Отчет по лабораторной работе №8

Дисциплина: Операционные системы

Иванов Сергей Владимирович

Содержание

# 1 Цель работы

Цель данной лабораторной работы - ознакомление с инструментами поиска файлов и фильтрации текстовых данных, а также приобретение практических навыков по управлению процессами (и заданиями), по проверке использования диска и по обслуживанию файловых систем.

# 2 Задание

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

# 3 Выполнение лабораторной работы

Вхожу в систему под соотвествующим именем пользователя, открываю терминал, записываю в файл file.txt названия файлов из каталога /etc с помощью перенаправления “>” . В файл я добавил также все файлы из подкаталогов (рис. 1).

Запись в файл

Рис. 1: Запись в файл

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

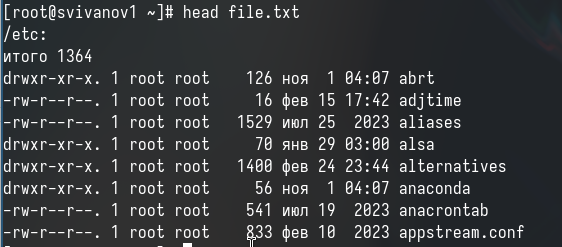


Рис. 2: Вывод содержимого файла

Добавил в созданный файл имена файлов из домашнего каталога, используя перенаправление “>>” в режиме добавления (рис. 3).

Добавление данных в файл

Рис. 3: Добавление данных в файл

Вывожу на экран имена всех файлов, имеющих расширение “.conf” с помощью утилиты grep (рис. 4).

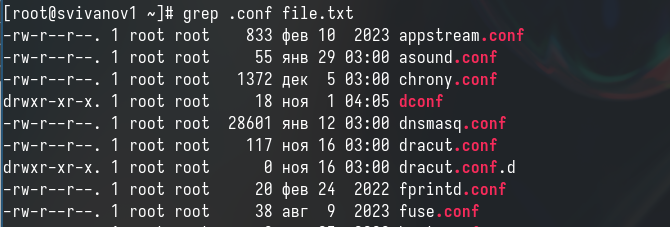


Рис. 4: Поиск файлов определенного расширения

Добавляю вывод прошлой команды в новый файл conf.txt с помощью перенаправления “>” (файл создается при выполнении этой команды) (рис. 5).

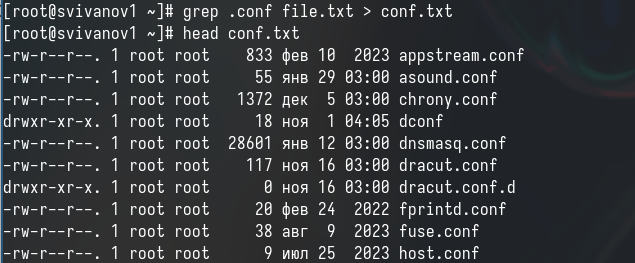


Рис. 5: Запись в файл

Определяю, какие файлы в домашнем каталоге начинаются с символа “c” с помощью утилиты find, прописываю ей в аргументах домашнюю директорию (тогда вывод будет относительно корневого каталога, а не домашнего), выбираю опцию -name (ищем по имени), и пишу маску, по которой будем искать имя, где \* - любое кол-во любых символов, добавляю опцию -print, чтобы мне вывелся результат. Но таким образом я получаю информацию даже о файлах из подкаталогов домашнего каталога. (рис. 6)

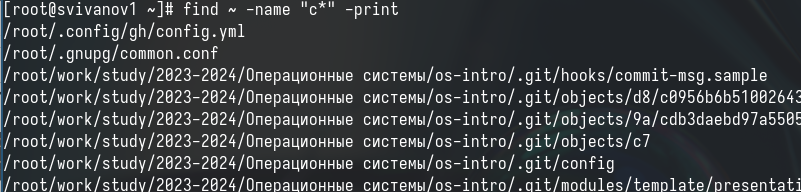


Рис. 6: Поиск файлов, начинающихся с определенного элемента

Второй способ 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. 7).

Поиск файлов, начинающихся с определенного элемента

Рис. 7: Поиск файлов, начинающихся с определенного элемента

С помощью метода find, чьи опции я расписаны ранее, ищу все файлы, начинающиеся с буквы “h” (рис. 8).

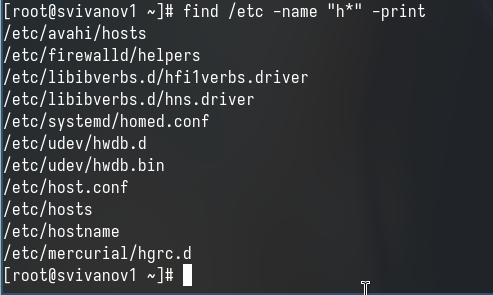


Рис. 8: Поиск файлов, начинающихся с определенного элемента

Запускаю в фоновом режиме (на это указывает символ &) процесс, который будет записывать в файл logfile (с помощью перенаправления >) файлы, имена которых начинаются с log (рис. 9).



Рис. 9: Создание фонового процесса

Проверяю, что файл создан, удаляю его, проверяю, что файл удален (рис 10).

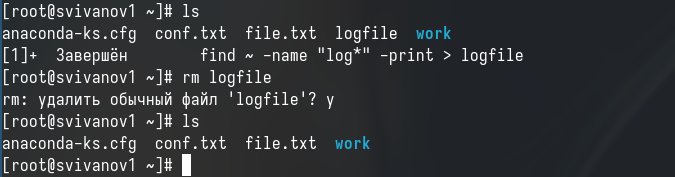


Рис. 10: Удаление файла

Запускаю в консоли в фоновом режиме (с помощью символа &) редактор gedit (рис. 11).

Создание фонового процесса

Рис. 11: Создание фонового процесса

С помощью утилиты ps определяю идентификатор процесса gedit, его значение 3338. Также мы можем определить идентификатор с помощью pgrep. (рис. 12)

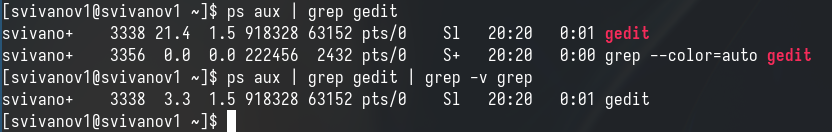


Рис. 12: Поиск идентификатора процесса

Прочитал справку команды kill (рис. 13).

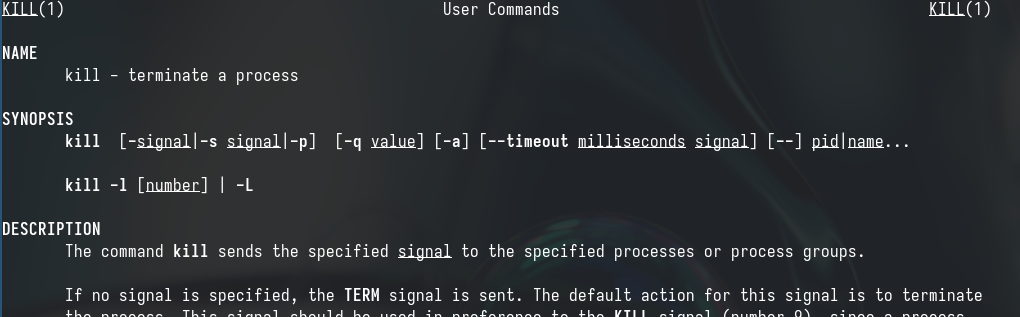


Рис. 13: Чтение документации

Использую команду kill и идентификатор процесса, чтобы его удалить. Замечаем, что закрылась программа gedit. (рис. 14)

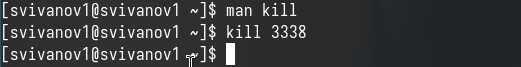


Рис. 14: Удаление процесса

Прочитал документацию про функции df и du (рис. 15).



Рис. 15: Чтение документации

Использую утилиту df опции -iv позволяют увидеть информацию об инодах и сделать вывод читаемым, игнорируя сообщения системы о нем. Эта утилита нам нужна, чтобы выяснить, сколько свободного места есть у нашей системы. (рис. 16)

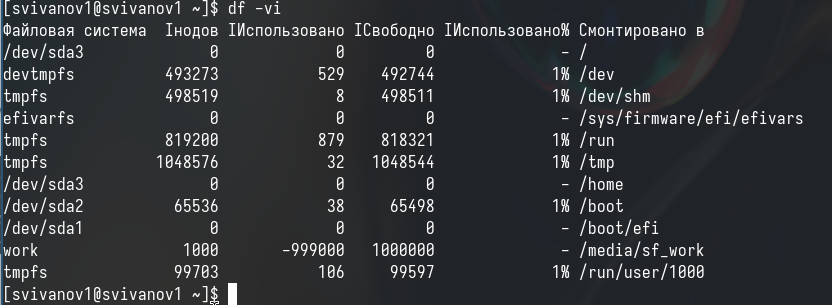


Рис. 16: Утилита df

Использую утилиту du. Она нужна чтобы просмотреть, сколько места занимают файлы в определенной директории и найти самые большие из них (рис. 17).

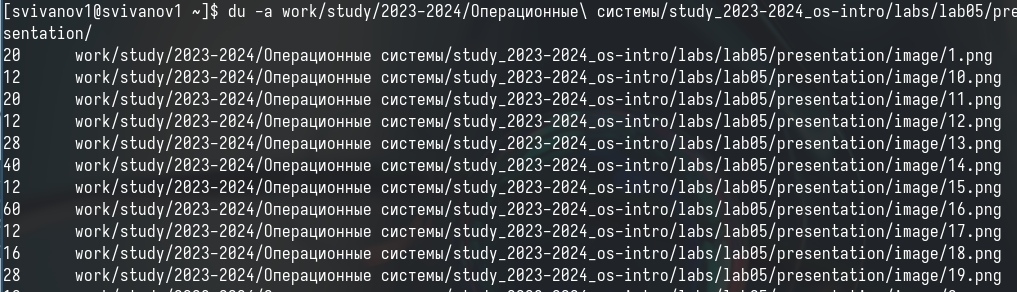


Рис. 17: Утилита du

Прочитал документацию о команде find (рис. 18).

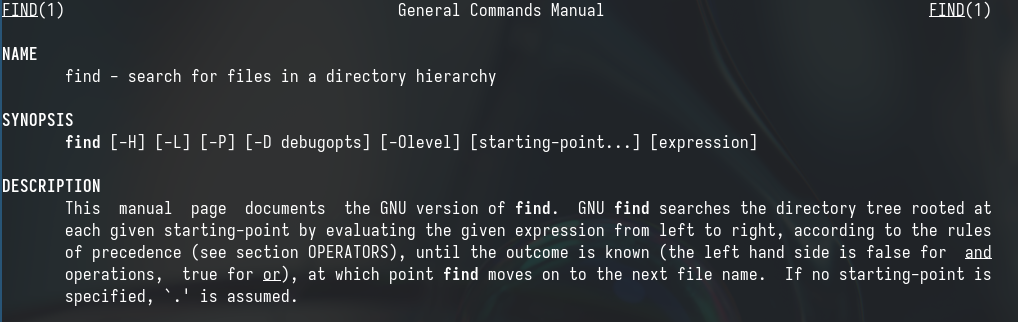


Рис. 18: Чтение документации

Вывожу имена всех директорий, имеющихся в моем домашнем каталоге, используя аргумент d у утилиты find опции -type, то есть указываю тип файлов, который мне нужен и этот тип Директория. Утилита -а позволит увидеть размер всех файлов, а не только директорий. (рис. 19)

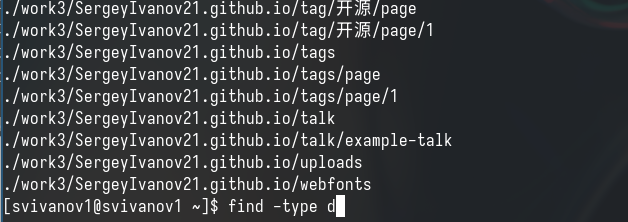


Рис. 19: Вывод директорий

# 4 Контрольные вопросы

1. Какие потоки ввода вывода вы знаете? 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. Объясните разницу между операцией > и ». Этот знак > - перенаправление ввода/вывода, а » - перенаправление в режиме добавления.
3. Что такое конвейер? Конвейер (pipe) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей.
4. Что такое процесс? Чем это понятие отличается от программы? 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. Что такое PID и GID? 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. Что такое задачи и какая команда позволяет ими управлять? Запущенные фоном программы называются задачами (jobs). Ими можно управлять с помощью команды jobs, которая выводит список запущенных в данный момент задач.
7. Найдите информацию об утилитах top и htop. Каковы их функции?

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. Можно ли по контексту (содержанию) найти файл? Если да, то как? find / -type f -exec grep -H ‘текстДляПоиска’ {} ;
2. Как определить объем свободной памяти на жёстком диске? С помощью команды df -h.
3. Как определить объем вашего домашнего каталога? С помощью команды du -s.
4. Как удалить зависший процесс? С помощью команды kill% номер задачи.

# 5 Выводы

В результате выполнения лабораторной работы я ознакомился с инструментами поиска файлов и фильтрации текстовых данных, а также приобрел практические навыки по управлению процессами (и заданиями), по проверке использования диска и по обслуживанию файловых систем.