|  |
| --- |
|  |
| 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

«Aрхитектура вычислительных машин и систем»

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

Москва 2022 г.

**Практическая работа №5**

**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.

**Ход работы:**

1. Скачать дистрибутив Oracle VirtualBox.

Для выполнения шага необходим компьютер, имеющий доступ в интернет. Запускаем браузер. В адресной строке браузера вводим https://www.virtualbox.org/wiki/Downloads На открывшейся странице переходим по ссылке Windows hosts. Сохраняем установочный файл на жесткий диск (рисунок 1.1).

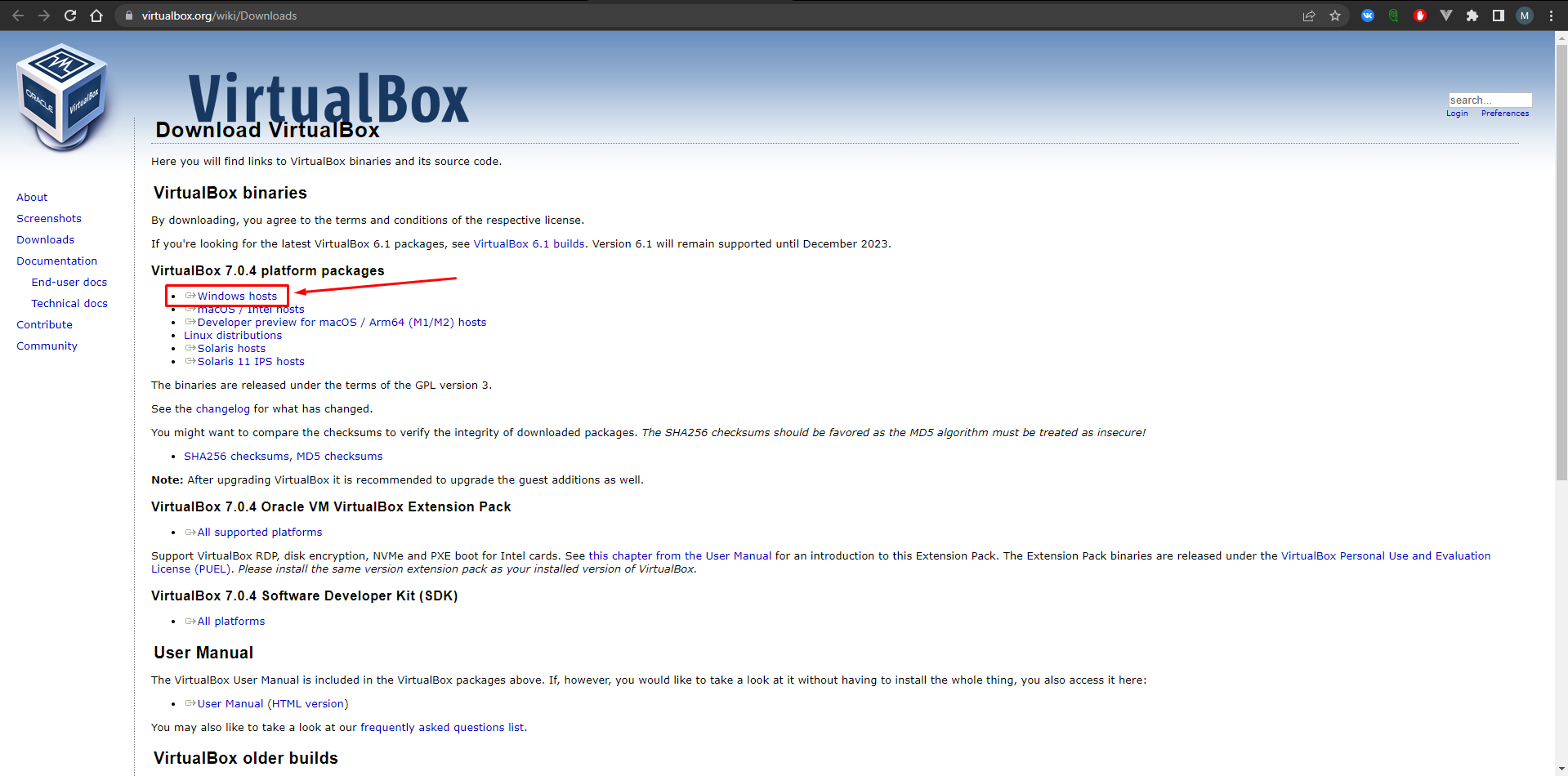


Рисунок 1.1 – установка дистрибутива Oracle VirtualBox

1. Установить Oracle VirtualBox на компьютер.

Для установки OracleVirtualBox на компьютере необходимо запустить скачанный на первом шаге установочный файл дистрибутива (рисунок 1.2).

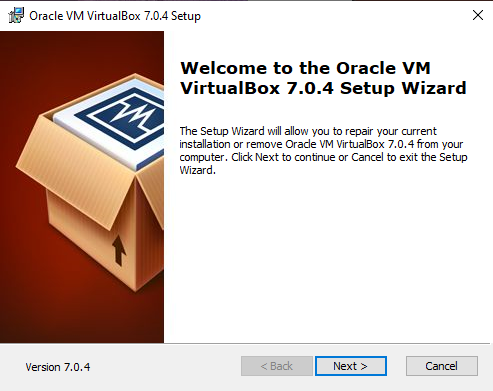


Рисунок 1.2 – запуск скачанного файла

В появившемся диалоге необходимо выбрать компоненты для установки. Для удобства на этапе обучения рекомендуется в диалоге выбрать установку всех предложенных компонентов (рисунок 1.3).

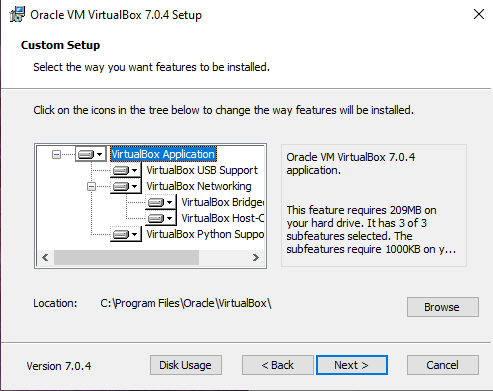


Рисунок 1.3 – выбор компонентов

По завершении установки на рабочем столе компьютера будет создан ярлык для запуска виртуальной машины OracleVirtualBox (рисунок 1.4).

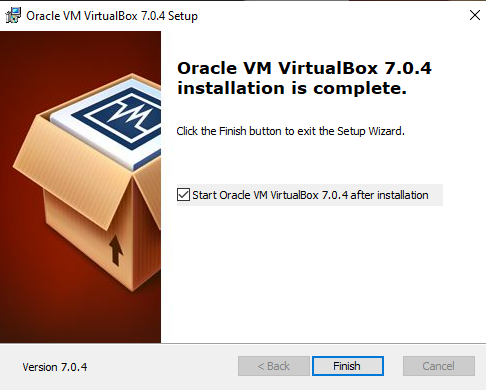


Рисунок 1.4 – окончание установки

1. Получить дистрибутив операционной системы Ubuntu Linux.

Для выполнения шага необходим компьютер, имеющий доступ в интернет. Запускаем браузер. В адресной строке браузера вводим <http://www.ubuntu.com/>. На открывшейся странице переходим по ссылке для скачки. Сохраняем образ установочного диска на жесткий диск.

1. Создать и настроить виртуальную машину.

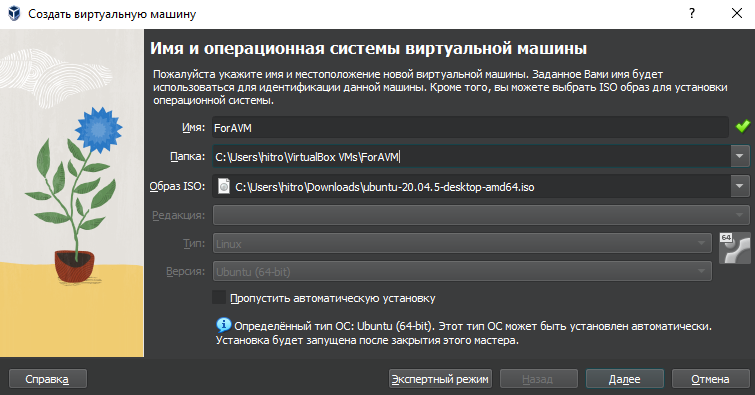


Рисунок 1.5 – Создание виртуальной машины

В зависимости от выбранного типа и версии операционной системы виртуальной машины будут установлены остальные параметры в значения, рекомендованные для функционирования. Все эти значения в процессе настройки можно будет изменить. Далее необходимо задать объем оперативной памяти, выделяемой для функционирования виртуальной машины (рисунок 1.9).

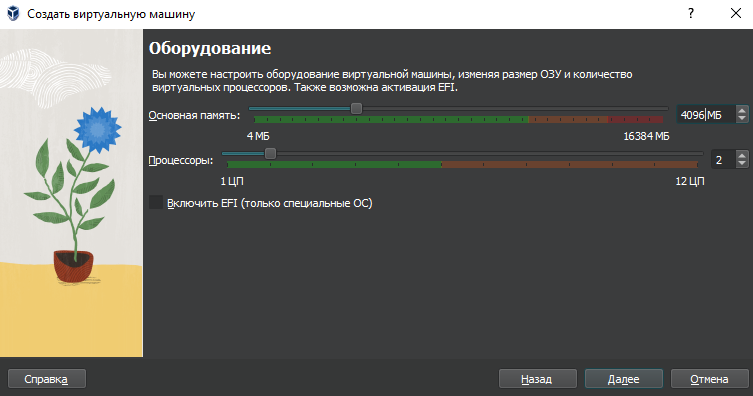


Рисунок 1.6 – Задание оперативной памяти и количества ядер

После выделения оперативной памяти для виртуальной машины необходимо задать размер жесткого диска (рисунок 1.10).

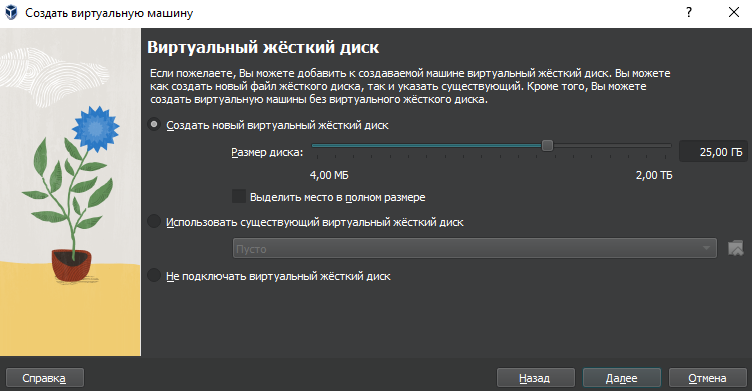


Рисунок 1.7 – Выделение размера жесткого диска

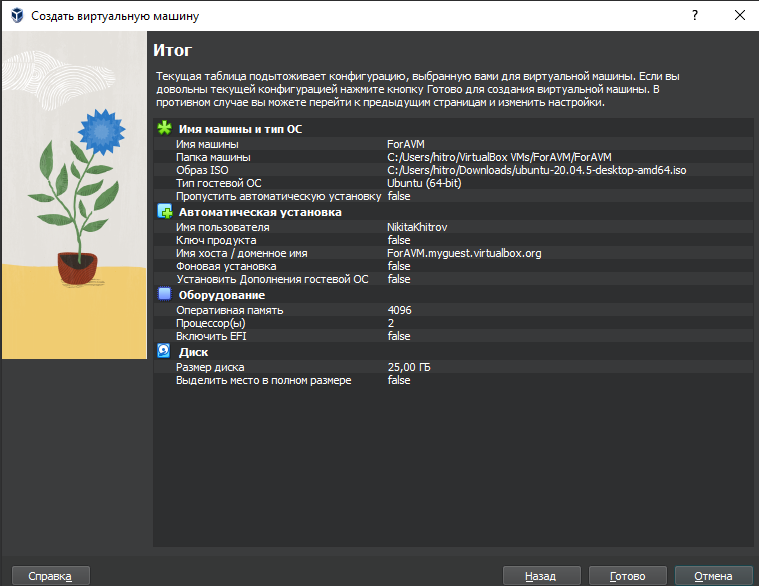


Рисунок 1.8 – Созданная виртуальная машина

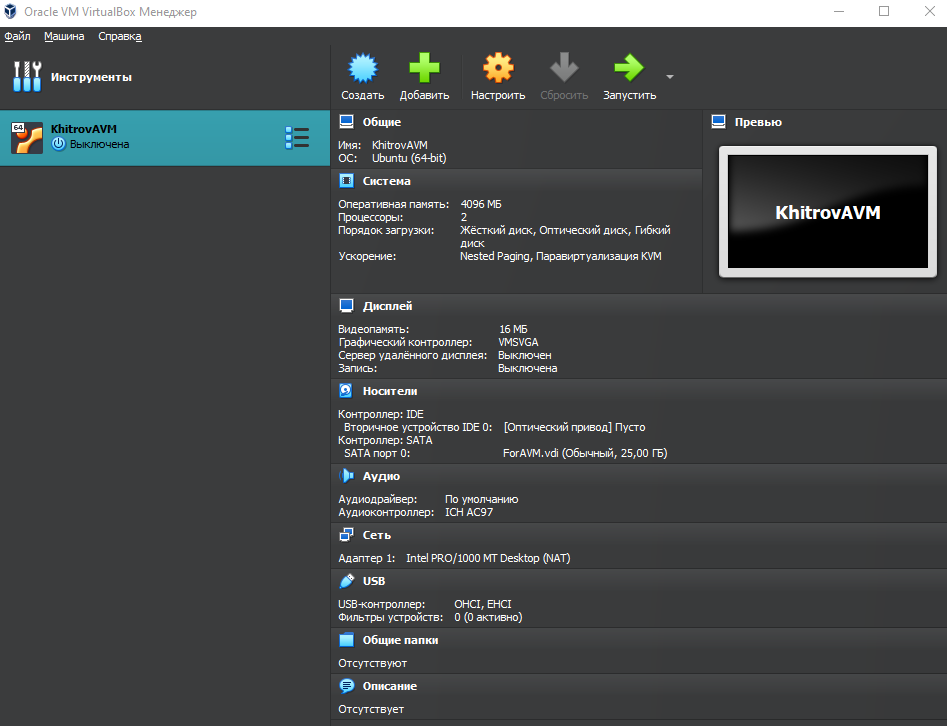


Рисунок 1.9 – Созданная виртуальная в списке виртуальных машин

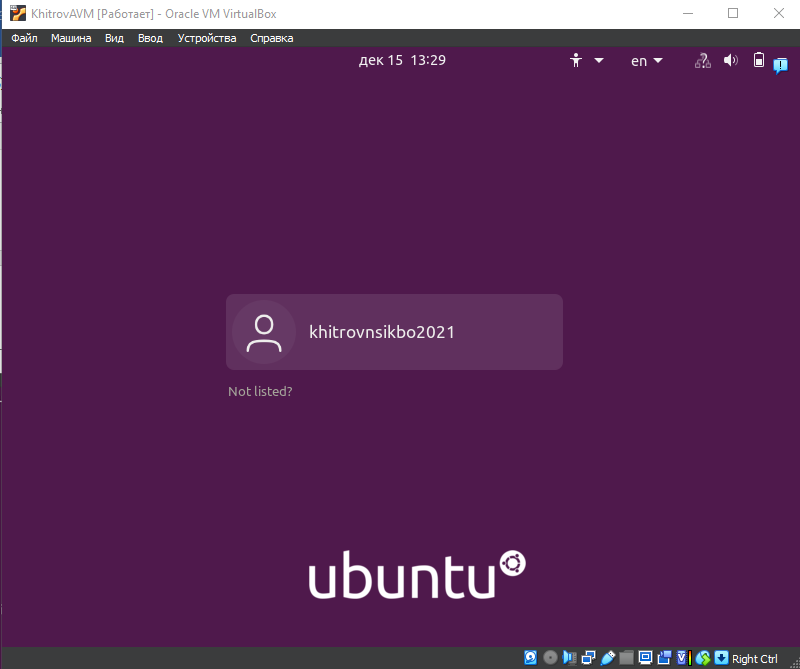


Рисунок 1.10 – Запуск виртуальной машины

**Вывод**

В результате выполнения практической работы был изучен процесс установки на персональный компьютер виртуальной машины OracleVirtualBox, получено представление о процессе создания и настройки виртуального окружения. На примере операционной системы UbuntuLinux был выполнен процесс установки и базовой настройки операционной системы.

**Практическая работа №6**

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

**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.

**Ход работы:**

Персональный вариант – вариант №4: Бухгалтерия (ФИО сотрудников, год поступления на работу, зарплата, номер отдела). Поиск по зарплате. Сортировка по отделам.

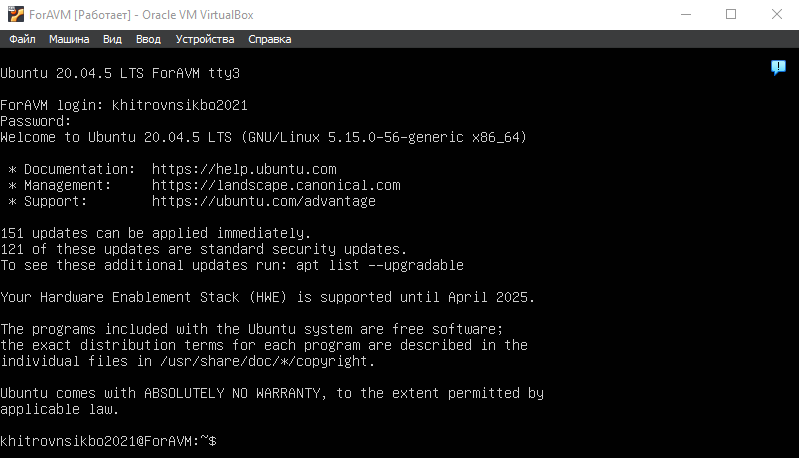
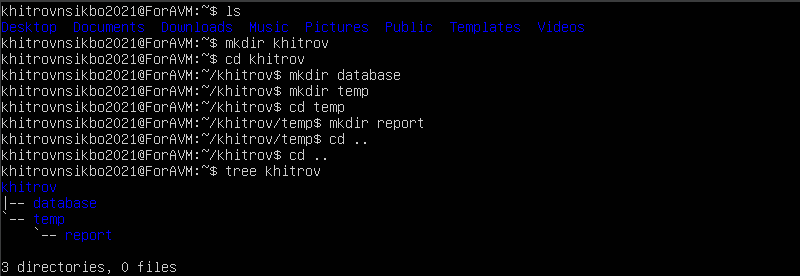
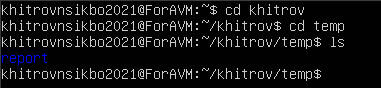


Рисунок 2.1 – вход в систему

Создадим необходимую структуру каталогов и выведем ее в виде дерева (рисунок 2.2).

  
Рисунок 2.2 – создание и вывод структуры созданного каталога

Перейдем в каталог temp. Выведем на экран его содержимое (рисунок 2.3).

  
Рисунок 2.3 – Содержимое каталога temp

Создадим в каталоге temp файл базы данных dataset1.txt и заполним тремя строчками данных, определенных персональным вариантом, с помощью команды встроенного текстового редактора (рисунок 2.4.1 – 2.4.2).

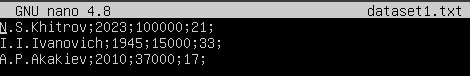
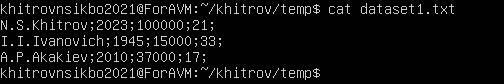


Рисунок 2.4.1 – создание файла dataset1.txt

  
Рисунок 2.4.2 – вывод файла dataset1.txt

Заполним файлы dataset2.txt и dataset3.txt, находящиеся в каталоге temp, с помощью методов, указанных в задании (рисунок 2.5.1 – 2.5.2).

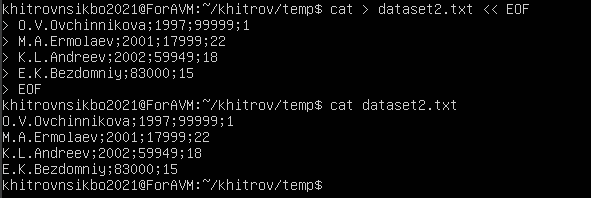
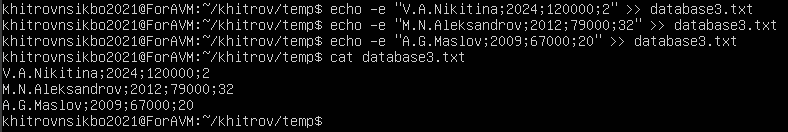


Рисунок 2.5.2 – создание при помощи конвейера и вывод файла dataset2.txt

  
Рисунок 2.5.2 – создание и вывод файла dataset3.txt

Объединим файлы dataset1.txt, dataset2.txt, dataset3.txt в data.txt и переместим полученный файл в нужный каталог. Объединение осуществляется с помощью команды cat.

  
Рисунок 2.6 – Создание файла data.txt

Найдем количество строк в файле data.txt и запишем полученное значение в файл output, находящийся на пути – khitrov/temp/report/ (рисунок 2.7).

  
Рисунок 2.7 – Вывод количества строк в файле data.txt и запись полученного значения в файл output.txt

Дополним файл data.txt двумя строчками и выведем содержимое в терминал (рисунок 2.8.1 – 2.8.2).

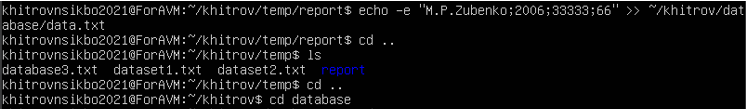
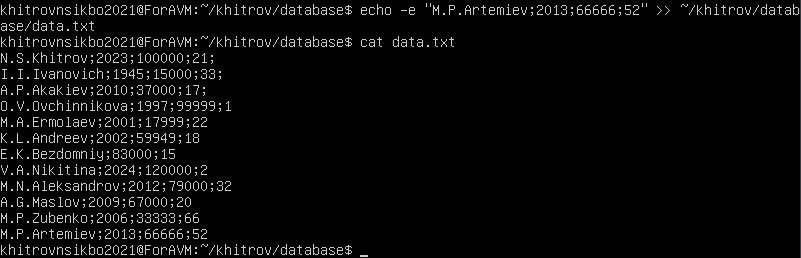
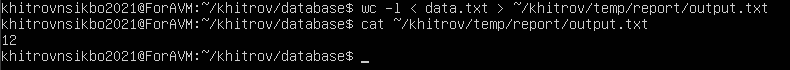


Рисунок 2.8.1 – Дополнение файла data.txt (1)  
Рисунок 2.8.2 – Дополнение файла data.txt (2)

Снова получим значение – количество строк в файле data.txt и запишем его в конец файла output.txt (рисунок 2.9).

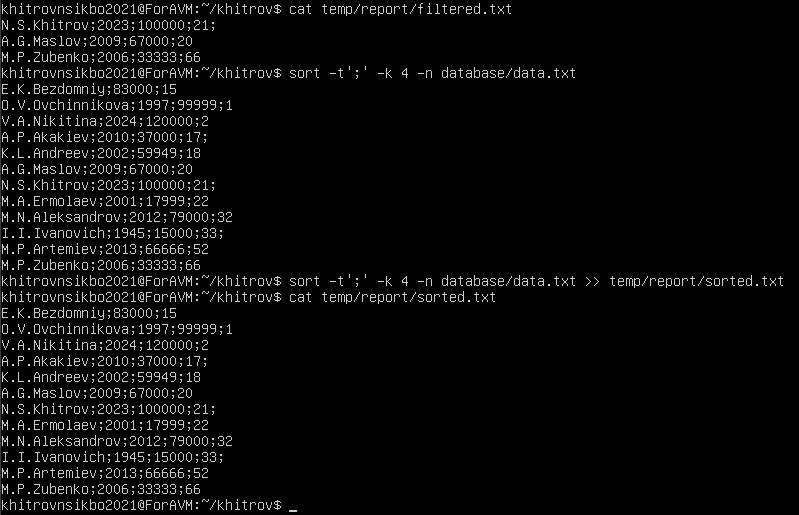
  
Рисунок 2.9 – дополнение содержимого файла output.txt снова подсчитанным кол-вом строк

Осуществим фильтрацию с разными значениями по категории, обусловленной персональным вариантом, и запишем результаты в файл filtered.txt, находящийся в каталоге report (рисунок 2.10).

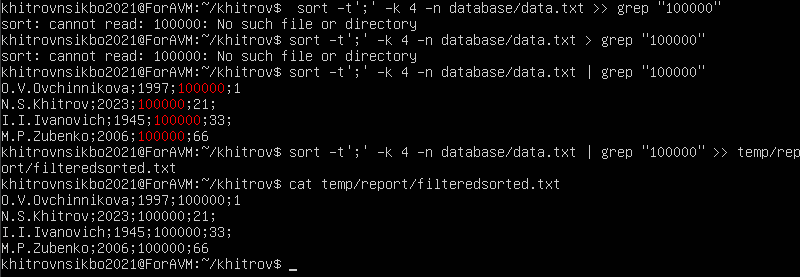


Рисунок 2.10 – Фильтрация данных и запись их в новый файл

Произведем сортировку содержимого файла data.txt. Выведем результат в терминал и запишем в файл sorted.txt (рисунок 2.11).

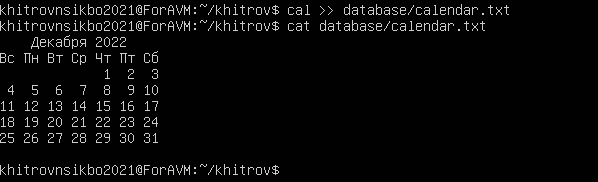
  
Рисунок 2.11 – Сортировка, вывод и запись в новый файл данных из data.txt

При помощи конвейера команд отсортируем файл по 4 столбцу применим команду фильтрации и добавим результат в filteredsorted.txt. Выведем содержимое этих файлов, покажем правильность выполненной работы (рисунок 2.12).

  
Рисунок 2.12 – Фильтрация и сортировка данных, запись в новый файл

Исследуем команды date, cal, pwd, who, clear, exit:

* Команда date – выводит текущую дату и время;
* Команда cal – выводит текстовый формат календаря с отметкой текущего дня, месяца и года;
* Команда pwd – выводит полный путь от корневого каталога к текущему рабочему каталогу;
* Команда who – выводит список текущих пользователей в компьютерной системе;
* Команда clear – очищает историю ввода в терминале;
* Команда exit – завершает процесс командной оболочки с кодом.

  
Рисунок 2.13 – Вывод календаря и запись его в файл calendar.txt

**Вывод**

Было проведено ознакомление с процессом создания структуры каталогов, изучены различные способы создания и манипулирования данными. На примере созданной в процессе практической работы базы данных на основе текстовых файлов были рассмотрены вопросы сортировки и фильтрации информации, вывод требуемых данных на экран и в файл.

# Практическая работа №3

**Использование программируемого фильтра awk**

**Цель работы:**

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

**Ход работы:**

Необходимо вывести на экран из файла calendar.txt день недели и текущее число в виде «сегодня вторник… августа»

На рисунке 3.1 представлен результат работы скрипта.

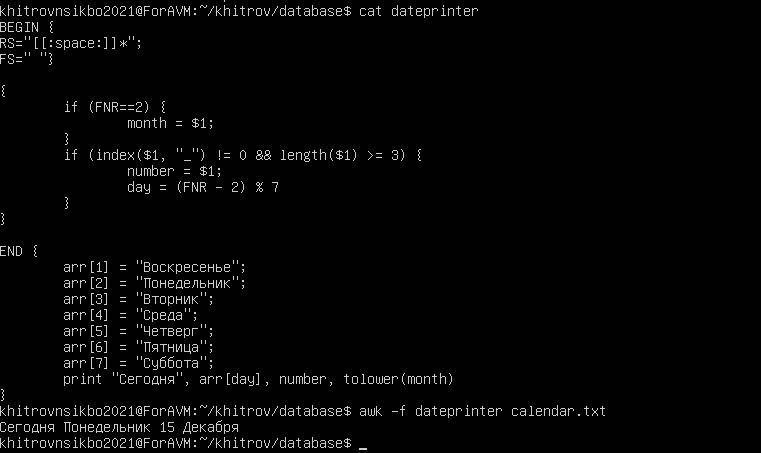


Рисунок 3.1– Вывод дня недели и числа

Выведем список каталогов, имена которых состоят из русских букв (рисунок 3.2)

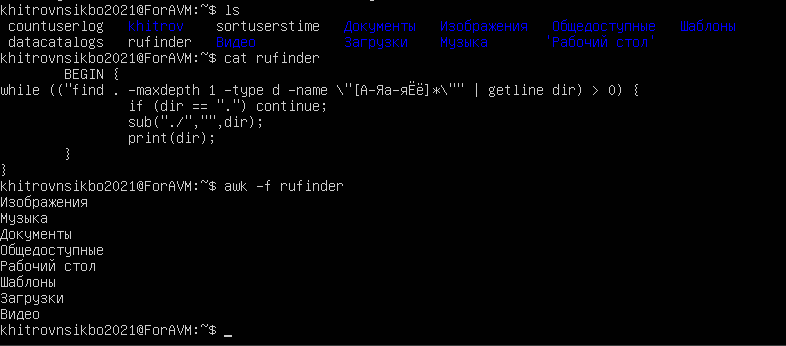


Рисунок 3.2 – Список каталогов из русских букв

Определим количество байт, занятых текстовыми файлами (рисунок 3.3).

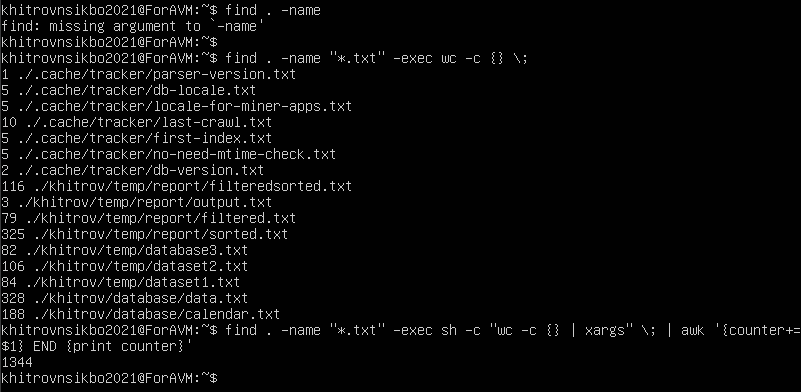


Рисунок 3.3 – Сумма байтов, занятая текстовыми файлами

Посчитаем количество блоков, занятых текущим каталогом (рисунок 3.4)



Рисунок 3.4 – Количество блоков, содержащих текущий каталог

Отсортируем файлы текущего каталога по правам доступа (рисунок 3.5).

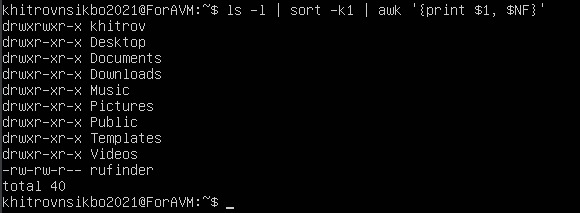


Рисунок 3.5 – Сортировка файлов по правам доступа

Напечатаем список каталогов, в которых обнаружены файлы с именами data\*.txt (рисунок 3.6).

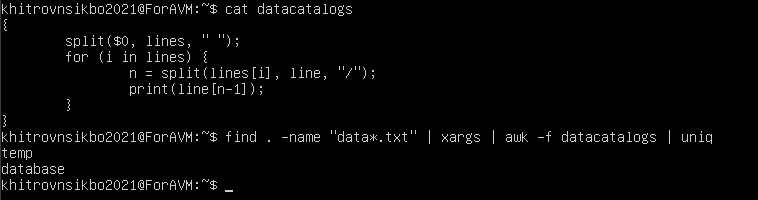


Рисунок 3.6 – Список каталогов, содержащих текстовый файлы

Посчитаем, сколько раз пользователь входил в систему (рисунок 3.7).

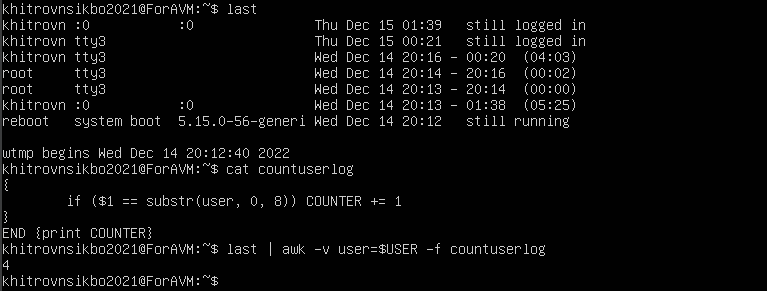


Рисунок 3.7 – Количество входов в систему

Напечатаем список пользователей (рисунки 3.8). Он отсортирован по времени.

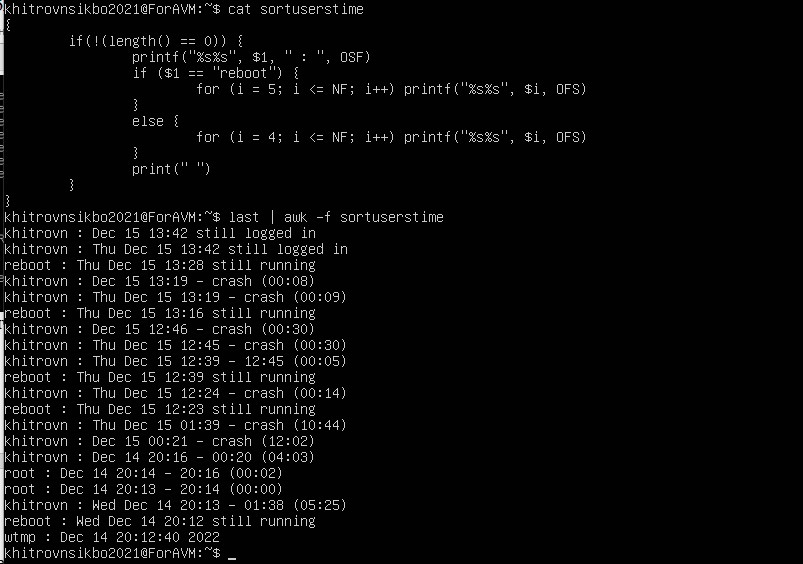


Рисунок 3.8 – Список пользователей, отсортированный по времени

**Вывод**

В данной практической работе мы познакомились с возможностями программируемого фильтра awk. Фильтр широко применяется для обработки данных и формирования различного вида отчетов.