Отчет по выполнению лабораторной работы №8

Операционные системы

Кузьмин Егор Витальевич

Содержание

# 1 Цель работы

Целью данной лабораторной работы является ознакомление с инструментами поиска файлов и фильтрации текстовых данных, а также приобретение практических навыков по управлению процессами, использованию диска и по обслуживанию файловых систем.

# 2 Задание

1. Ознакомиться с необходимой технической информацией
2. Осуществите вход в систему, используя соответствующее имя пользователя.
3. Запишите в файл file.txt названия файлов, содержащихся в каталоге /etc. Допи- шите в этот же файл названия файлов, содержащихся в вашем домашнем каталоге.
4. Выведите имена всех файлов из file.txt, имеющих расширение .conf, после чего запишите их в новый текстовой файл conf.txt.
5. Определите, какие файлы в вашем домашнем каталоге имеют имена, начинавшиеся с символа c? Предложите несколько вариантов, как это сделать.
6. Выведите на экран (по странично) имена файлов из каталога /etc, начинающиеся с символа h.
7. Запустите в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log.
8. Удалите файл ~/logfile.
9. Запустите из консоли в фоновом режиме редактор gedit.
10. Определите идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep. Как ещё можно определить идентификатор процесса?
11. Прочтите справку (man) команды kill, после чего используйте её для завершения процесса gedit.
12. Выполните команды df и du, предварительно получив более подробную информацию об этих командах, с помощью команды man.
13. Воспользовавшись справкой команды find, выведите имена всех директорий, имею- щихся в вашем домашнем каталоге.

# 3 Теоретическое введение

В интерфейсе командной строки есть очень полезная возможность перенаправления (переадресации) ввода и вывода (англ. термин I/O Redirection). Как мы уже заметили, многие программы выводят данные на экран. А ввод данных в терминале осуществляется с клавиатуры. С помощью специальных обозначений можно перенаправить вывод многих команд в файлы или иные устройства вывода (например, распечатать на принтере). Тоже самое и со вводом информации, вместо ввода данных с клавиатуры, для многих программ можно задать считывание символов их файла. Кроме того, можно даже вывод одной программы передать на ввод другой программе.

К каждой программе, запускаемой в командной строке, по умолчанию подключено три потока данных:

STDIN (0) — стандартный поток ввода (данные, загружаемые в программу). STDOUT (1) — стандартный поток вывода (данные, которые выводит программа). По умолчанию — терминал. STDERR (2) — стандартный поток вывода диагностических и отладочных сообщений (например, сообщениях об ошибках). По умолчанию — терминал.

Pipe (конвеер) – это однонаправленный канал межпроцессного взаимодействия. Термин был придуман Дугласом Макилроем для командной оболочки Unix и назван по аналогии с трубопроводом. Конвейеры чаще всего используются в shell-скриптах для связи нескольких команд путем перенаправления вывода одной команды (stdout) на вход (stdin) последующей, используя символ конвеера ‘|’.

# 4 Выполнение лабораторной работы

Записываю в файл file.txt названия файлов из каталога /etc с помощью перенаправления “>” (записав в него то, что могло быть выведено ls -lR /etc). В файл я включил также все файлы из подкаталогов (рис. 1).

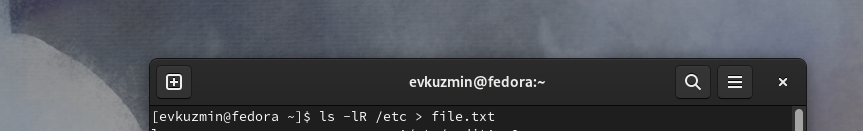


Рис. 1: Запись в файл

Проверяю, что в файл записались нужные значения, делая это с помощью утилиты head: она выводит первые 10 строк файла на экран (рис. 2).

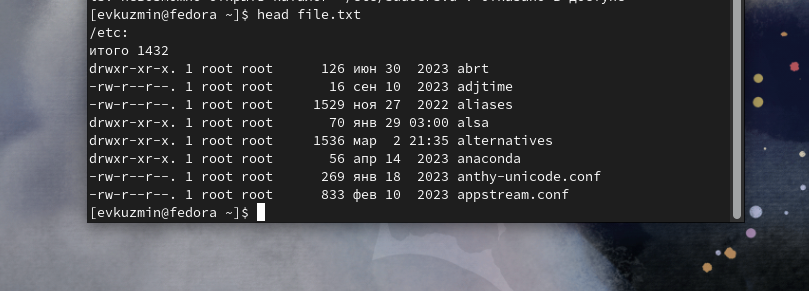


Рис. 2: Вывод содержимого файла

Добавляю в созданный файл имена файлов из домашнего каталога, используя перенаправление “>>” в режиме добавления, вывожу на экран имена всех файлов, имеющих расширение “.conf” с помощью утилиты grep (рис. 3).

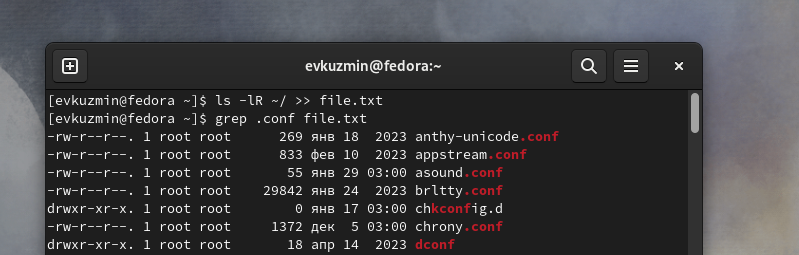


Рис. 3: Добавление данных в файл, поиск файлов определенного расширения

Добавляю вывод прошлой команды в новый файл conf.txt с помощью перенаправления “>” (файл создается при выполнении этой команды). Делаю проверку (рис. 4).

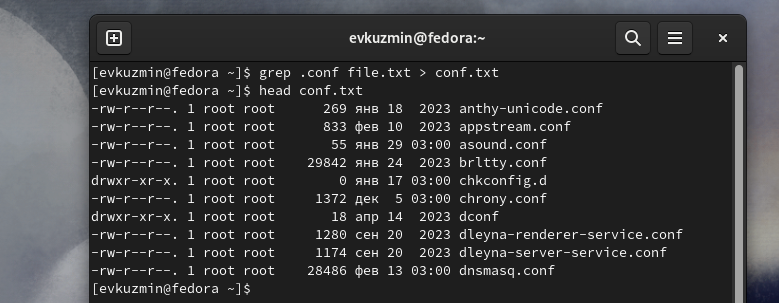


Рис. 4: Запись в файл, проверка

Первый способ состоит в том, чтобы определить, какие файлы в домашнем каталоге начинаются с символа “c”, делая это с помощью утилиты find. Прописываю ей в аргументах домашнюю директорию, выбираю опцию -name, и пишу маску, по которой будем искать имя, где \* - любое кол-во любых символов, Аналогично добавляю опцию -print, чтобы мне вывелся результат (рис. 5)

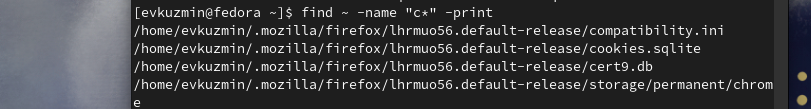


Рис. 5: Поиск файлов, начинающихся с определенного элемента

Второй способ - это использовать утилиту ls -lR и использовать grep, чтобы найти элементы с первым символом c. Однако этот способ не работает для поиска файлов из подкаталогов каталога (рис. 6).

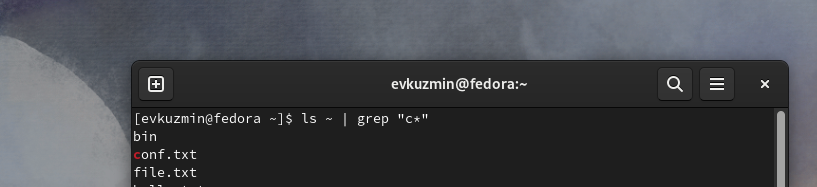


Рис. 6: Поиск файлов, начинающихся с определенного элемента

С помощью метода find ищу все файлы, начинающиеся с буквы “h” (рис. 7).

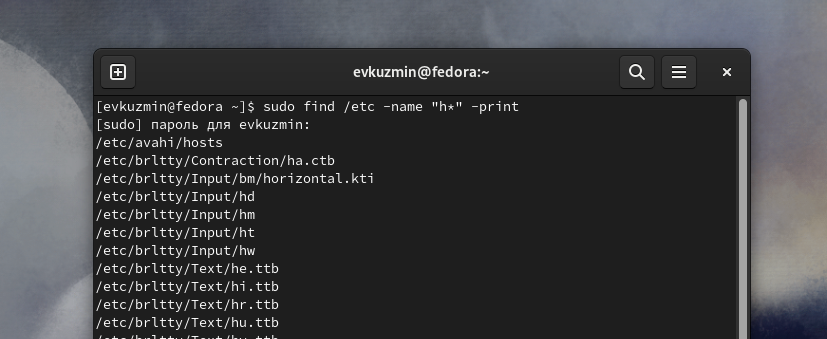


Рис. 7: Поиск файлов, начинающихся с определенного элемента

Запускаю в фоновом режиме, с помощью “&” процесс, который “>” будет записывать в logfile файлы, имена которых начинаются с log. Проверяю, что файл создан, удаляю его, проверяю, что файл удален (рис. 8).

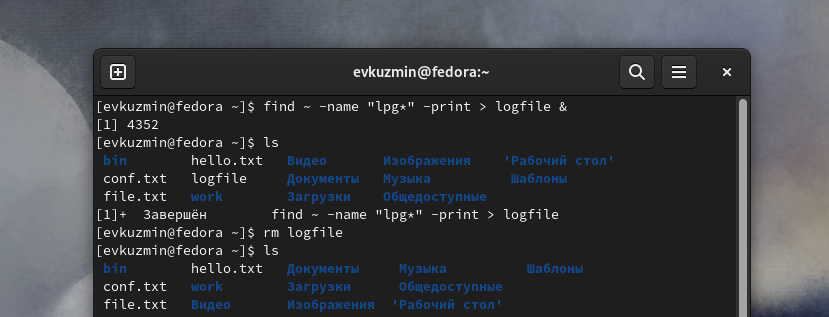


Рис. 8: Создание, запись, удаление фонового процесса

Запускаю в консоли в фоновом режиме редактор gedit. С помощью утилиты ps определяю идентификатор процесса. Также мы можем определить это значение с помощью pgrep, pidof. (рис. 9).

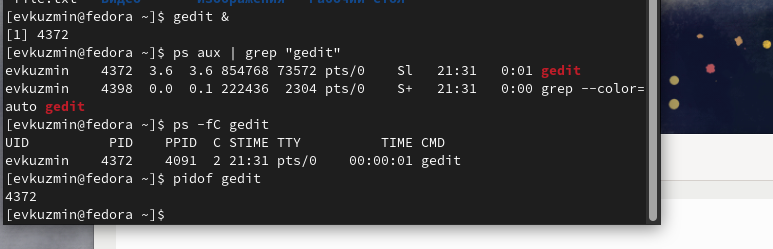


Рис. 9: Создание фонового, поиск идентификатора процесса

Читаю справку команды kill (рис. 10).

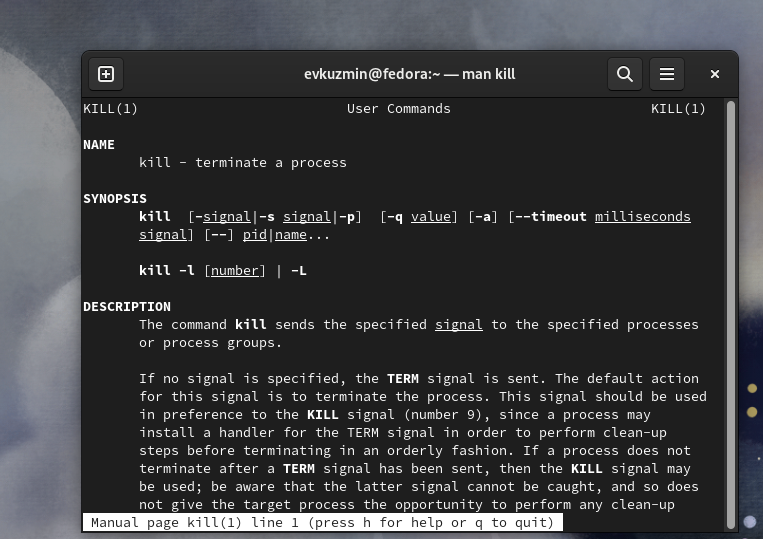


Рис. 10: Чтение документации

Использую команду kill и идентификатор процесса, чтобы его удалить. Процесс gedit закрылся. Далее читаю документацию про функции du и df (рис. 11).



Рис. 11: Удаление процесса, чтение документации

Использую утилиту df с опцией -vi, которая позволяет выяснить, сколько свободного места есть у нашей системы. Использую утилиту du. Она нужна чтобы просмотреть, сколько места занимают файлы в определенной директории (рис. 12).

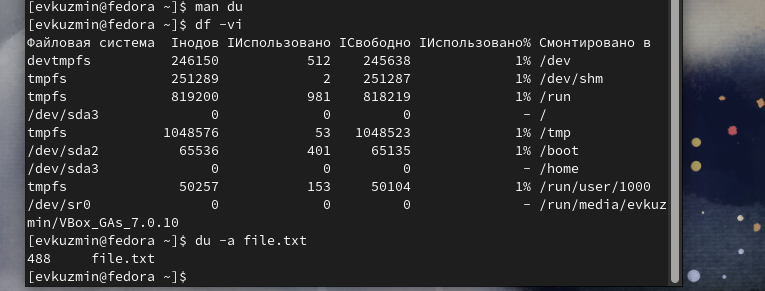


Рис. 12: Утилиты df и du

Читаю документацию о команде find (рис. 13).

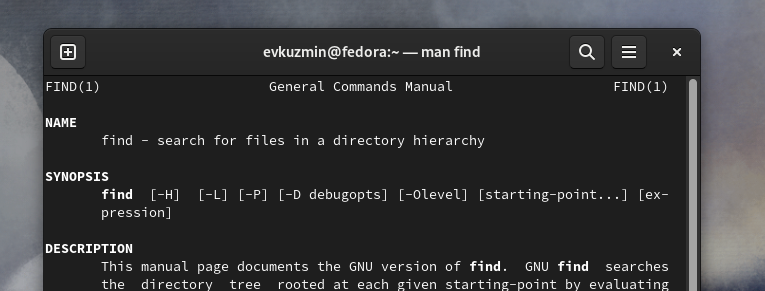


Рис. 13: Чтение документации

Вывожу имена всех директорий, имеющихся в моем домашнем каталоге, используя аргумент d у утилиты find, а также опцию -type, указывающую тип файлов, который мне нужен (рис. 14)

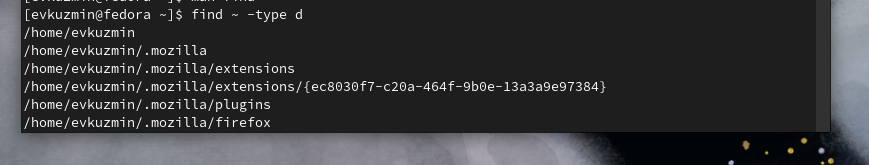


Рис. 14: Вывод содержимого директорий

# 5 Выводы

В результате данной лабораторной работы я ознакомился с инструментами поиска файлов и фильтрации текстовых данных, а также приобрел практические навыки по управлению процессами, по проверке диске и обслуживанию файловых систем

# 6 Ответы на 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.
2. Этот знак > - перенаправление ввода/вывода, а » - перенаправление в режиме добавления.
3. Конвейер (pipe) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей.
4. Главное отличие между программой и процессом заключается в том, что программа - это набор инструкций, который позволяет ЦПУ выполнять определенную задачу, в то время как процесс - это исполняемая программа.
5. PPID - (parent process ID) идентификатор родительского процесса. Процесс может порождать и другие процессы. UID, GID - реальные идентификаторы пользователя и его группы, запустившего данный процесс.
6. Запущенные фоном программы называются задачами (jobs). Ими можно управлять с помощью команды jobs, которая выводит список запущенных в данный момент задач.
7. Команда htop похожа на команду top по выполняемой функции: они обе показывают информацию о процессах в реальном времени, выводят данные о потреблении системных ресурсов и позволяют искать, останавливать и управлять процессами.

У обеих команд есть свои преимущества. Например, в программе htop реализован очень удобный поиск по процессам, а также их фильтрация. В команде top это не так удобно — нужно знать кнопку для вывода функции поиска.

Зато в top можно разделять область окна и выводить информацию о процессах в соответствии с разными настройками. В целом top намного более гибкая в настройке отображения процессов.

1. Назовите и дайте характеристику команде поиска файлов. Приведите примеры использования этой команды.

Команда find - это одна из наиболее важных и часто используемых утилит системы Linux. 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.

Утилита find предустановлена по умолчанию во всех Linux дистрибутивах, поэтому вам не нужно будет устанавливать никаких дополнительных пакетов. Это очень важная находка для тех, кто хочет использовать командную строку наиболее эффективно.

Команда find имеет такой синтаксис: find [папка] [параметры] критерий шаблон [действие] Пример: find /etc -name “p\*” -print

1. find / -type f -exec grep -H ‘текстДляПоиска’ {} ;
2. С помощью команды df -h.
3. С помощью команды du -s.
4. С помощью команды kill номер задачи.

## Список литературы

[Архитектура компьютеров и ОС/Электронный ресурс](https://esystem.rudn.ru/mod/page/view.php?id=1098787)