|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА - Российский технологический университет»**  **РТУ МИРЭА** |

Институт Информационных Технологий

Кафедра Вычислительной Техники (ВТ)

**ОТЧЁТ ПО ПРАКТИЧЕСККОЙ РАБОТЕ № 5-7**

по дисциплине

«Архитектура вычислительных машин и систем»

|  |  |
| --- | --- |
| Выполнил студент группы  ИКБО-20-21 | Сидоров С.Д. |
| Принял ассистент кафедры ВТ | Кузнецова А.Л. |

**Оглавление**

[Практическая работа 1 3](#_Toc26340)

[Выводы 7](#_Toc27742)

[Практическая работа 2 8](#_Toc405)

[Выводы 16](#_Toc4141)

[Практическая работа 3 17](#_Toc31772)

[Выводы 22](#_Toc10715)

# Практическая работа 1

**Виртуализация: установка и настройка операционной системы**

**UbuntuLinux на виртуальной машине OracleVirtualBox**

**Цель работы:**

Целью данной практической работы является получение практических навыков установки и создания виртуальных машин в Oracle VirtualBox, а также изучение принципов инсталляции и начальной настройки операционной системы Ubuntu.

**Задание:**

Для выполнения практической работы необходимо скачать с официального сайта компании Oracle дистрибутив виртуальной машины VirtualBox и выполнить установку скачанного дистрибутива на компьютер. После установки необходимо с помощью инструментов, предоставляемых VirtualBox создать и настроить виртуальную машину, и установить на нее операционную систему UbuntuLinux. Дистрибутив для установки необходимо скачать из интернета. В процессе создания виртуальной машины необходимо определить расположение файлов виртуальной машины на компьютере, выделить объем оперативной памяти, видеопамяти, жесткого диска, необходимых для функционирования устанавливаемой операционной системы. Задать количество ядер центрального процессора, используемых виртуальной машиной и предельный уровень загрузки процессора. При установке операционной системы необходимо задать способ разбиения жесткого диска на логические разделы.

**Ход работы:**

Задание оригинальное имя, которое сопоставимо с ФИО студента, новой виртуальной машине (Рисунок 1).

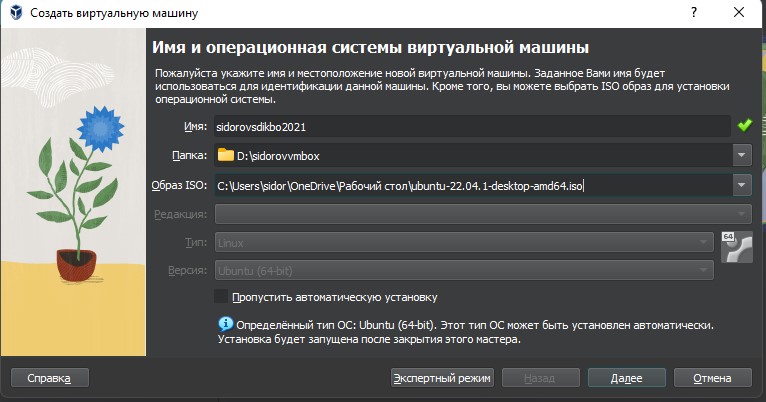
****

Рисунок 1 – Задание имени виртуальной машине

Добавление оборудования (Рисунок 3) и виртуального жёсткого диска (Рисунок 4)

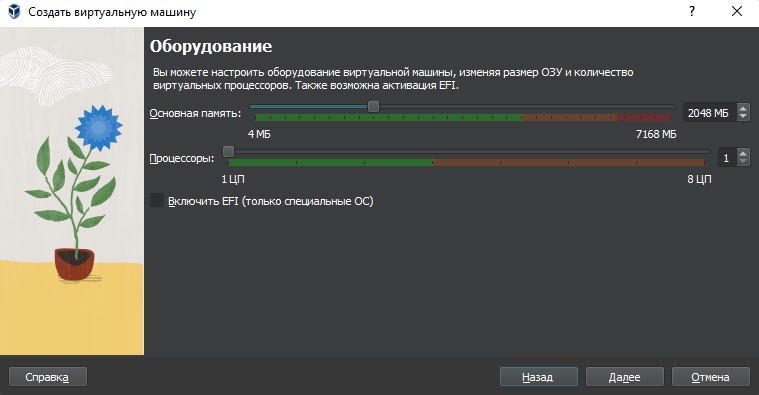
****

Рисунок 3 – Оборудование

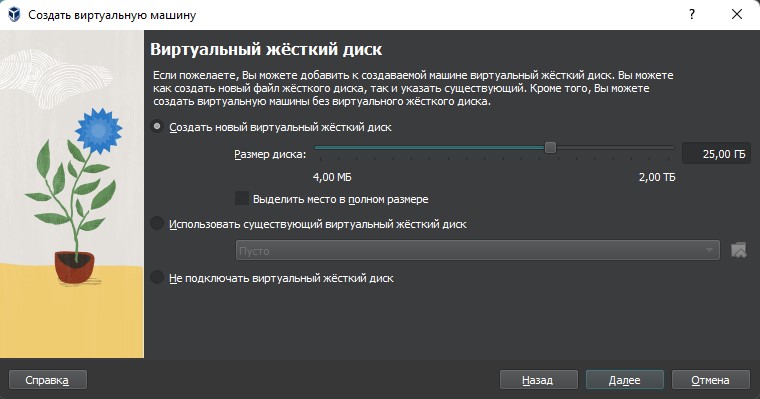
****

Рисунок 4 – Виртуальный жёсткий диск

Итог создания виртуальной машины изображён на рисунке 5.

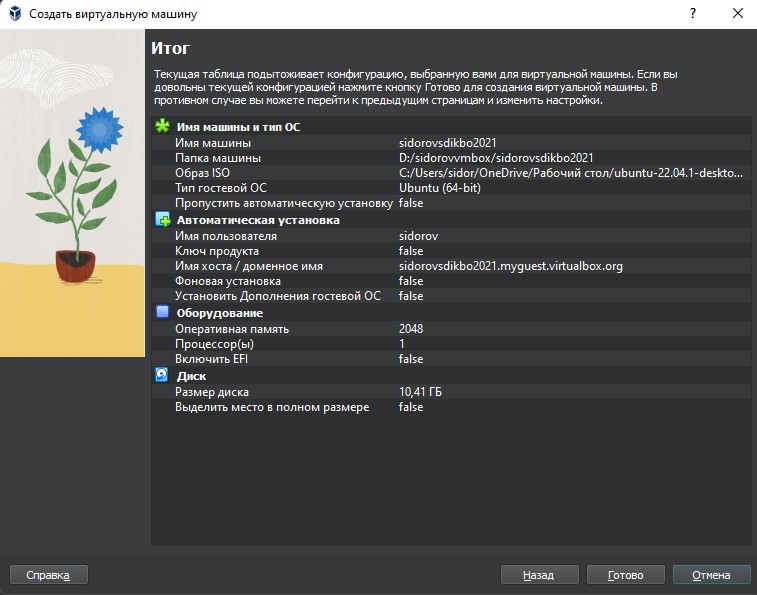


Рисунок 5 – Итог

Статус виртуальной машины изображен на рисунке 6.

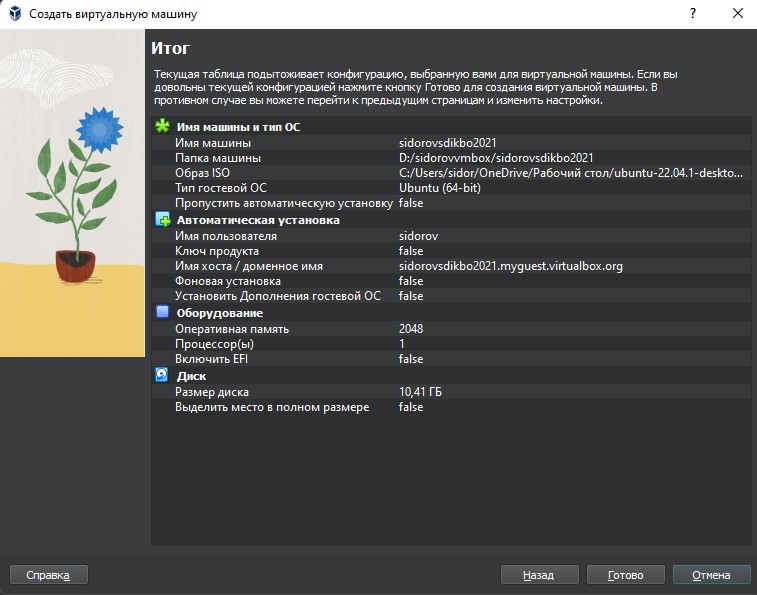
****

Рисунок 6 – Статус виртуальной машины

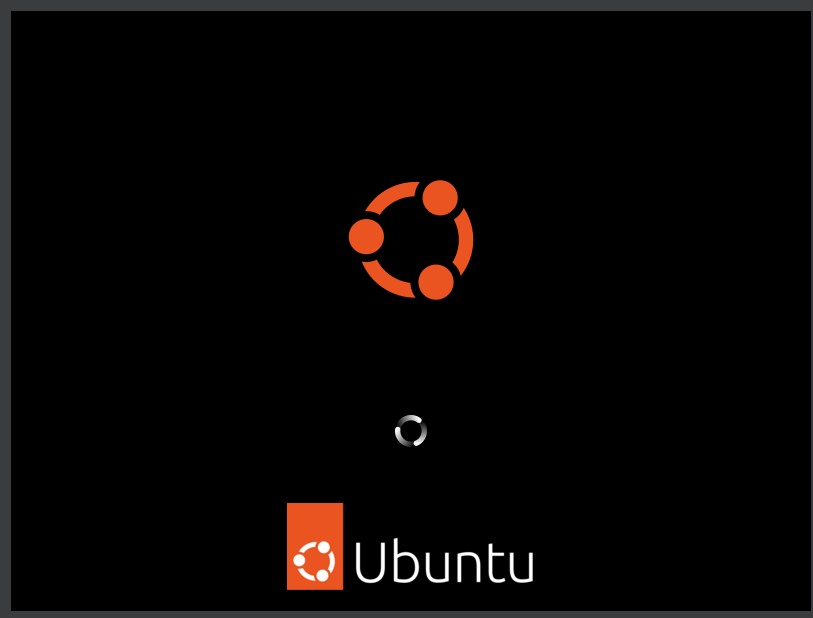


Рисунок 7 - Результат запуска виртуальной машины

## Выводы

В ходе выполнения практической работы были получены практические навыки создания виртуальных машин и начальной настройки и установки операционной системы Ubuntu Linux.

# Практическая работа 2

**Файловая система: изучение команд работы с файлами и каталогами**

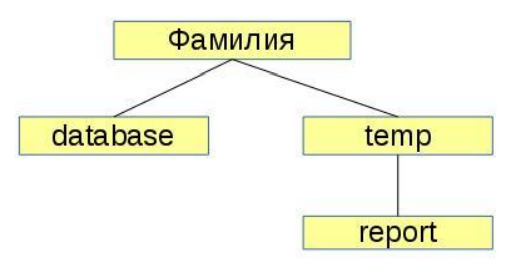
**Вариант № 21**

**Цель работы:**

Целью данной практической работы является изучение команд операционной системы GNU Linux по работе с элементами файловой системы, а также получение практических навыков создания, изменения, манипулирования и удаления файлов и каталогов.

**Задание:**

1. Все этапы выполнения работы необходимо фиксировать.
2. Войти в систему от имени своей учетной записи. В случае, если вход осуществлен в графическую оболочку, переключиться на текстовую консоль или запустить терминал. Вся дальнейшая работа выполняется исключительно в терминале. Использование root прав запрещено.
3. Создать родительский каталог. В качестве имени каталога задать свою фамилию. Все остальные действия данной лабораторной работы будут выполняться внутри данного каталога.
4. Внутри каталога, созданного на 2-м шаге создать структуру каталогов, представленную на рисунке. Вывести на экран содержимое текущего каталога и убедиться, что все созданные каталоги созданы без ошибок. Для отображения используйте утилиту tree. При необходимости произведите обновление компонентов и установите утилиту tree вручную.



1. Перейти в каталог temp. Убедиться, что он является текущим. Вывести на экран содержимое каталога.
2. С помощью встроенного текстового редактора внутри каталога temp создать файл базы данных dataset1.txt. Заполнить файл данными в соответствии с номером варианта задания. В качестве разделителя столбцов данных в файле использовать символ “;” без пробелов. Файл должен содержать не менее 3-х строк.
3. С помощью конвеера команд внутри каталога temp создать файл базы данных dataset2.txt. Заполнить файл данными в соответствии с номером варианта задания. В качестве разделителя столбцов данных в файле использовать символ “;” без пробелов. Файл должен содержать не менее 4-х строк. Данные должны отличаться от введенных ранее.
4. С помощью перенаправления вывода в файл, либо используя команду echo создать файл базы данных dataset3.txt. Заполнить файл данными в соответствии с номером варианта задания. В качестве разделителя столбцов данных в файле использовать символ “;” без пробелов. Файл должен содержать не менее 3-х строк. Данные должны отличаться от введенных ранее.
5. Вывести на экран содержимое всех созданных файлов базы данных.
6. Объединить содержимое всех созданных файлов базы данных в один файл data.txt и поместить его в каталог /database.
7. Перейти в каталог /database. Убедиться, что он является текущим. Вывести на экран содержимое каталога. Убедиться, что созданный файл data.txt содержит все необходимые данные.
8. Подсчитать количество строк файла data.txt. Результат подсчета вывести на экран и в файл отчета output.txt, расположенный в каталоге report (см. рисунок).
9. С помощью любого из использованных выше способов дополнить файл data.txt 2-я строками данных в соответствии с номером варианта задания. В качестве разделителя столбцов данных в файле использовать символ “;” без пробелов. Убедиться, что файл data.txt содержит все необходимые данные.
10. Повторно подсчитать количество строк файла data.txt. Результат подсчета вывести на экран и дописать в конец файла отчета output.txt, расположенного в каталоге report.
11. Осуществить фильтрацию данных файла data.txt в соответствии с номером варианта задания. Результат фильтрации вывести на экран и в файл отчета filtered.txt, расположенный в каталоге report. Повторить фильтрацию с различными значениями фильтра. Результаты фильтрации выводить на экран и дописывать в файл отчета filtered.txt.
12. Выполнить сортировку содержимого файла data.txt в соответствии с номером варианта задания. Результат сортировки вывести на экран и в файл отчета sorted.txt, расположенный в каталоге report.
13. Выполнить фильтрацию содержимого файла data.txt с сортировкой результата фильтрации. Фильтрацию и сортировку выполнить в соответствии с номером варианта задания. Результат вывести на экран и в файл отчета filteredsorted.txt, расположенный в каталоге report.
14. Исследовать самостоятельно команды: date, cal, pwd, who, clear, exit.
15. Выполнить команду вывода календаря на экран и любым известным способом записать значение в файл calendar.txt, находящийся в каталоге /database. Результат вывести на экран.

**Ход работы:**

Создание родительского каталога и структуры каталогов изображено на рисунке 8.

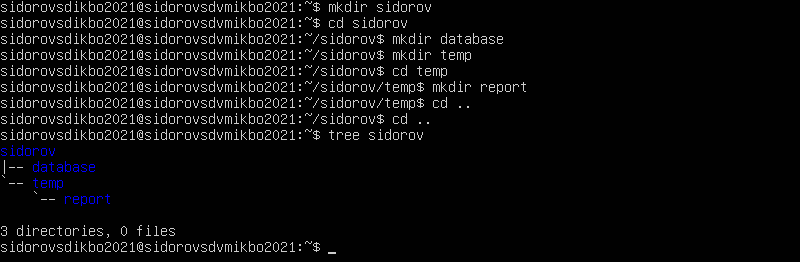


Рисунок 8 – Создание структуры каталогов.

Отображение каталогов изображено на рисунке 9.

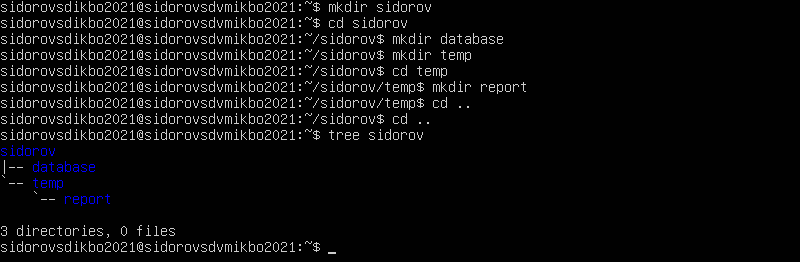


Рисунок 9 – Отображение каталогов

Создание файла базы данных изображено на рисунке 10.

task6_dataset1_viopen

Рисунок 10 – Создание файла базы данных

Заполнение файла базы данных изображено на рисунке 11.

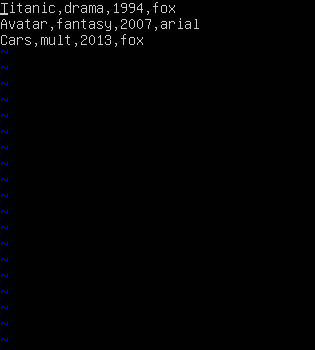


Рисунок 11 – Заполнение файла базы данных с помощью vim

Заполнение базы данных с помощью конвейера команд представлено на рисунке 12.

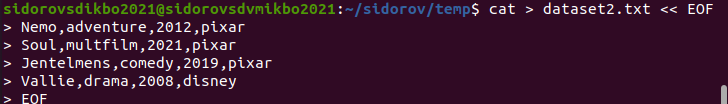


Рисунок 12 - Заполнение файла с помощью конвейера команд.

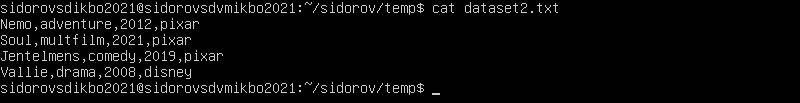


Рисунок 13 - Заполненный файл с помощью конвейера команд.

Заполнение данных с помощью перенаправления вывода в файл и команды echo представлено на рисунке 14.

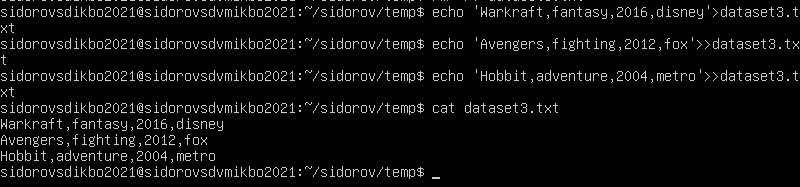


Рисунок 14 - Заполнение файла с помощью перенаправления вывода в файл

Содержимое всех файлов баз изображено на рисунке 15.



Рисунок 15 - Содержимое всех созданных файлов

Объединение содержимого в один файл data.txt изображено на рисунке 16.

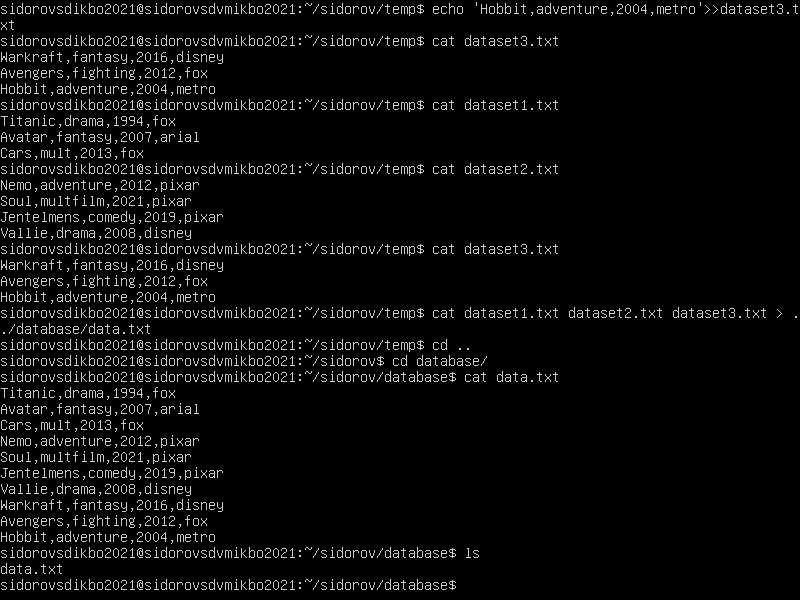


Рисунок 16 - Объединение файлов

Подсчёт строк в файле data.txt представлен на рисунке 17.

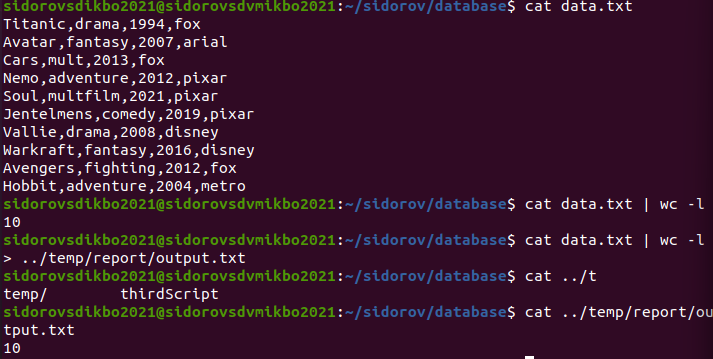


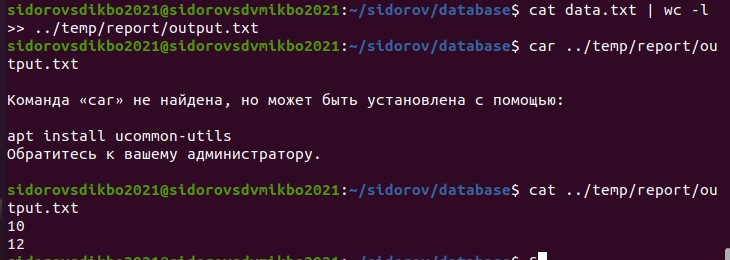
Рисунок 17 - Подсчёт количества строк в файле.

Добавление новых записей в файл data.txt изображено на рисунке 18.



Рисунок 18 - Добавление двух новых записей

Подсчёт количества строк в файле data.txt с двумя добавленными строками изображена на рисунке 19.



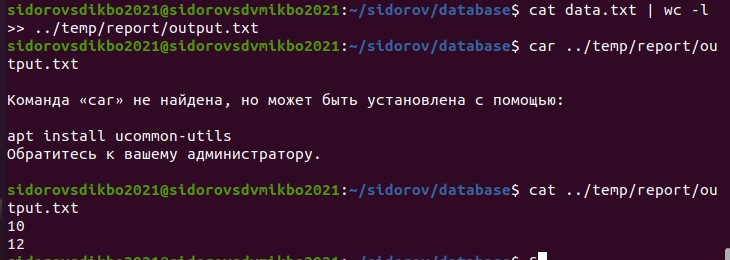


Рисунок 19 - Подсчёт количества строк в обновлённом файле

Фильтрация содержимого файла по жанру с помощью команды grep изображена на рисунке 20.

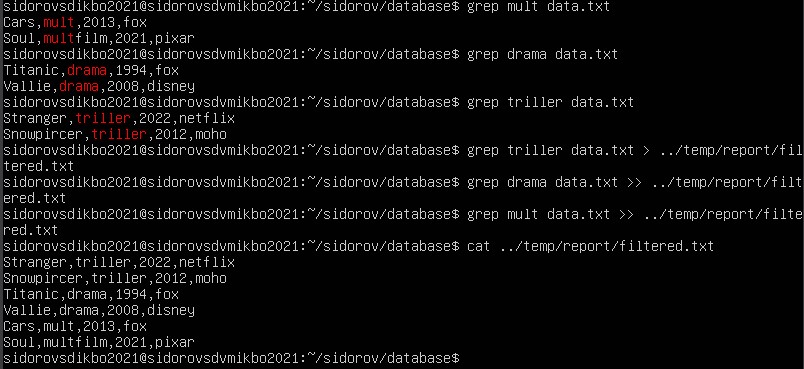


Рисунок 20 - Фильтрация содержимого файла

Сортировка содержимого файла по году выпуска инверсивно - столбец номер 3, с помощью команды sort представлена на рисунке 21.



Рисунок 21 - Сортировка содержимого файла инверсивно

Фильтрация содержимого файла по жанру с помощью команды grep и последующая сортировка результат по году выпуска инверсивно изображена на рисунке 22.

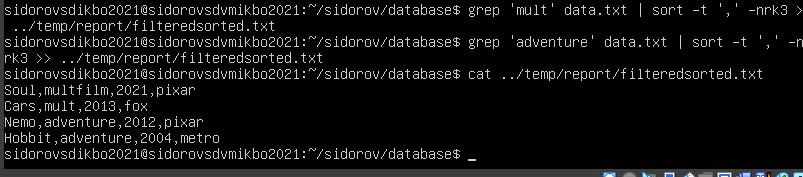


Рисунок 22 - Фильтрация содержимого файла с последующей сортировкой

Исследование команд представлено на рисунке 23.

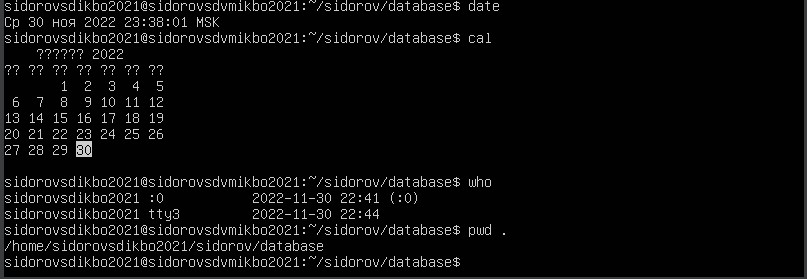


Рисунок 23 - Исследование команд

Запись результата команды cal в файл представлена на рисунке 24.

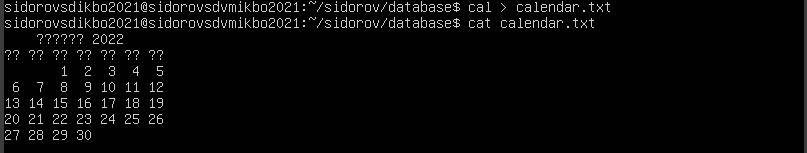


Рисунок 24 - Запись результата вызова cal в файл

## Выводы

В ходе выполнения практической работы были изучены команды операционной системы GNU Linux для работы с элементами файловой системы, а также получены практические навыки по созданию, изменению, манипулированию и удалению файлов и каталогов, а также применению различных алгоритмов к их содержимому.

# Практическая работа 3

**Изучение программируемого фильтра awk**

**Цель работы:**

Целью данной лабораторной работы является изучение возможностей программируемого фильтра AWK при обработке текстовой информации.

**Ход работы:**

**Задание 1:** вывести на экран из файла calendar.txt день недели и текущее число в виде «сегодня вторник … августа»;

Содержимое файла кода для вывода информации представлено на рисунке 24.

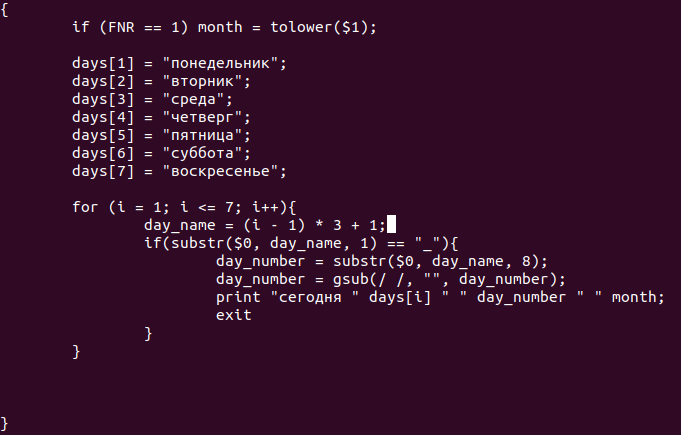


Рисунок 24 - Содержимое файла firstScript

Результат выполнения команды изображён на рисунке 25.



Рисунок 25 - Результат выполнения команды

**Задание 2:**вывести список каталогов, имена которых состоят из русских букв, без дополнительных полей.

Содержимое файла кода для вывода списка каталогов представлено на рисунке 26.

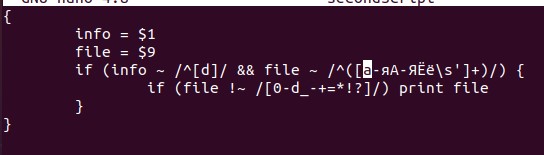
****

Рисунок 26 - Содержимое файла secondScript

Результат выполнения команды изображён на рисунке 27.

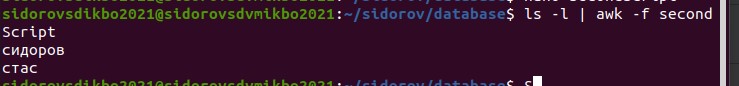


Рисунок 27 - Результат выполнения команды

**Задание 3:** определить количество(сумму) байтов, занятых всеми вашими текстовыми файлами (txt) в каталогах и подкаталогах.

Список текстовых файлов представлен на рисунке 28.

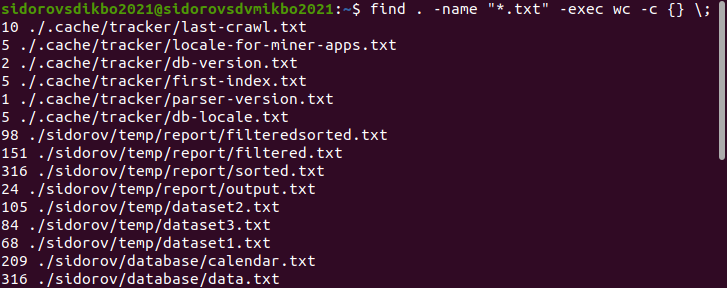


Рисунок 28 - Список текстовых файлов

Результат применения awk изображён на рисунке 29.

task3_res_awk

Рисунок 29 - Результат применения awk

**Задание 4 :** определить количество блоков, содержащих ваш текущий каталог.

Вывод списка файлов текущего каталога представлен на рисунке 30.

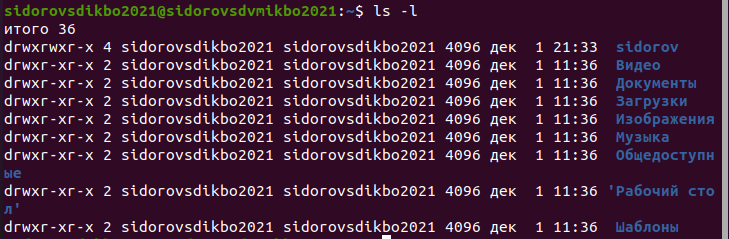


Рисунок 30 - Список файлов текущего каталога

Применение awk для подсчёта количества блоков содержащих текущий каталог изображено на рисунке 31.

count_blocks

Рисунок 31 - Результат подсчёта количества блоков

**Задание 5:** изменить права доступа для некоторых файлов текущего каталога и провести сортировку списка по возможностям доступа.

Вывод отсортированного по правам доступа списка файлов представлен на рисунке 32.

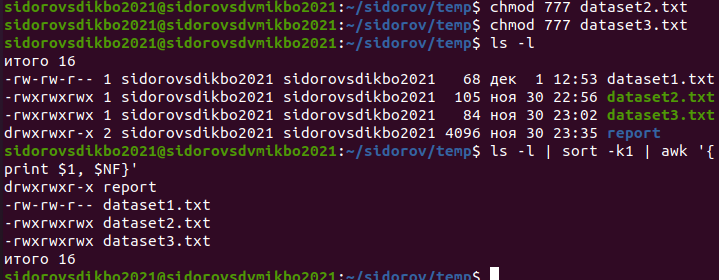


Рисунок 32 - Вывод отсортированного по правам доступа списка файлов

**Задание 6 :** напечатать список каталогов, в которых обнаружены файлы с именами data\*.txt.

Содержимое файла с исполняемым кодом изображено на рисунке 33.

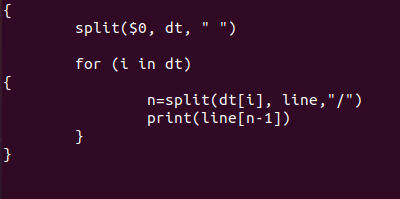


Рисунок 33 - Содержимое файла thirdScript

Результат применения awk представлен на рисунке 34.

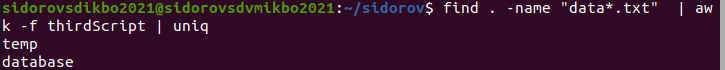


Рисунок 34 - Результат применения awk

**Задание 7 :** подсчитать, сколько раз пользователь входил в систему.

Содержимое исполняемого файла представлено на рисунке 35.

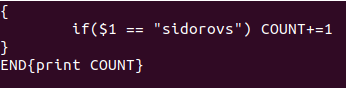


Рисунок 35 - Содержимое файла fourthScript

Результат применения awk изображен на рисунке 36.

task7_res

Рисунок 36 - Количество входов в систему.

**Задание 8 :** напечатать список пользователей, отсортированный по времени.

Содержимое исполняемого файла представлено на рисунке 37.

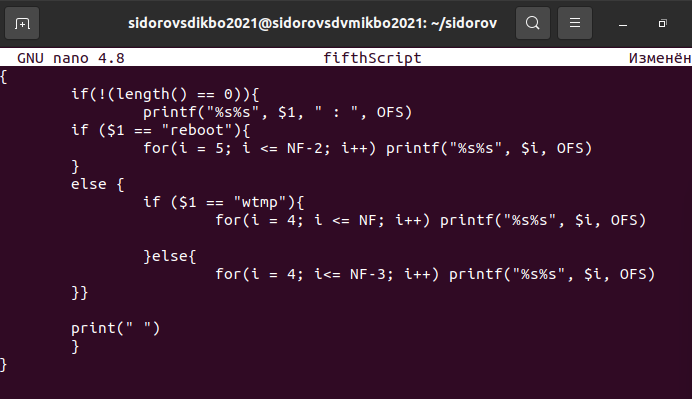


Рисунок 37 - Содержимое файла fifthScript

Результат применения awk изображен на рисунке 38.

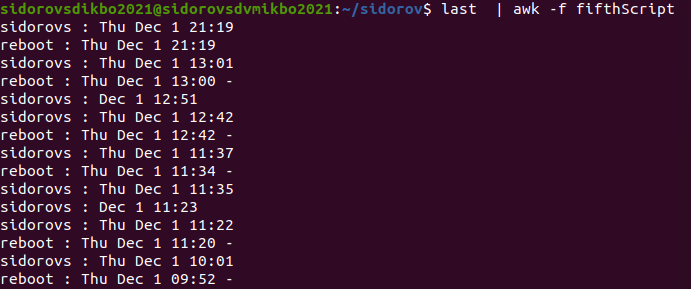


Рисунок 38 - Отсортированный список пользователей.

## Выводы

В ходе выполнения данной практической работы были получены навыки использования программируемого фильтра awk. Были изучены его базовые возможности, а также на практике пройдены возможные сценарии использования.