РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

ОТЧЕТ

по лабораторной работе №7

**«Поиск файлов.**

**Перенаправление ввода-вывода.**

**Просмотр запущенных процессов»**

дисциплина: Операционные системы

Студентка:

Бочкарева Елена Дмитриевна

Студенческий билет номер №: 1032207514

Группа:

НПМбв-01-19

**МОСКВА**

2023

**Содержание:**

Цель работы……………………………………………………………………….4

Задания. Последовательность выполнения лабораторной работы…………….4

Пункт 1………………………………………………………………………….…5

Пункт 2…………………………………………………………………………..6-8

Пункт 3…………………………………………………………………………….9

Пункт 4………………………………………………………………………….…9

Пункт 5…………………………………………………………………………9-10

Пункт 6……………………………………………………………………...…10-11

Пункт 7……………………………………………………………………………11

Пункт 8……………………………………………………………………………12

Пункт 9……………………………………………………………………………13

Пункт 10………………………………………………………………………13-14

Пункт 11………………………………………………………………………14-18

Пункт 12………………………………………………………………………18-20

Ответы на контрольные вопросы……………………………………………21-30

Выводы, согласованные с целью работы……………………………………….31

**Список иллюстраций:**

1 Рисунок 1………………………………………………………………………………………………………..5

2 Рисунок 2………………………………………………………………………………………………………..5

3 Рисунок 3………………………………………………………………………………………………………..6

4 Рисунок 4………………………………………………………………………………………………………..6

5 Рисунок 5………………………………………………………………………………………………………..7

6 Рисунок 6………………………………………………………………………………………………………..7

7 Рисунок 7………………………………………………………………………………………………………..8

8 Рисунок 8………………………………………………………………………………………………………..8

9 Рисунок 9………………………………………………………………………………………………………..9

10 Рисунок 10……………………………………………………………………………………………………..9

11 Рисунок 11……………………………………………………………………………………………………..9

12 Рисунок 12……………………………………………………………………………………………………..9

13 Рисунок 13…………………………………………………………………………………………………….10

14 Рисунок 14…………………………………………………………………………………………………….10

15 Рисунок 15…………………………………………………………………………………………………….11

16 Рисунок 16…………………………………………………………………………………………………….11

17 Рисунок 17…………………………………………………………………………………………………….11

18 Рисунок 18…………………………………………………………………………………………………….12

19 Рисунок 19…………………………………………………………………………………………………….12

20 Рисунок 20…………………………………………………………………………………………………….13

21 Рисунок 21…………………………………………………………………………………………………….13

22 Рисунок 22…………………………………………………………………………………………………….14

23 Рисунок 23…………………………………………………………………………………………………….14

24 Рисунок 24…………………………………………………………………………………………………….14

25 Рисунок 25…………………………………………………………………………………………………….15

26 Рисунок 26…………………………………………………………………………………………………….15

27 Рисунок 27…………………………………………………………………………………………………….16

28 Рисунок 28…………………………………………………………………………………………………….16

29 Рисунок 29…………………………………………………………………………………………………….17

30 Рисунок 30…………………………………………………………………………………………………….17

31 Рисунок 31…………………………………………………………………………………………………….18

32 Рисунок 32…………………………………………………………………………………………………….18

33 Рисунок 33…………………………………………………………………………………………………….18

34 Рисунок 34…………………………………………………………………………………………………….19

35 Рисунок 35…………………………………………………………………………………………………….19

36 Рисунок 36…………………………………………………………………………………………………….20

**Цель работы:** ознакомление с инструментами поиска файлов и фильтрации текстовых данных. Приобретение практических навыков: по управлению процессами (и заданиями), по проверке и использования диска и обслуживанию файловых систем.

**Задания: 4.3. Последовательность выполнения работы.**

1. Осуществите вход в систему, используя соответствующее имя пользователя.

2. Запишите в файл file.txt названия файлов, содержащихся в каталоге /etc. Допишите 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 (постранично) 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. Можно ли определить этот идентификатор более простым способом?

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, имеющихся в вашем домашнем каталоге.

**Результаты выполнения задания:**

**Пункт 1. Осуществите вход в систему, используя соответствующее имя пользователя.**

Вход в систему осуществляю в графическом интерфейсе под именем **пользователя edbochakreva (рис.1).**

* 1. Запускаем операционную систему:

Изображение выглядит как текст

Автоматически созданное описание

Рис.1: Рисунок 1

* 1. Вхожу от имени пользователя edbochkareva. Ввожу пароль (рис.2).

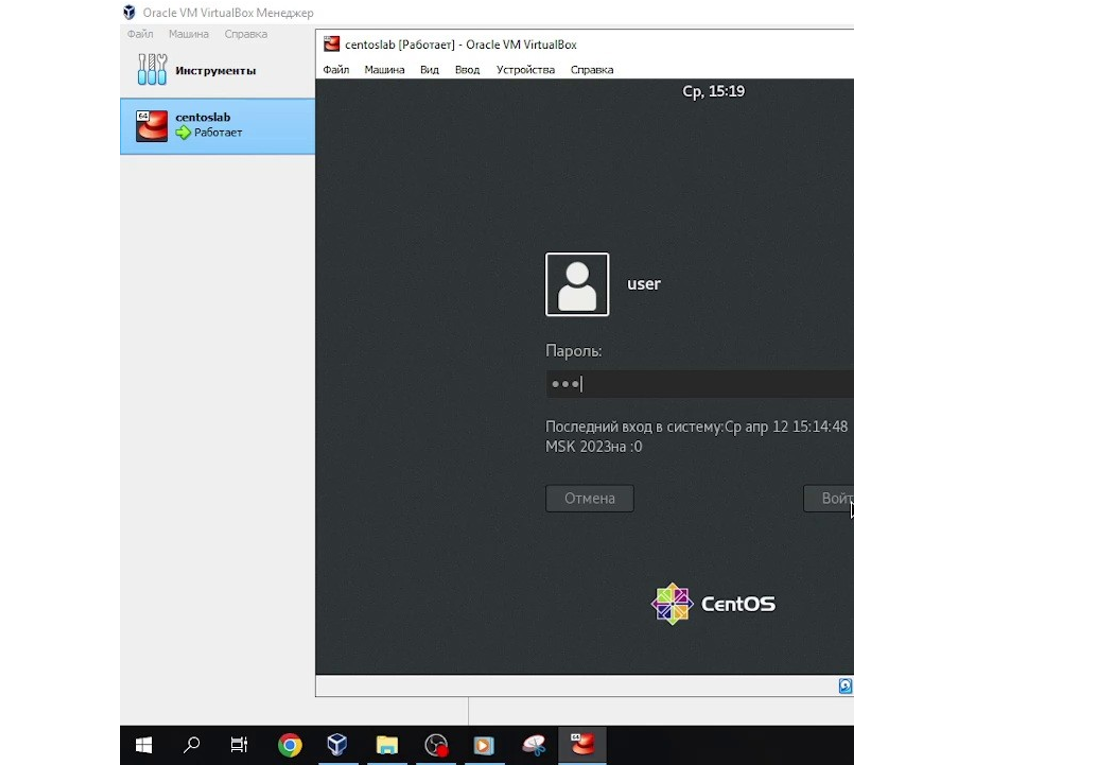


Рис.2: Рисунок 2

**Пункт 2. Запишите в файл file.txt названия файлов, содержащихся в каталоге /etc. Допишите в этот же файл названия файлов, содержащихся в вашем домашнем каталоге.**

Для того, чтобы сохранить список файлов из каталога /etc в файле ﬁle.txt, выполню **команду: ls /etc > file.txt (рис.3).**

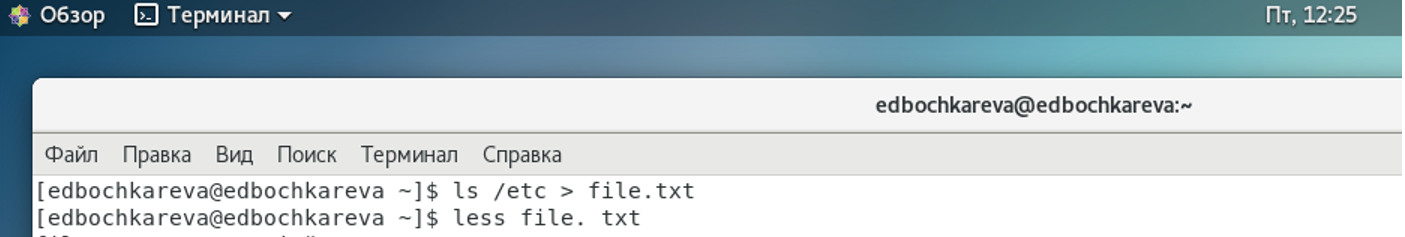


Рис.3: Рисунок 3

Ввод команды для сохранения в файл содержимого **/etc (рис.4).**

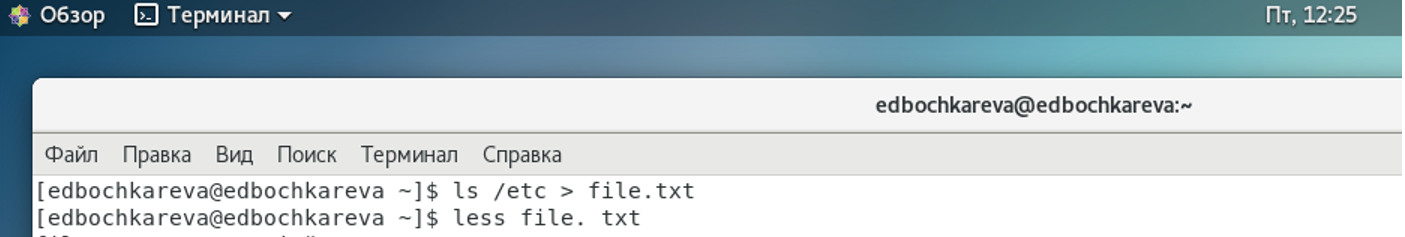


Рис.4: Рисунок 4

Для того, чтобы добавить в файл ﬁle.txt список файлов из домашнего каталога, выполним **команду: ls ~ >> file.txt (рис.5).**

Изображение выглядит как текст

Автоматически созданное описание

Рис.5: Рисунок 5

Осуществляем ввод команды для сохранения в файл содержимого домашнего каталога пользователя.

Поскольку для **перенаправления вывода используется >>,** новые данные будут добавлены в конец файла.

Для просмотра содержимого файла ﬁle.txt выполним **команду less file.txt (рис.6).**

Изображение выглядит как текст

Автоматически созданное описание

Рис.6: Рисунок 6

Осуществляем **проверку содержимого файла ﬁle.txt.**

Просматриваем содержимое файла (рис.7)

Изображение выглядит как текст

Автоматически созданное описание

Рис.7: Рисунок 7

Содержимое файла ﬁle.txt.

Как видно, в конец файла были добавлены имена файлов из текущего каталога пользователя.

Осуществляем проверку выполнения новых команд в конце списка (рис.8).

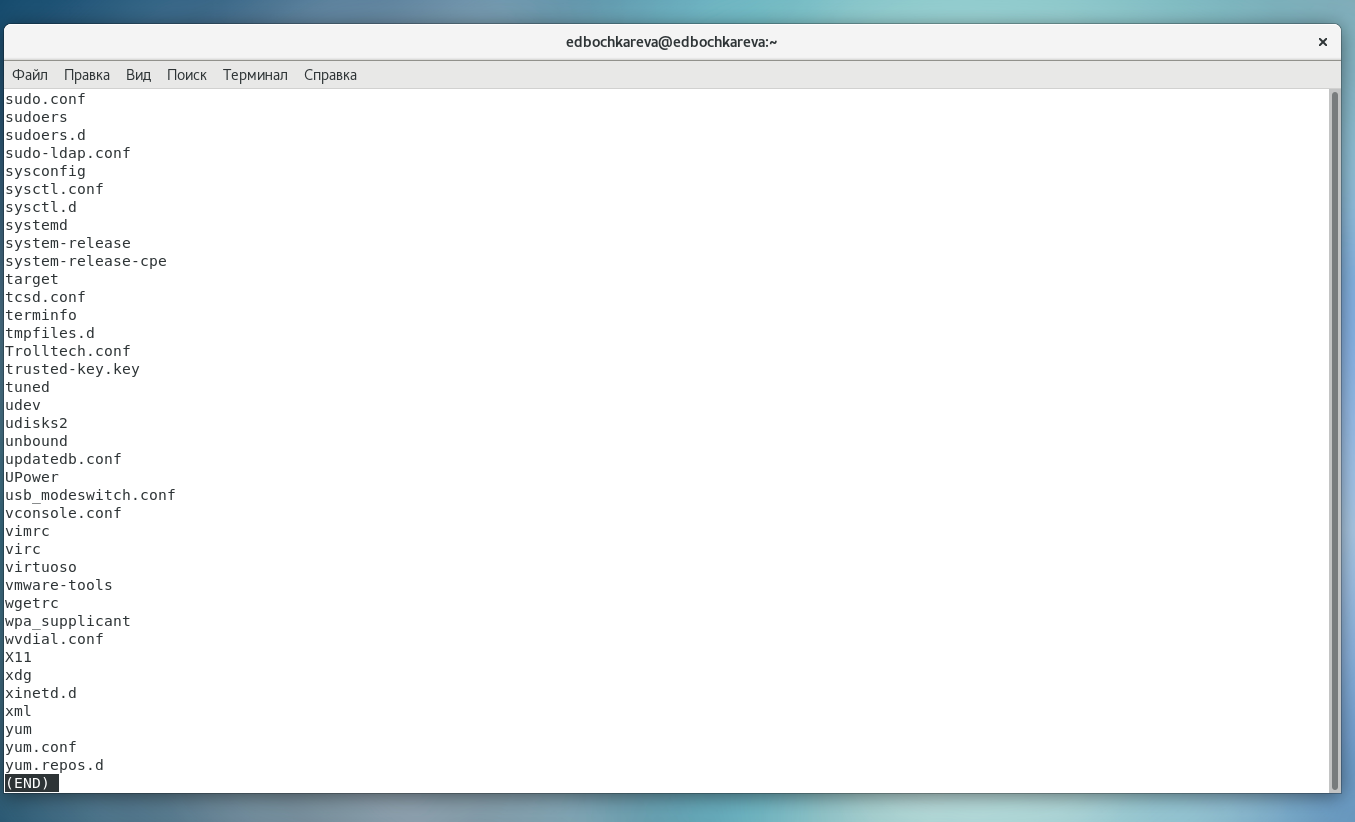


Рис.8: Рисунок 8

**Пункт 3. Выведите имена всех файлов из file.txt, имеющих расширение .conf, после чего запишите их в новый текстовой файл conf.txt.**

В каталоге txt ищем имена файлов, заканчивающихся на **.conf (рис.9).**



Рис.9: Рисунок 9

**Пункт 4. Определите, какие файлы в вашем домашнем каталоге имеют имена, начинавшиеся с символа c? Предложите несколько вариантов, как это сделать.**

Просмотрим имена файлов с помощью команды ls **(рис.10).**



Рис.10: Рисунок 10

Сделаем по аналогии выполненные выше действия при помощи **команды find (рис.11).**



Рис.11: Рисунок 11

**Пункт 5. Выведите на экран (постранично) имена файлов из каталога /etc, начинающиеся с символа h.**

Постранично просматриваем в каталоге etc **имена файлов, начинающихся на h (рис.12).**



Рис.12: Рисунок 12

В данном случае только одна страница таких файлов (рис.13).

Изображение выглядит как текст

Автоматически созданное описание

Рис.13: Рисунок 13

**Пункт 6. Запустите в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log.**

Запустим фоновый процесс, записывающий в файл logfile имена файлов, начинающиеся на log. При этом при первой попытке не будет записано ничего, так как таких файлов нет (рис.14).



Рис.14: Рисунок 14

Введем команду ls -l. Команда ls выдаст список всех файлов из этих каталогов, кроме скрытых файлов (рис.15).

Изображение выглядит как текст

Автоматически созданное описание

Рис. 15: Рисунок 15

При второй попытке ввода ls в данный файл попадет единственное имя: имя этого же файла (рис.16).

Изображение выглядит как текст

Автоматически созданное описание

Рис.16: Рисунок 16

**Пункт 7. Удалите файл ~/logfile.**

Удаляем файл logfile (рис.17).



Рис.17: Рисунок 17

**Пункт 8. Запустите из консоли в фоновом режиме редактор gedit.**

Запускаем из консоли в фоновом режиме редактор gedit (рис 18.).



Рис.18: Рисунок 18

Gedit - свободный текстовый редактор для среды GNOME

Открывается «Безымянный документ»: окно программы gedit, зпаустим командой gedit &. Знак амперсанд означает, что программа будет запущена в фоновом режиме, но это относится к тому, что происходит в самой консоли.

Просматриваем окно «Безымянный документ» (рис.19).

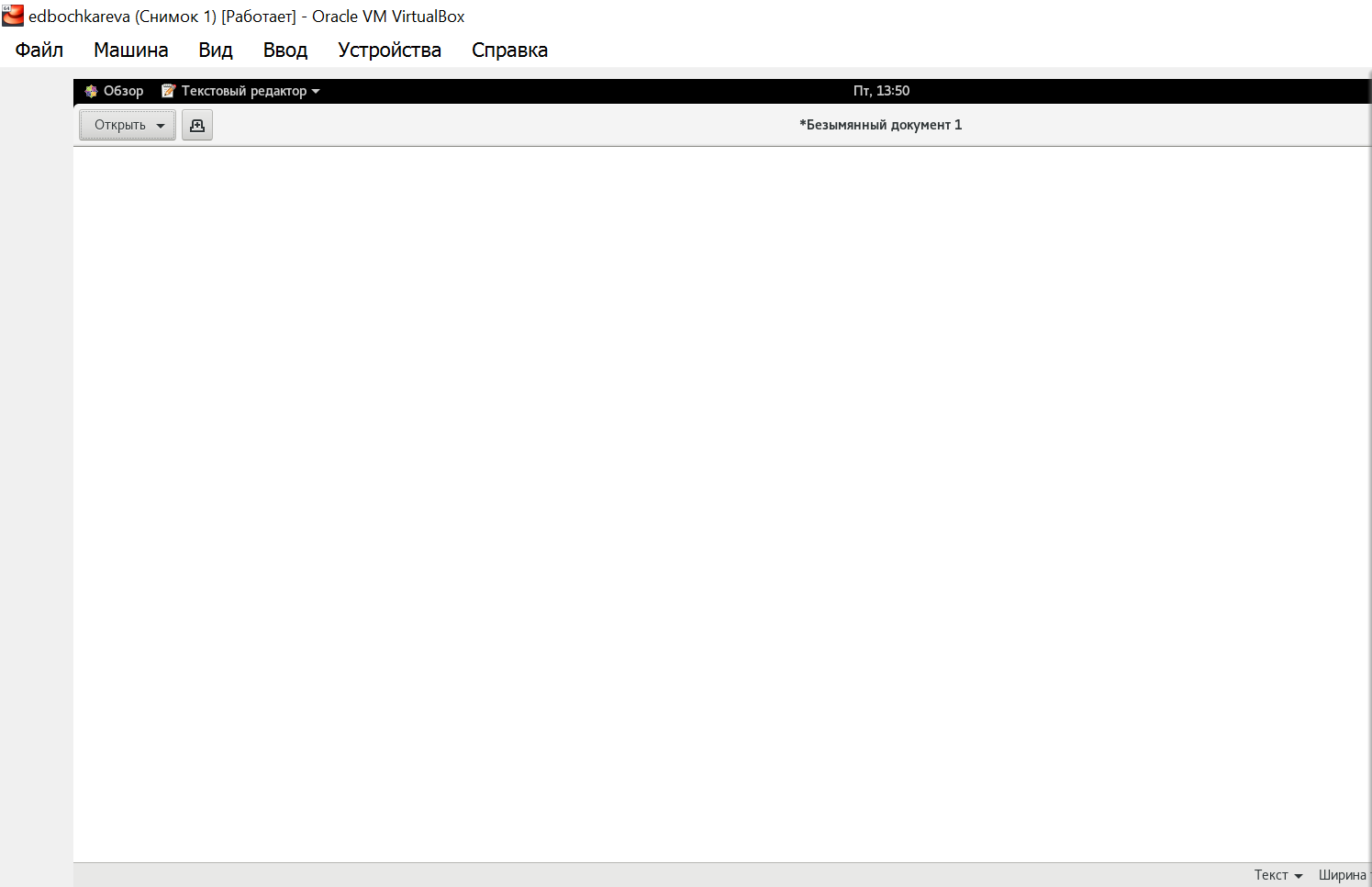


Рис.19: Рисунок 19

**Пункт 9. Определите идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep. Можно ли определить этот идентификатор более простым способом?**

Определяем индентификатор процесса программы GEDIT при помощи команды ps (рис.20).



Рис.20: Рисунок 20

**Команда ps** выводит сведения о процессах в статическом виде.

**Grep** — это команда для поиска внутри текстовых файлов.

Определяем индентификатор процесса программы gedit при помощи **команды pidof (рис.21).**



Рис. 21: Рисунок 21

**Команда pidof** используется для определения идентификаторов PID конкретной запущенной программы.

**Пункт 10. Прочтите справку (man) команды kill, после чего используйте её для завершения процесса gedit.**

Просматриваем **справку по команде man kill (рис.22).**

Изображение выглядит как текст

Автоматически созданное описание

Рис.22: Рисунок 22

**Команда kill** является встроенной командой командной оболочки, предназначенной для отправки системных сигналов определенным процессам. Команда принимает числовые идентификаторы процессов, а также числовые или текстовые идентификаторы сигналов.

**Закрываем gedit при помощи команды kill (рис.23).**



Рис.23: Рисунок 23

**Пункт 11. Выполните команды df и du, предварительно получив более подробную информацию об этих командах, с помощью команды man.**

Выполним **команду df,** предварительно получив более подробную информацию об этой команде, с помощью команды man (рис.24).



Рис.24: Рисунок 24

**df (аббревиатура от disk free)** — утилита в UNIX и UNIX-подобных системах, показывает список всех файловых систем по именам устройств, сообщает их размер, занятое и свободное пространство и точки монтирования:

Получаем более подробную **справочную информацию о команде df (рис.25).**

Изображение выглядит как текст

Автоматически созданное описание

Рис.25: Рисунок 25

Запускаем **команду df** для того, чтобы посмотреть «свободное место» (рис.26).

Изображение выглядит как стол

Автоматически созданное описание

Рис.26: Рисунок 26

Выполним **команду man du,** предварительно получив более подробную информацию об этой команде, с помощью команды man:



**Команда du** позволяет задействовать одноименную утилиту, предназначенную для вывода информации об объеме дискового пространства, занятого файлами и директориями.

**man** (от англ. manual — руководство) — команда Unix, предназначенная для форматирования и вывода справочных страниц.

Просматриваем подробную информацию о **команде du (рис.27)**

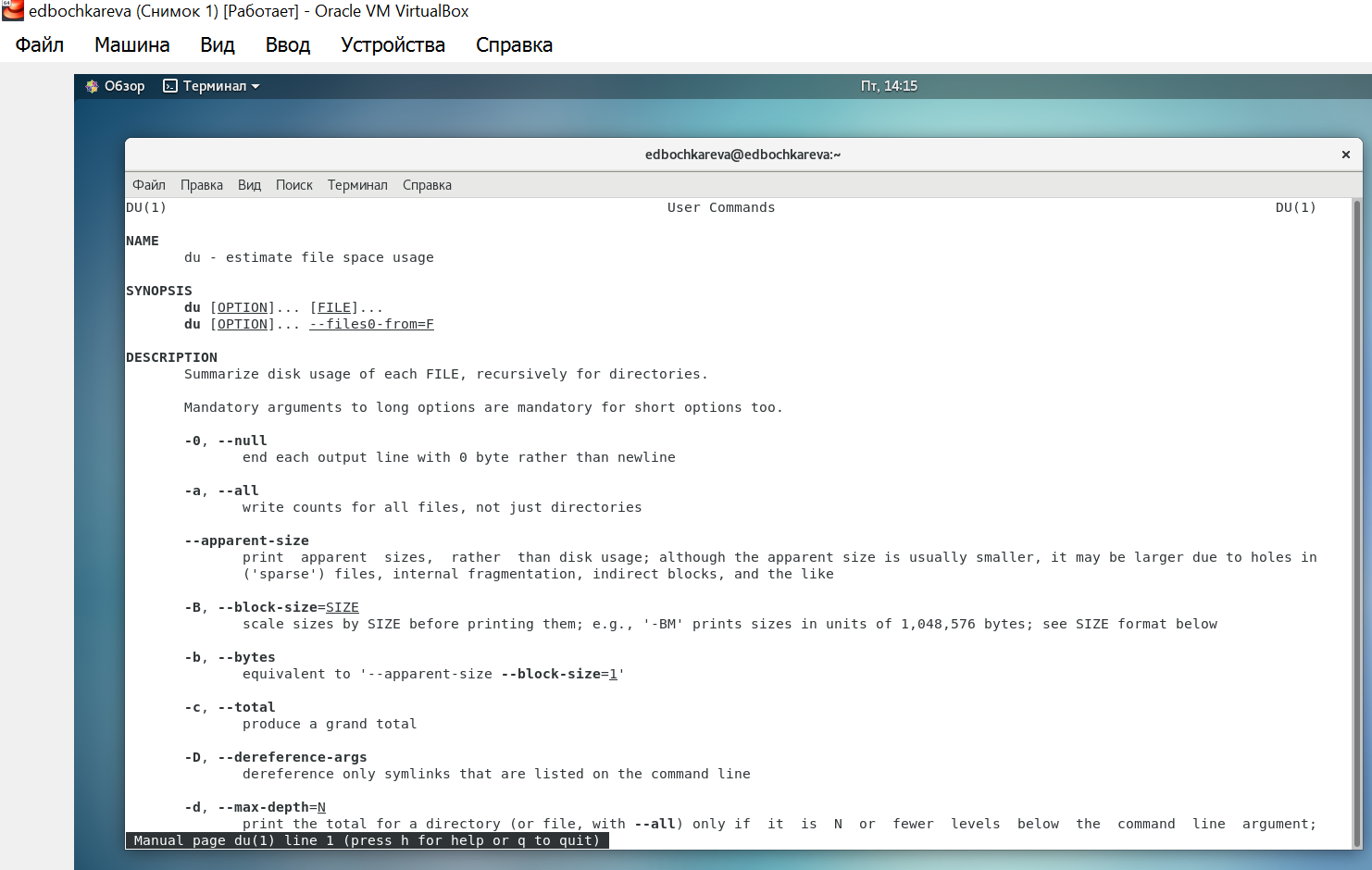


Рис.26: Рисунок 27

При помощи **команды du определим «используемое место» (рис.28)**



Рис.28: Рисунок 28

Просмотр содержимого после выполнения **команды du (рис.29).**

Изображение выглядит как текст

Автоматически созданное описание

Рис.29: Рисунок 29

Определим «используемое место» при помощи **команды du | less (рис.30).**



Рис.30: Рисунок 30

**Команда less** позволяет перематывать текст не только вперёд, но и назад, осуществлять поиск в обоих направлениях, переходить сразу в конец или в начало файла. Особенность less заключается в том, что команда не считывает текст полностью, а загружает его небольшими фрагментами.

Просмотр содержимого после выполнения **команды du | less (рис. 31).**

Изображение выглядит как текст

Автоматически созданное описание

Рис.31: Рисунок 31

Определим используемое место при помощи **команды du -hs (рис.32).**Изображение выглядит как текст

Автоматически созданное описание

Рис.32: Рисунок 32

**Пункт 12. Воспользовавшись справкой команды find, выведите имена всех директорий, имеющихся в вашем домашнем каталоге.**

При помощи **команды find** найдем все каталоги в текущем каталоге (рис.33).



Рис.33: Рисунок 33

Просматриваем выполнение **команды find,** найдем все каталоги в текущем каталоге (рис.34)

Изображение выглядит как текст

Автоматически созданное описание

Рис.34: Рисунок 34

При помощи **команды find** ~` -type d | less найдем все каталоги в текущем каталоге (рис.35).



Рис.35: Рисунок 35

Просматриваем выполнение **команды find** ~` -type d | less до конца списка (рис.36).

Изображение выглядит как текст

Автоматически созданное описание

Рис.36: Рисунок 36

**Ответы на контрольные вопросы:**

## **Какие потоки ввода вывода вы знаете?**

**Ответ:** Существуют потоки стандартного ввода, вывода и вывода ошибок.

Стандартный ввод при работе пользователя в терминале передается через клавиатуру.

Стандартный вывод и стандартная ошибка отображаются на дисплее терминала пользователя в виде текста.

Ввод и вывод распределяется между тремя стандартными потоками:

* stdin — стандартный ввод (клавиатура),
* stdout — стандартный вывод (экран),
* stderr — стандартная ошибка (вывод ошибок на экран).

Потоки также пронумерованы:

* stdin — 0,
* stdout — 1,
* stderr — 2.

Из стандартного ввода команда может только считывать данные, а два других потока могут использоваться только для записи.

Данные выводятся на экран и считываются с клавиатуры, так как стандартные потоки по умолчанию ассоциированы с терминалом пользователя.

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

**Ответ:** Операция > означает, что вывод команды будет записан в файл, прежнее содержимое которого будет уничтожено (если ранее существовало). При внесении изменений в файл, когда вы хотите перезаписать существующие данные, использую оператор «>».

Операция >> дописывает вывод команды в конец файла (или создаёт новый, если такого файла нет). Если я хочу что-то добавить в этот файл, использую оператор «>>».

## Оба оператора являются операторами направления вывода. Основное отличие указано ниже:

* > : Перезаписывает существующий файл или создает файл, если файл с указанным именем отсутствует в каталоге.
* >> : добавляет существующий файл или создает файл, если файл с указанным именем отсутствует в каталоге.

**Примеры:**

$ echo «Добро пожаловать в ElenaEx» > my\_file\_1.txt

$ echo «Добро пожаловать в ElenaEx» >> my\_file\_2.txt

**3.Что такое «Конвейер?»**

**Ответ: Конвейер** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *pipeline*) в терминологии [операционных систем](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) семейства [Unix](https://ru.wikipedia.org/wiki/Unix) — некоторое множество [процессов](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)), для которых выполнено следующее [перенаправление ввода-вывода](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D0%BD%D0%B0%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5)): то, что выводит на поток стандартного вывода предыдущий процесс, попадает в поток стандартного ввода следующего процесса. **Конвейер** - это передача с вывода одной команды на стандартный ввод следующей, соединение стандартного вывода одной команды со стандартным вводом другой.

Осуществляется при помощи символа | между

соответствующими командами, **пример: ls | less**

Можем сделать это, указав команды в нужном порядке и разделив их вертикальной чертой `|' (иногда называемой ``трубой (pipe)'').

**Запуск конвейера реализован с помощью системного вызова pipe().**

[Код возврата](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B4_%D0%B2%D0%BE%D0%B7%D0%B2%D1%80%D0%B0%D1%82%D0%B0) конвейера равен коду возврата последней команды.

В [bash](https://ru.wikipedia.org/wiki/Bash" \o "Bash) можно изменить это поведение, включив опцию pipefail:

**set -o pipefail**

## **Что такое процесс? Чем это понятие отличается от программы?**

Процесс - это запущенная программа. При выполнении одной программы может быть запущено несколько принадлежащих к ней процессов.

Термин "процесс" впервые появился при разработке операционной системы Multix и имеет несколько определений, которые используются в зависимости от контекста.

Процесс - это:

1. программа на стадии выполнения
2. "объект", которому выделено процессорное время
3. асинхронная работа

Для описания состояний процессов используется несколько моделей.

Самая простая модель - это модель трех состояний. Модель состоит из:

1. состояния выполнения
2. состояния ожидания
3. состояния готовности

## **5.Что такое PID и GID?**

**PID** - **это идентификатор процесса**.

**GID - идентификатор группы.**

**PID -** идентификатор процесса PPID - идентификатор процесса, породившего данный UID и GID - идентификаторы прав процесса (соответствует UID и GID пользователя, от которого запущен процесс).

**PID обозначает идентификационный номер процесса**, который обычно используется большинством ядер операционной системы, таких как Linux, Unix, macOS и Windows. Это уникальный идентификационный номер, который автоматически присваивается каждому процессу, когда он создается в операционной системе.

**Идентификатор группы (GID)**

Кроме идентификационного номера пользователя с учётной записью связан идентификатор группы. Группы пользователей применяются для организации доступа нескольких пользователей к некоторым ресурсам. У группы, так же, как и у пользователя, есть имя и **идентификационный номер — GID (Group ID).**

## **6.Что такое задачи и какая команда позволяет ими управлять?**

**Задача** - это программа, запущенная в фоне.

Для управления задачами используется программа jobs: **jobs** — **просмотр списка собственных задач (процессов).**

Запущенные из консоли с помощью амперсанда команды, работают в фоновом режиме и называются ***задачами*** (**jobs**). Можно сказать, что ***задачи*** это процессы, привязанные к командному интерпретатору. Такие задачи помимо традиционного **PID** имеют еще свою нумерацию начинающуюся с единицы.

Просмотреть запущенные ***задачи*** интерпретатора, можно командой **jobs**.

**В примере** ниже показана ситуация когда есть две ***задачи*** и выполнение одного из них остановлено:

|  |  |
| --- | --- |
| 1 2 3 | $ jobs [1]+  Stopped                 top [2]-  Running                 sleep 100 & |

## **7.Найдите информацию об утилитах top и htop . Каковы их функции?**

**Утилиты top и htop используются для просмотра списка запущенных процессов.**

Функции **утилит top и htop:** в частности они позволяют сортировать процессы по потреблению ресурсов процессора и памяти, что позволяет, например, находить процессы, которые потребляют значительную часть ресурсов системы.

**Команда top.** Она немного проще чем та же утилита htop, но в отличие от ps позволяет выводить информацию о системе, а также список процессов динамически обновляя информацию о потребляемых ими ресурсах. Утилита не всегда установлена по умолчанию, для её установки в Ubuntu используйте команду:

**sudo apt install top**

Затем для запуска просто выполните в терминале:

**top**

**HTOP - монитор процессов. htop**— продвинутый монитор процессов, написанный для Linux. Он был задуман заменить стандартную программу top. Htop показывает динамический список системных процессов, список обычно выравнивается по использованию ЦПУ. В отличие от top, htop показывает все процессы в системе. Также показывает время непрерывной работы, использование процессоров и памяти. Htop часто применяется в тех случаях, когда информации даваемой утилитой top недостаточно, например при поиске утечек памяти в процессах.

## Установка:

**sudo apt-get install htop**  
  
Запускаем:  
  
**sudo htop**

## **8.Назовите и дайте характеристику команде поиска файлов. Приведите примеры использования этой команды.**

Для поиска файлов может использоваться **программа ﬁnd.**

Пример использования программы, для поиска в текущем каталоге файлов, которые содержат в назывании слово user в любом регистре:

**find . -iname '\*user\*'**

Рассмотрим несколько наиболее популярных способов поиска файлов в Linux, используя терминал.

**find:** для поиска файлов из командной строки вы можете использовать команду “find”.

У этой команды следующий синтаксис:

***“path”*** **-** Секция для указания директории поиска. Если ничего не указано поиск идет по текущей директории.

***“criteria”*** **-** Опции поиска.

***“action”*** **-**Опции, которые влияют на состояние поиска или контролируют его, например,

“–print”

Изображение выглядит как текст

Автоматически созданное описание

## **Можно ли по контексту (содержанию) найти файл? Если да, то как?**

Для поиска по содержимому может использоваться **команда grep,** которой передаётся, первым параметром, строка для поиска, а далее имена файлов, в которых нужно искать.

Например, для поиска строки test в файлах в текущем каталоге можено использовать такую команду:

**grep "test" ./\***

Для поиска файла по содержимому проще всего воспользоваться **командой grep (вместо find).**

**Пример:**

**grep -r строка\_поиска каталог**

или:

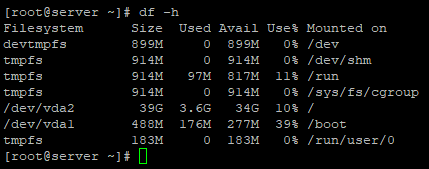
**grep -lir 'class List' /home/balancer/programming/java/jbforth**

## **Как определить объем свободной памяти на жёстком диске?**

Для того, чтобы определить объём памяти на диске, можно **воспользоваться командой df,** в качестве параметра можно передать путь, тогда будет выведено свободное место на соответствующем разделе (путь не обязательно должен указывать на точку монтирования раздела).

**df**– это команда позволяет отобразить информацию о свободном/доступном месте на диске, файловой системы раздела.Чтобы листинг команды был более читабелен, нужно использовать ее с опциями. Например:

**# df -h**



Описание столбцов:

* Filesystems – имя файловой системы
* Size – размер раздела
* Used – используемое дисковое пространство
* Avail – доступное дисковое пространство(свободное)
* Use% — занятое дисковое пространство в процентах
* Mounted on – смонтировано. Указывает директорию, к которой примонтирован раздел.

**При использовании опции -h дисковое пространство выводится в Гб. Если размер меньше 1Гб, то вывод будет в Мб.**

**Дополнительные опции:**

**df -m** – информация будет отображена в Мб;

**df -k** — информация будет отображена в Кб;

**df -T** – к выводу добавиться тип файловой системы.

## **Как определить объем вашего домашнего каталога?**

В операционных системах на базе Linux **посмотреть размер папки (директории) можно с помощью команды du.**

Эта команда, выполняемая в консоли, позволяет оценить используемый объем места на жестком диске отдельно по папкам и файлам, просуммировать результат, узнать общий размер папки.

Чтобы определить объём домашнего каталога, можно выполнить следующую команду:

**du -hs ~**

**Общий синтаксис команды du следующий:**du  
du имяПапки  
du [ключи] имяПапки

**Пример.** Без передачи каких-либо параметров команда du выводит название и размер каждой папки по текущему пути, при этом включая в вывод все подкаталоги:

Изображение выглядит как текст

Автоматически созданное описание

**Ключ -h** обеспечит вывод в удобном для восприятия человеком виде, а, благодаря ключу -s, будет выведен суммарный объём всего каталога.

## **Как удалить зависший процесс?**

Для того, чтобы удалить зависший процесс, необходимо, зная его идентификатор, выполнить команду kill с ключом -9 и идентификатором процесса, переданном в качестве следующего.

Для завершения процесса нужно вызвать утилиту kill с параметром "-9".

Когда известен PID процесса, мы можем убить его **командой kill**.

Команда kill принимает в качестве параметра PID процесса.

Например, убьем процесс с номером 25609:

kill 25609

# Выводы, согласованные с целью работы:

В процессе выполнения лабораторной работы были использованы приложения, предназначенные для поиска файлов (ﬁnd, ls), фильтрации данных (grep), управлению процессами (ps, pidof, kill) и проверке использования диска (команды df, du).

Данные приложения позволяют управлять работой системы в консоли, что, в свою очередь, позволяет администрировать систему, подключаясь к ней без использования графического интерфейса.

Вышеописанные действия администрируют систему, в том числе используя относительно медленные соединения, где работа используя графический интерфейс может быть затруднена или некомфортна.