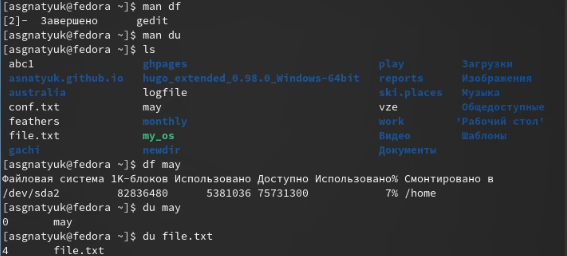


Рис.21

1. С помощью команды find -type d находим все каталоги нашей домашней директории.

Рис.22

Рис.22

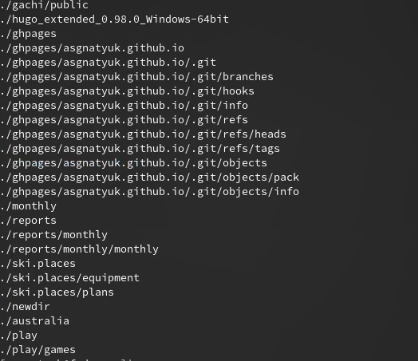


Рис.23

# Выводы

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

# Контрольные вопросы

**1. Какие потоки ввода вывода вы знаете?**

– 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.

**2. Объясните разницу между операцией > и >>.**

“>” - это открывает файл на перезапись, когда “>>” открывает файл на дозапись

**3. Что такое конвейер?**

Направление вывода на вход для следующей команды

**4. Что такое процесс? Чем это понятие отличается от программы?**

Процесс - это исполняемая программа.

Программа - это набор инструкций, которые выполняют определенную задачу при выполнении компьютером, в то время как процесс является экземпляром выполняемой компьютерной программы. Таким образом, в этом главное отличие программы и процесса.

**5. Что такое PID и GID?**

pid: это идентификатор процесса (PID) процесса, который вы вызываете

GID: идентификатор группы. Все группы Linux определяются GID (идентификаторами групп). GID хранятся в файле / etc / groups.