



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА - Российский технологический университет»

РТУ МИРЭА

Институт Информационных Технологий
Кафедра Вычислительной Техники (ВТ)

ОТЧЁТ ПО ПРАКТИЧЕСКИМ РАБОТАМ

по дисциплине

«Архитектура вычислительных машин и систем»

Выполнил студент группы
ИКБО-13-22

Лещенко Вячеслав Романович

Принял преподаватель кафедры ВТ

Рыжова Анастасия Андреевна

Практические работы выполнены

«__»_____2023 г.

«Зачтено»

«__»_____2023 г.

Москва 2023 г.

СОДЕРЖАНИЕ

Практическая работа №1	3
Практическая работа №2	6
Практическая работа №3	9
Практическая работа №4	12
Практическая работа №5	15
Практическая работа №6	18
Практическая работа №7	23
Практическая работа №8	26
Практическая работа №9	37
Практическая работа №10	52
Список информационных источников	59

Практическая работа №1

Тема: Графический ввод схемы и симуляция в САПР QUARTUS II

Цель работы: Спроектировать логическую схему при помощи графического редактора САПР QUARTUS II. Исследовать работу схемы с использованием сигнального редактора САПР QUARTUS II.

Постановка задачи

1. Изучить правила построения, принцип работы логических схем.
2. Синтезировать электрическую принципиальную схему логического устройства, описанного заданным преподавателем уравнением в алгебраической форме.
3. Нарисовать синтезированную схему в графическом редакторе САПР QUARTUS II.
4. Произвести симуляцию работы схемы. Зарисовать диаграммы работы и по ее результатам заполнить таблицу истинности смоделированной схемы.

Персональный вариант: 10) $Y = A\bar{C} \vee A\bar{B} \vee AC\bar{D} \vee \bar{B}\bar{C}$

Ход работы

На первом этапе практической работы были изучены основы построения и функционирования логических схем. Это включало в себя понимание базовых операций И, ИЛИ, НЕ, а также понимание булевой алгебры и законов Де Моргана.

Далее, была проведена работа по синтезу электрической принципиальной схемы для логического устройства, описанного уравнением в алгебраической форме. Для этого были применены знания, полученные на первом этапе. Схема была разработана с использованием логических элементов, таких как И, ИЛИ, НЕ.

Созданная электрическая схема была воплощена в графическом редакторе САПР QUARTUS II. В этой программе были использованы доступные инструменты для создания и визуализации логических схем (рис. 1).

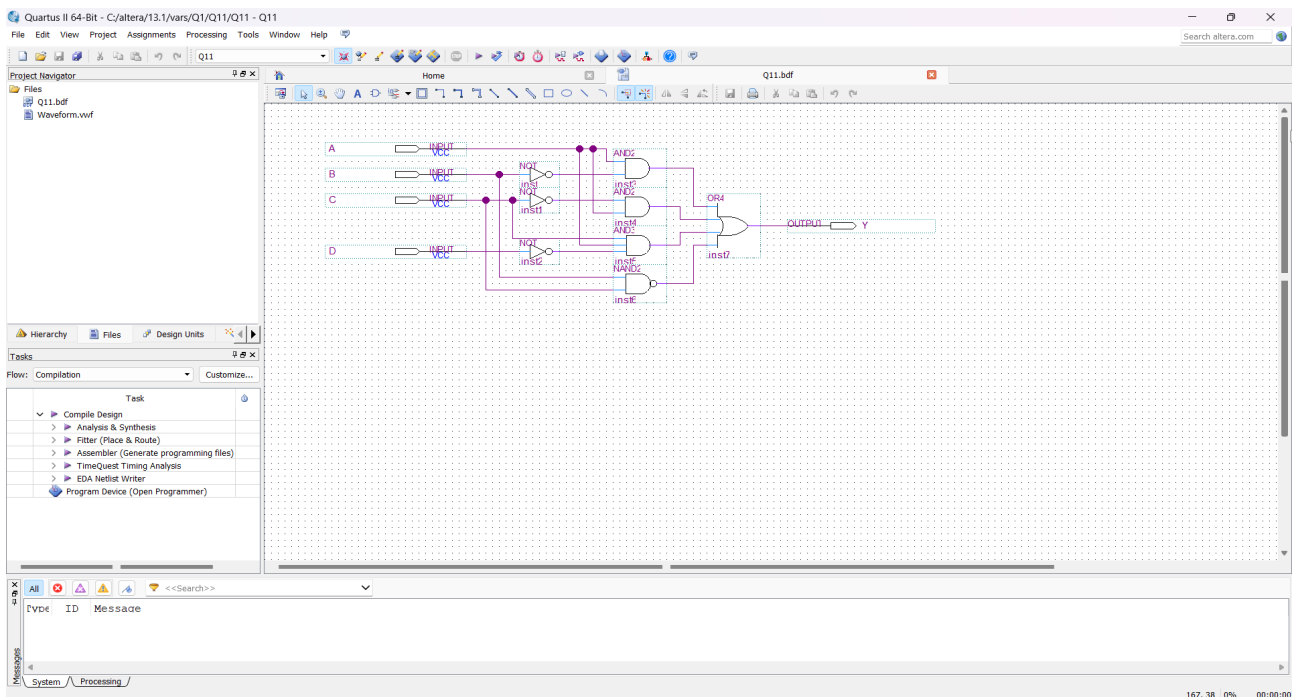


Рисунок 1 – Цифровая схема

Для проверки работоспособности синтезированной схемы была проведена симуляция работы. Диаграммы работы схемы были созданы с использованием сигнального редактора внутри QUARTUS II (рис. 2).

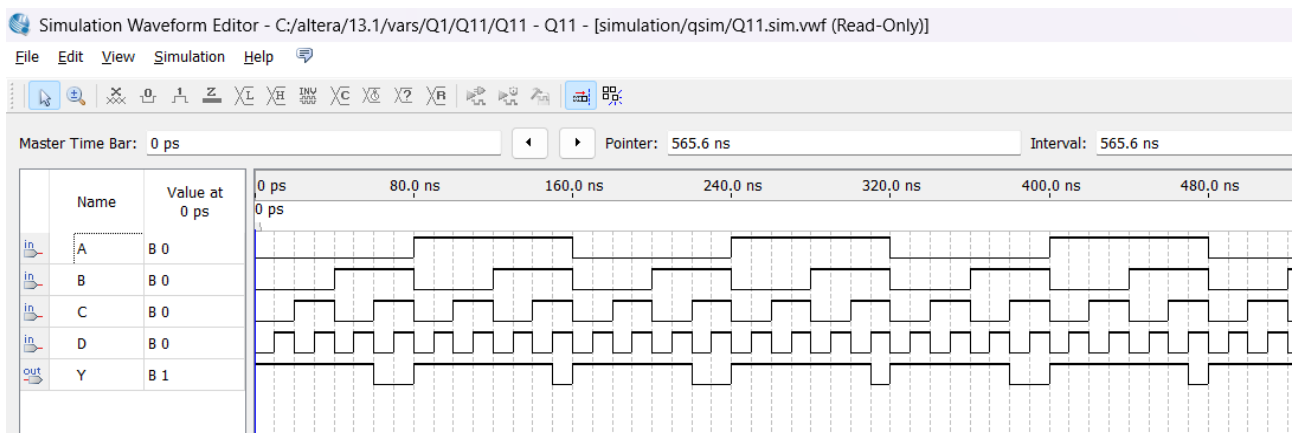


Рисунок 2 – Результаты моделирования работы схемы в сигнальном редакторе

ВЫВОД

Практическая работа дала ценные навыки проектирования и анализа логических схем с использованием САПР QUARTUS II. Были получены навыки разработки электрических принципиальных схем, моделирования и анализа их работы. Эти навыки могут быть полезными в будущей профессиональной деятельности в области цифровой электроники и схемотехники.

Практическая работа №2

Тема: Описание логических схем при помощи языка AHDL

Цель работы: Приобретение основных навыков описания цифровых схем с помощью языка описания аппаратуры AHDL. Смоделировать логическую схему при помощи текстового редактора САПР QUARTUS II.

Постановка задачи

1. Изучить основные элементы языка AHDL и правила описания логических схем.
2. Сделать описание электрической схемы заданной в предыдущей работе при помощи текстового редактора САПР QUARTUS II.
3. Произвести симуляцию работы схемы. Зарисовать диаграммы работы и по ее результатам заполнить таблицу истинности смоделированной схемы.
4. Сравнить результаты, полученные в ходе выполнения практической работы с результатами, полученными в работе №1.

Персональный вариант: 10) $Y = A\bar{C} \vee A\bar{B} \vee AC\bar{D} \vee \bar{B}\bar{C}$

Ход работы

После изучения основных элементов языка было произведено описание логической схемы в текстовом редакторе (рис. 3).

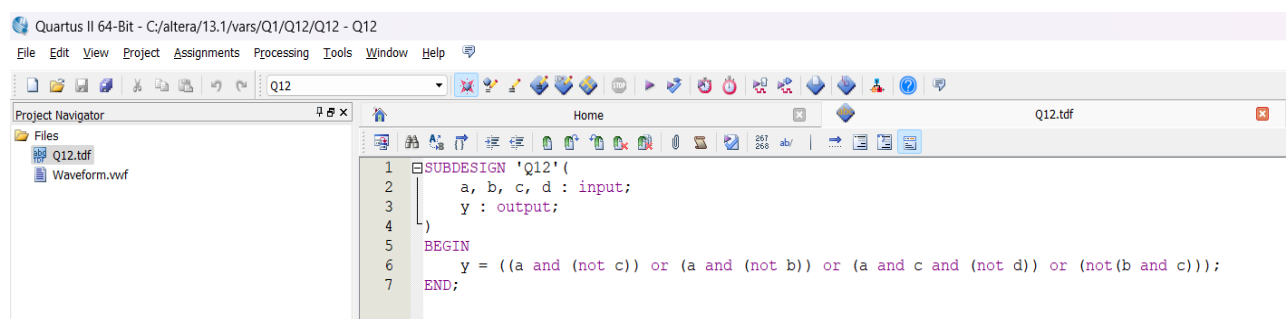


Рисунок 3 – Описание логической схемы на языке AHDL

Далее была получена диаграмма работы схемы. Сравним результат с 1 практической работой (рис. 4-5).

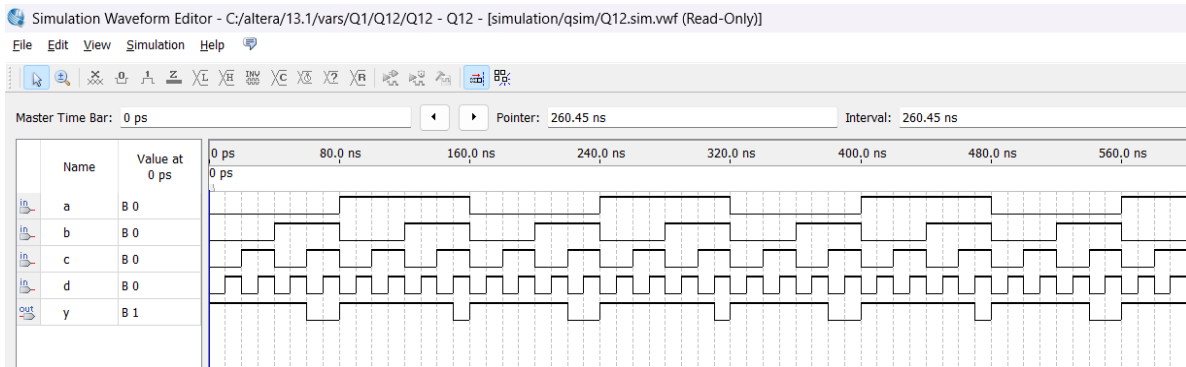


Рисунок 4 - Результаты моделирования работы схемы в сигнальном редакторе 2 практической

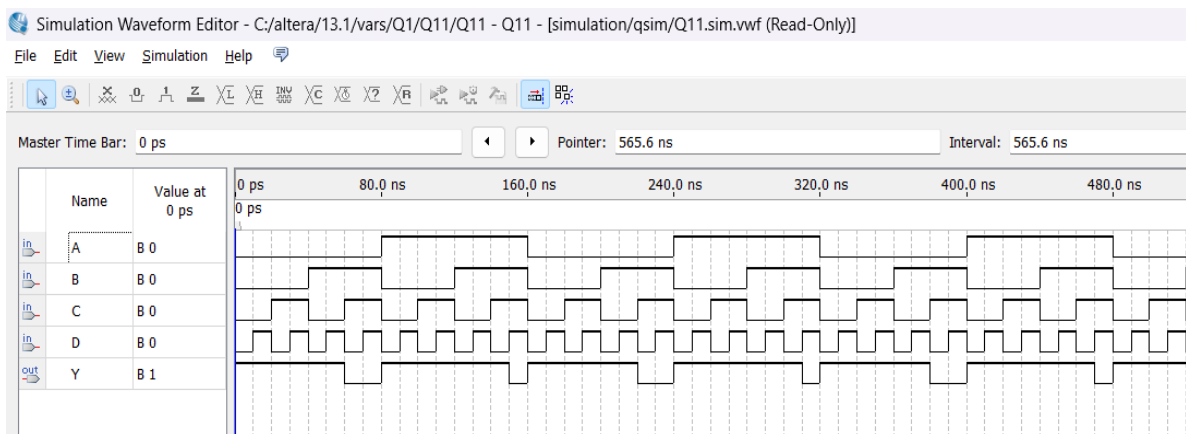


Рисунок 5 - Результаты моделирования работы схемы в сигнальном редакторе 1 практической

ВЫВОД

В ходе практической работы были приобретены основные навыки описания цифровых схем с использованием языка описания аппаратуры AHDL. Была успешно смоделирована логическая схема при помощи текстового редактора САПР QUARTUS II. Это позволило углубить понимание работы цифровых схем и освоить инструменты для их создания и анализа. Результаты работы подтвердили эффективность использования AHDL и САПР QUARTUS II для моделирования и анализа цифровых схем. Это дает основание для дальнейшего изучения и применения этих инструментов в будущих проектах.

Практическая работа №3

Тема: Графический ввод схемы дешифратора и симуляция в САПР QUARTUS II

Цель работы: Спроектировать работу дешифратора при помощи графического редактора САПР QUARTUS II. Исследовать работу схемы с использованием сигнального редактора САПР QUARTUS II.

Постановка задачи

1. Смоделировать схему приоритетного шифратора 4х2.
2. Нарисовать синтезированную схему в графическом редакторе САПР QUARTUS II.
3. Произвести симуляцию работы схемы. Зарисовать диаграммы работы и по ее результатам заполнить таблицу истинности смоделированной схемы.

Ход работы

В результате выполнения практической работы была получена схема и ее временная диаграмма (рис. 6-7).

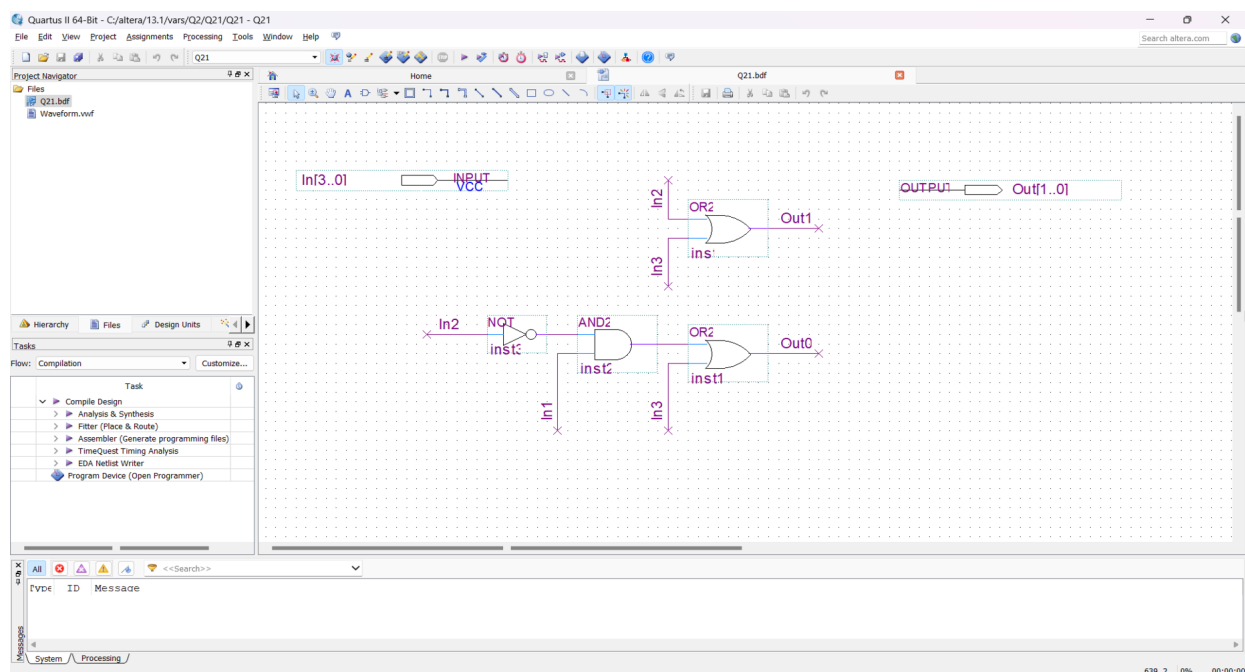


Рисунок 6 – Схема Приоритетного шифратора

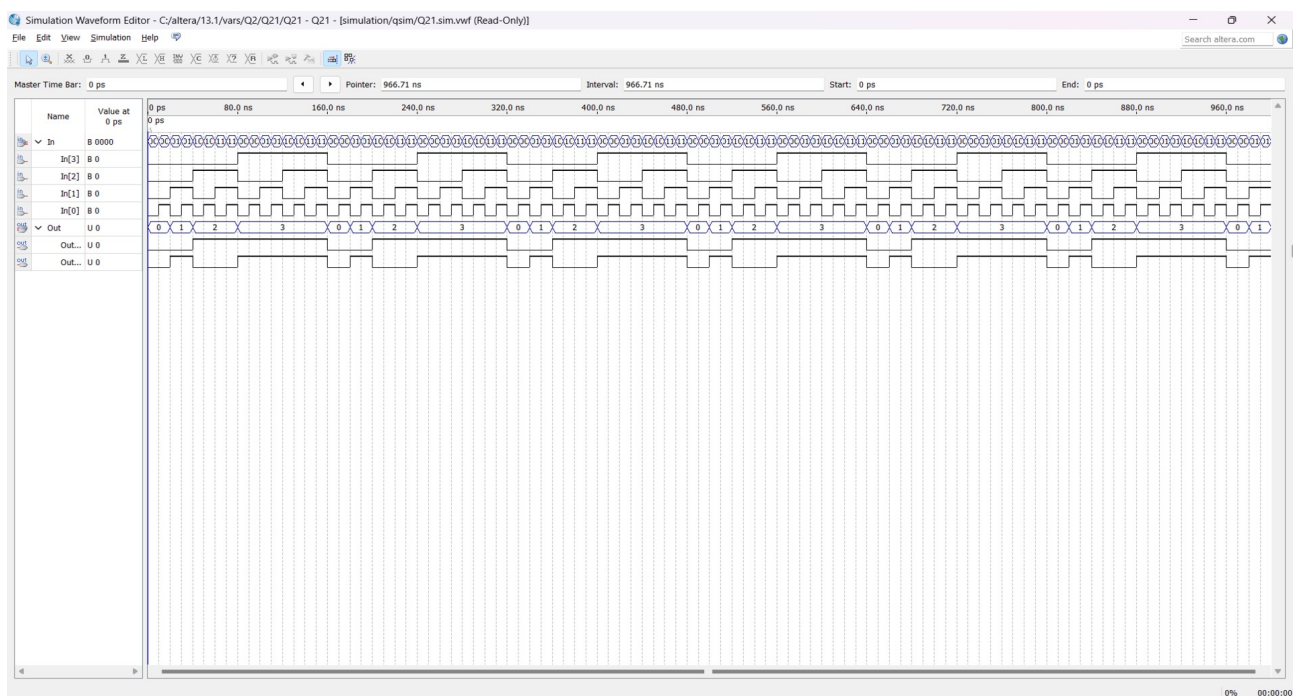


Рисунок 7 - Результаты моделирования работы схемы в сигнальном редакторе

ВЫВОД

В ходе практической работы по моделированию схемы шифратора были получены важные навыки и знания. Было проведено моделирование схемы шифратора, что позволило углубить понимание принципов работы шифраторов и освоить инструменты для их создания и анализа. Результаты работы подтвердили эффективность использования моделирования для анализа работы шифраторов. Это дает основание для дальнейшего изучения и применения этих инструментов в будущих проектах.

Практическая работа №4

Тема: Описание логической схемы дешифратора при помощи языка AHDL

Цель работы: Смоделировать логическую схему дешифратора при помощи текстового редактора САПР QUARTUS II.

Постановка задачи

1. Сделать описание электрической схемы заданной в предыдущей работе при помощи текстового редактора САПР QUARTUS II.
2. Произвести симуляцию работы схемы. Зарисовать диаграммы работы и по ее результатам заполнить таблицу истинности смоделированной схемы.
3. Сравнить результаты, полученные в ходе выполнения практической работы с результатами, полученными в работе №1.

Ход работы

В результате практической работы получен код на языке AHDL (рис. 8). Для сравнения результата работы кода (рис. 9) приведен скриншот работы прошлой программы (рис. 10).

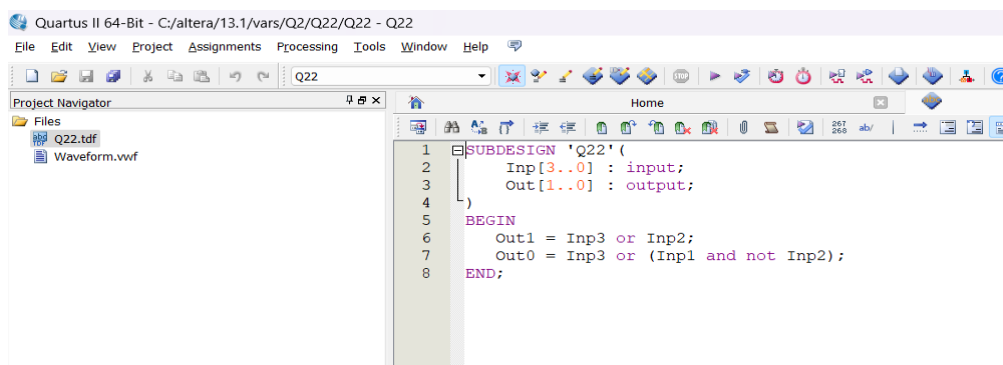
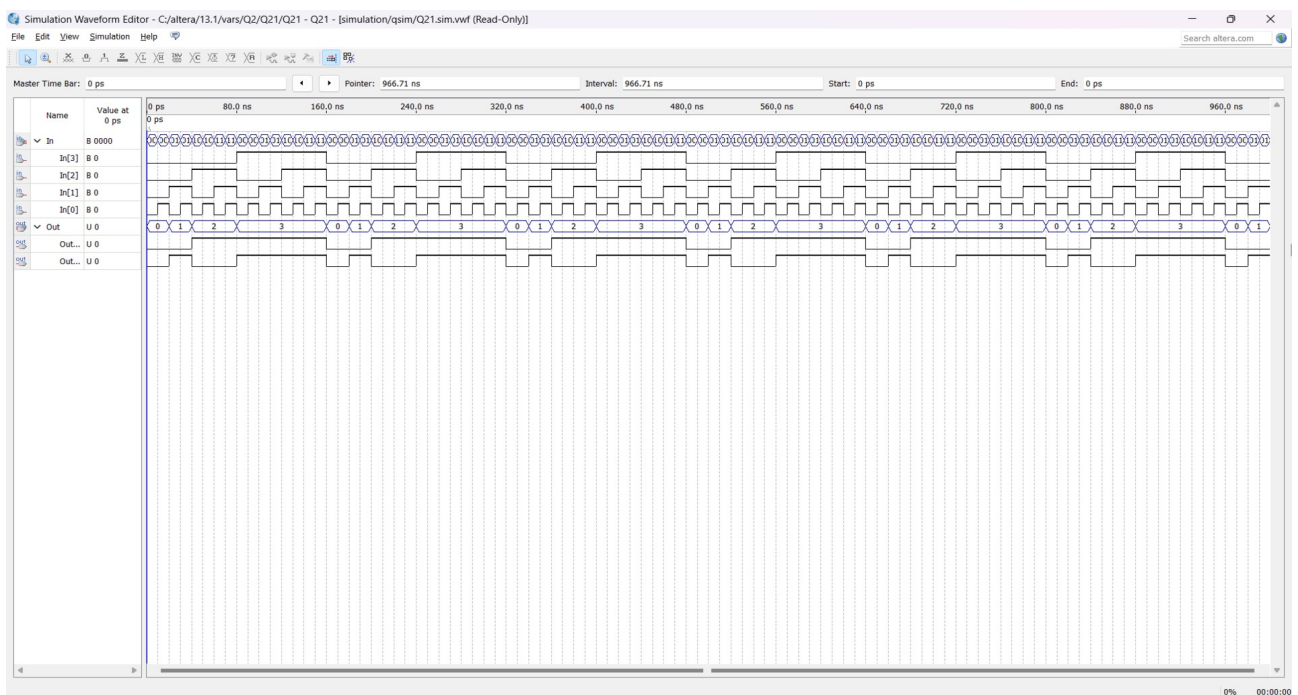
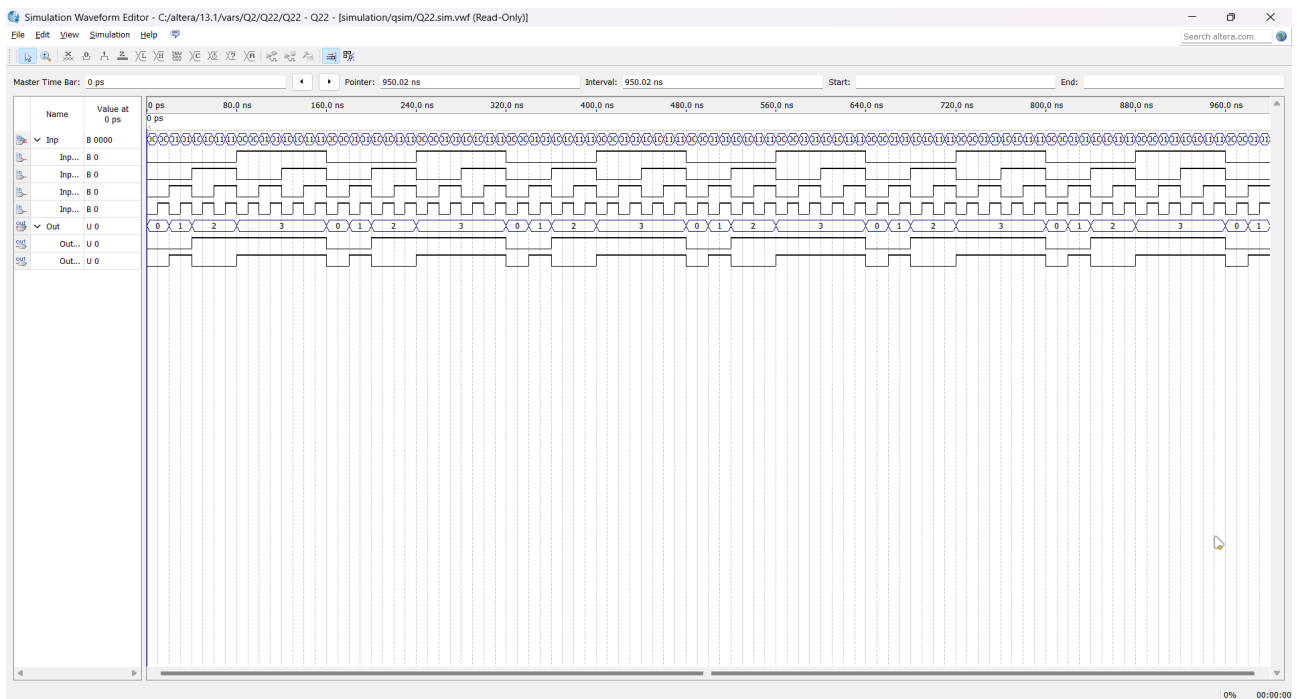


Рисунок 8 - Описание логической схемы на языке AHDL



ВЫВОД

В ходе практической работы по моделированию схемы шифратора на AHDL были получены ценные навыки и знания. Было проведено моделирование схемы шифратора с использованием AHDL, что позволило углубить понимание принципов работы шифраторов и освоить инструменты для их создания и анализа. Результаты работы подтвердили эффективность использования AHDL для моделирования и анализа работы шифраторов. Это дает основание для дальнейшего изучения и применения этих инструментов в будущих проектах.

Практическая работа №5

Тема: Моделирование цифровых схем с использованием параметрических элементов

Цель работы: Приобретение навыков использования параметрических элементов (LPM function) в САПР QUARTUS II, экспериментальное исследование счетчиков и регистров, построенных на их основе.

Постановка задачи

1. Изучить правила построения и принцип работы триггеров и построение на их основе логических схем.
2. Спроектировать электрическую схему обратного счетчика с модулем счета 17 при помощи графического редактора и с использованием параметрических элементов САПР QUARTUS II.
3. Произвести симуляцию работы схемы, зарисовать диаграммы работы схемы.

Ход работы

В результате практической работы была получена параметризованная схема счетчика (рис. 11). Результат работы схемы (рис. 12).

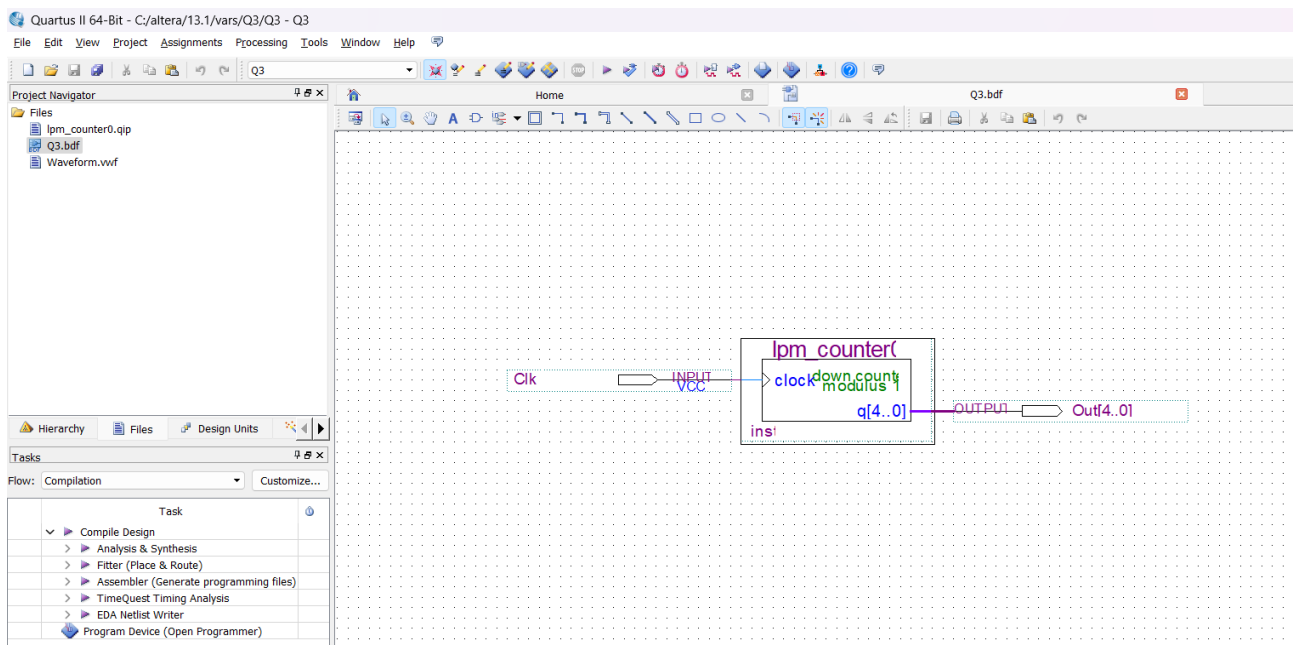


Рисунок 11 – Схема параметризованного дешифратора

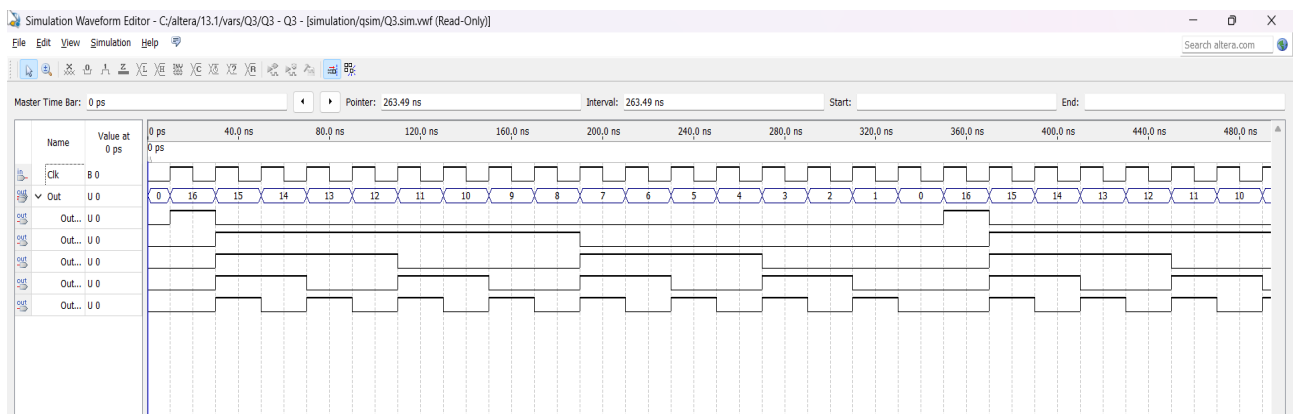


Рисунок 12 - Результаты моделирования работы схемы в сигнальном редакторе

ВЫВОД

В ходе практической работы по моделированию цифровых схем с использованием параметрических элементов были приобретены важные навыки и знания. Было проведено экспериментальное исследование счетчика, построенного на основе параметрических элементов (LPM function) в САПР QUARTUS II. Результаты работы подтвердили эффективность использования параметрических элементов для моделирования и анализа работы цифровых схем. Это дает основание для дальнейшего изучения и применения этих инструментов в будущих проектах.

Практическая работа №6

Тема: Счетчик с произвольным модулем счета

Цель работы: Ознакомиться с САПР QUARTUS II фирмы Altera, получить практические навыки создания проектов по схемотехнике ЭВМ в САПР (ввод схем, компиляция и моделирование).

Постановка задачи

1. Согласно своему варианту графа состояний автомата разработать функциональную электрическую схему цифрового программируемого устройства преобразования кодов.
2. Включить ЭВМ и запустить САПР QUARTUS II.
3. Создать проект, ввести разработанную схему, откомпилировать и отмоделировать её.
4. Проверить полученные результаты, сверив их с таблицей истинности устройства.

Персональный вариант: №20: 12 11 3 0 5 15 10 13 4 6 7 8 14 2 9 1

Ход работы

На первом этапе был графически визуализирован граф переходов автомата, согласно персональному варианту (рис. 13).

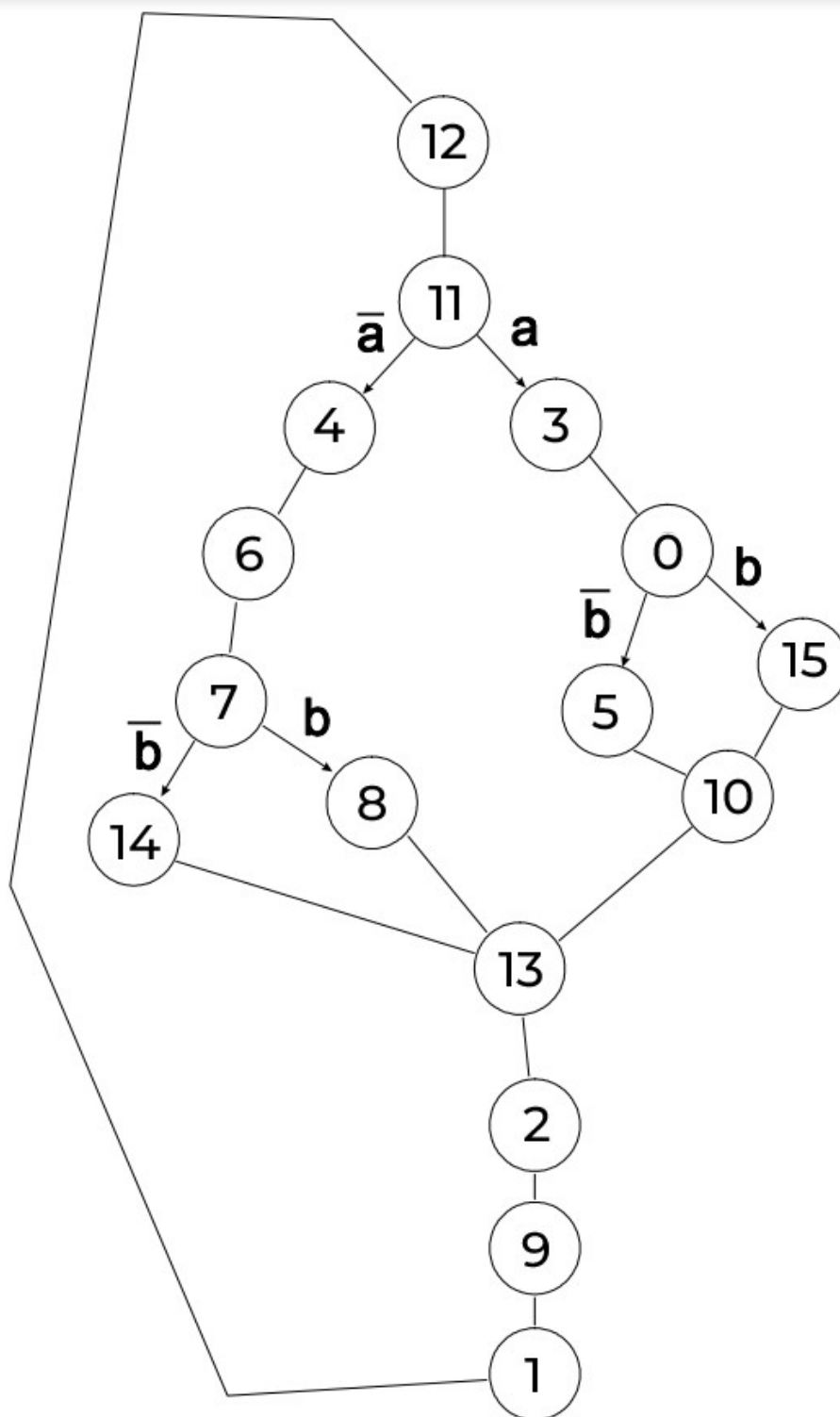


Рисунок 13 – Граф переходов состояний счётчика

Далее необходимо было построить таблицу переходов состояний счётчика (рис. 14-15).

№ состояния	№ состояния из табл.1	Двоичный код q3,q2,q1,q0
0	12	1100
1	11	1011
2	3	0011
3	0	0000
4	5	0101
5	15	1111
6	10	1010
7	13	1101
8	4	0100
9	6	0110
10	7	0111
11	8	1000
12	14	1110
13	2	0010
14	9	1001
15	1	0001

Рисунок 14 – Таблица переходов состояний счётчика

Старое состояние		Условие	Новое состояние	
№	Код		№	Код
12	1100		11	1011
11	1011	A=0	4	0100
11	1011	A=1	3	0011
4	0100		6	0110
6	0110		7	0111
7	0111	B=0	14	1110
7	0111	B=1	8	1000
14	1110		13	1101
13	1101		2	0010
2	0010		9	1001
9	1001		1	0001
1	0001		12	1100
6	0110		4	0100
0	0000	B=0	5	0101
0	0000	B=1	15	1111
5	0101		10	1010
10	1010		13	1101
8	1000		13	1101
15	1111		10	1010

Рисунок 15 – Таблица переходов состояний счётчика

Реализуем счётчик в САПР QUARTUS II (рис. 15).

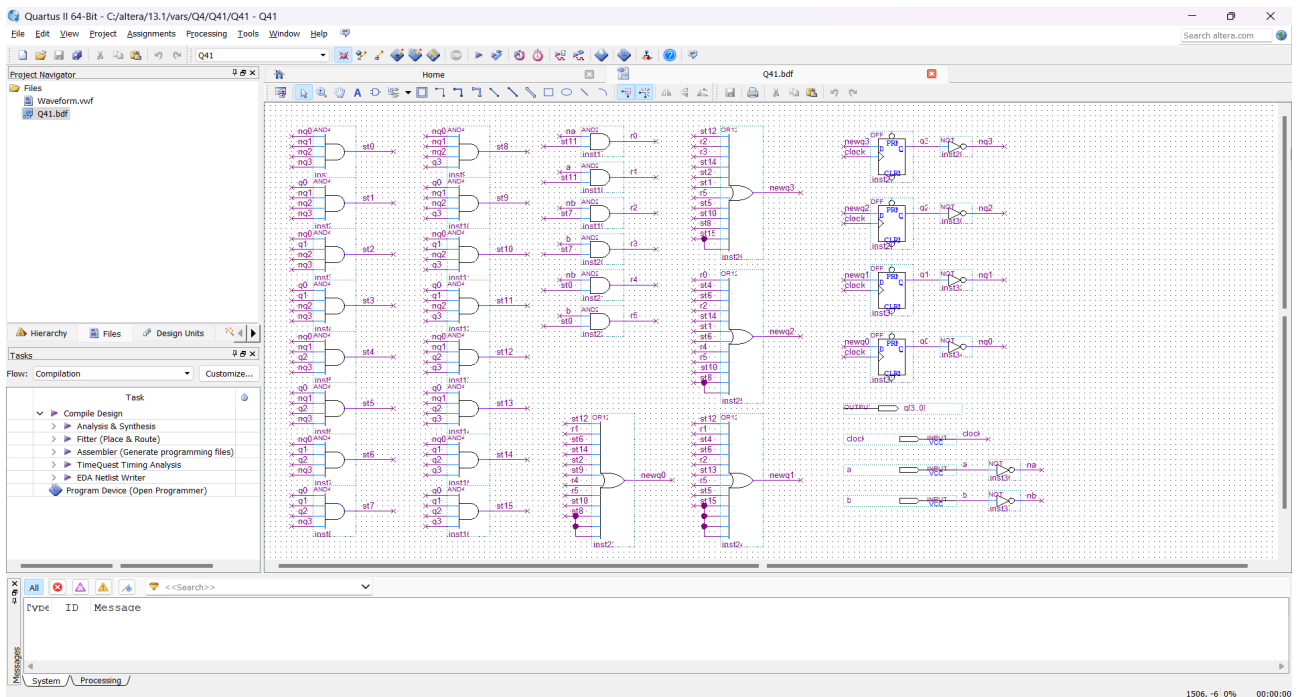


Рисунок 15 – Функциональная схема работы счётчика

Сверим результат работы счётчика с графом, как видно переходы совпадают, следовательно счётчик построен корректно (рис. 16).

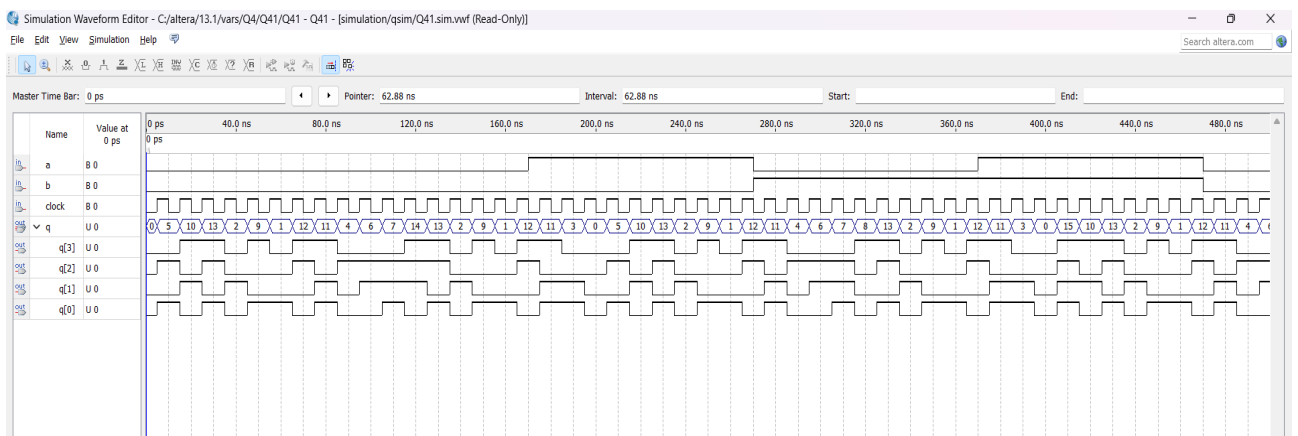


Рисунок 16 - Результаты моделирования работы схемы в сигнальном редакторе

ВЫВОД

В ходе практической работы были получены важные навыки моделирования счетчиков. Это позволило углубить понимание принципов работы счетчиков и освоить инструменты для их создания и анализа. Результаты работы подтвердили эффективность использования САПР QUARTUS II для моделирования счетчиков с произвольным модулем счета. Это дает основание для дальнейшего изучения и применения этих инструментов в будущих проектах.

Практическая работа №7

Тема: Счетчик с произвольным модулем счета на языке AHDL

Цель работы: Приобрести навыки описания счётчиков в Quartus II на языке AHDL.

Постановка задачи

1. Сделать описание электрической схемы заданной в предыдущей работе при помощи текстового редактора САПР QUARTUS II.
2. Произвести симуляцию работы схемы. Зарисовать диаграммы работы и по ее результатам заполнить таблицу истинности смоделированной схемы.
3. Сравнить результаты, полученные в ходе выполнения практической работы с результатами, полученными в работе №6.

Персональный вариант: №20: 12 11 3 0 5 15 10 13 4 6 7 8 14 2 9 1

Ход работы

В результате работы получаем описание электрической схемы на языке AHDL (рис. 17).

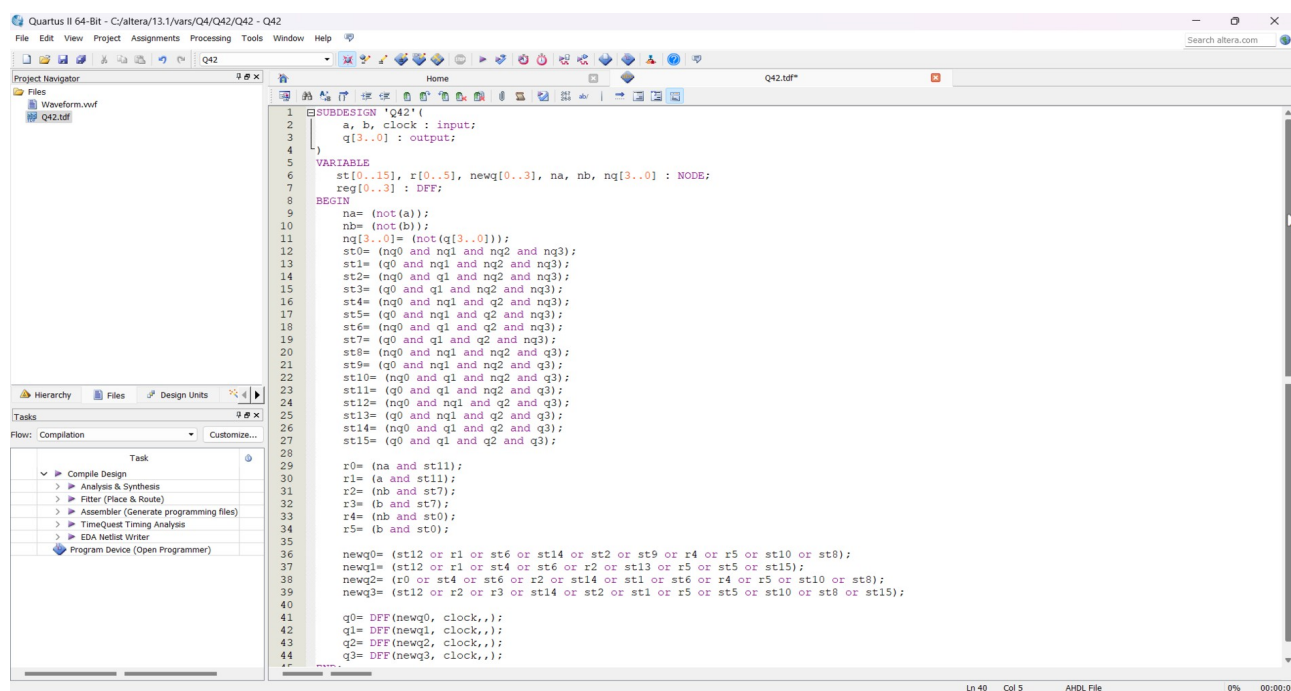


Рисунок 17 – Реализация цифровой схемы на языке AHDL

Полученный граф переходов на диаграмме полностью совпадает с заданным графом переходов состояний счётчика персонального варианта (рис. 18).

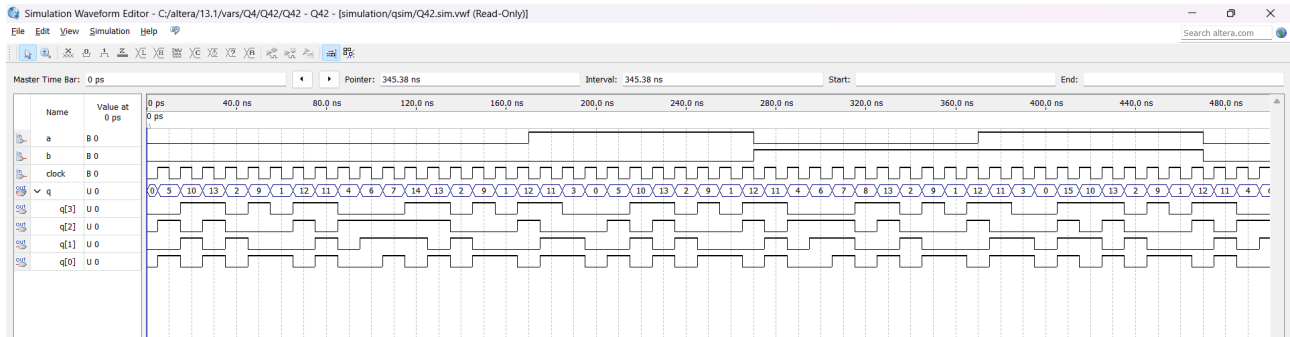


Рисунок 18 - Результаты моделирования работы схемы в сигнальном редакторе

ВЫВОД

В ходе практической работы было выполнено описание счетчика на языке AHDL. Это позволило углубить понимание принципов работы счетчиков и освоить инструменты для их создания и анализа. Результаты работы подтвердили эффективность использования AHDL для описания и моделирования счетчиков. Это дает основание для дальнейшего изучения и применения этих инструментов в будущих проектах.

Практическая работа №8

Тема: Виртуализация: установка и настройка операционной системы

Ubuntu Linux на виртуальной машине Oracle VirtualBox

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 Linux.

В результате выполнения практической работы студенты познакомятся с процессом установки на персональный компьютер виртуальной машины Oracle VirtualBox, получают представление о процессе создания и настройки виртуального окружения. На примере операционной системы Ubuntu Linux будет выполнен процесс установки и базовой настройки операционной системы.

Постановка задачи

Перед выполнением практической работы следует ознакомиться с архитектурой операционной системы Linux, представленной в приложении. По окончании необходимо ответить на контрольные вопросы по теме практической работы.

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 Ubuntu Linux. 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

Полученный установщик с сайта Oracle VirtualBox <https://www.virtualbox.org/wiki/Downloads> (рис. 19-20).



Рисунок 19 – Сайт Oracle



Рисунок 20 – Установщик виртуальной машины

Были установлены все предложенные компоненты (рис. 21).

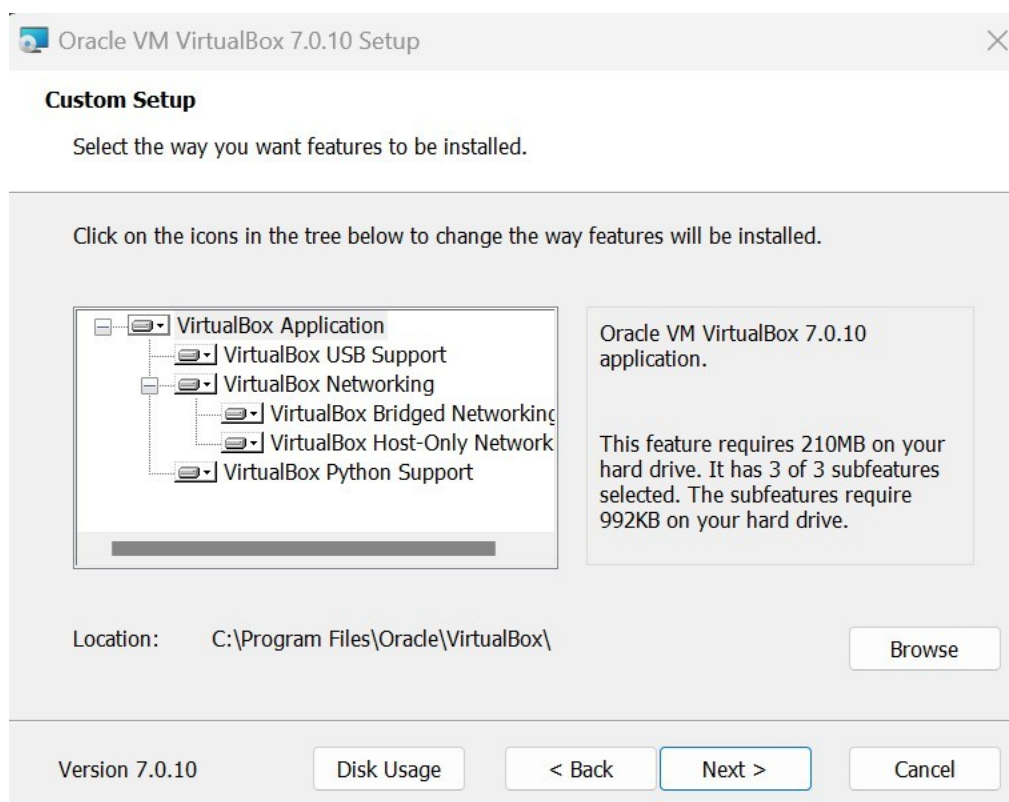


Рисунок 21 – Установка всех предложенных компонентов

По завершении установки на рабочем столе компьютера был создан ярлык для запуска виртуальной машины OracleVirtualBox (рис. 22).



Рисунок 22 - Завершение установки виртуальной машины

Далее был скачен образ установочного диска с сайта <http://www.ubuntu.ru/> (рис. 23-24)

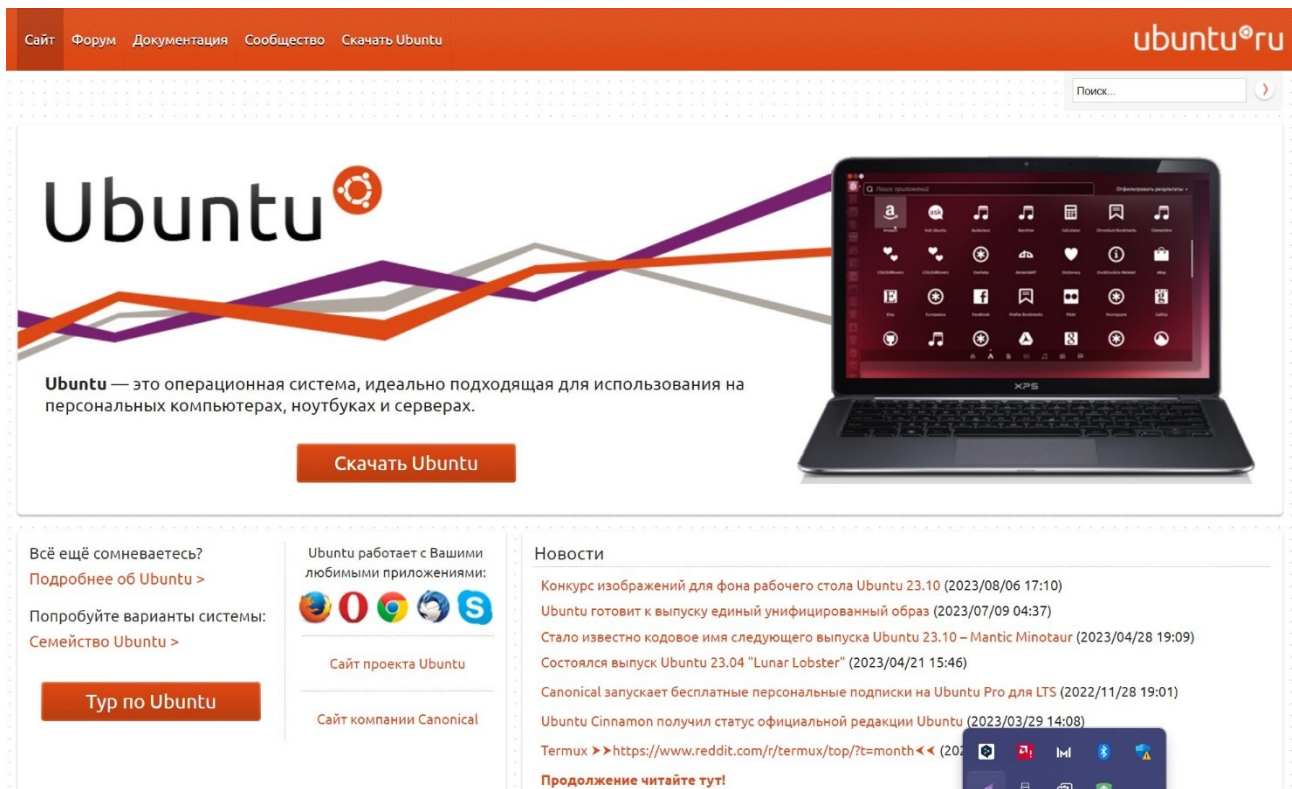


Рисунок 23 – Сайт Ubuntu 1

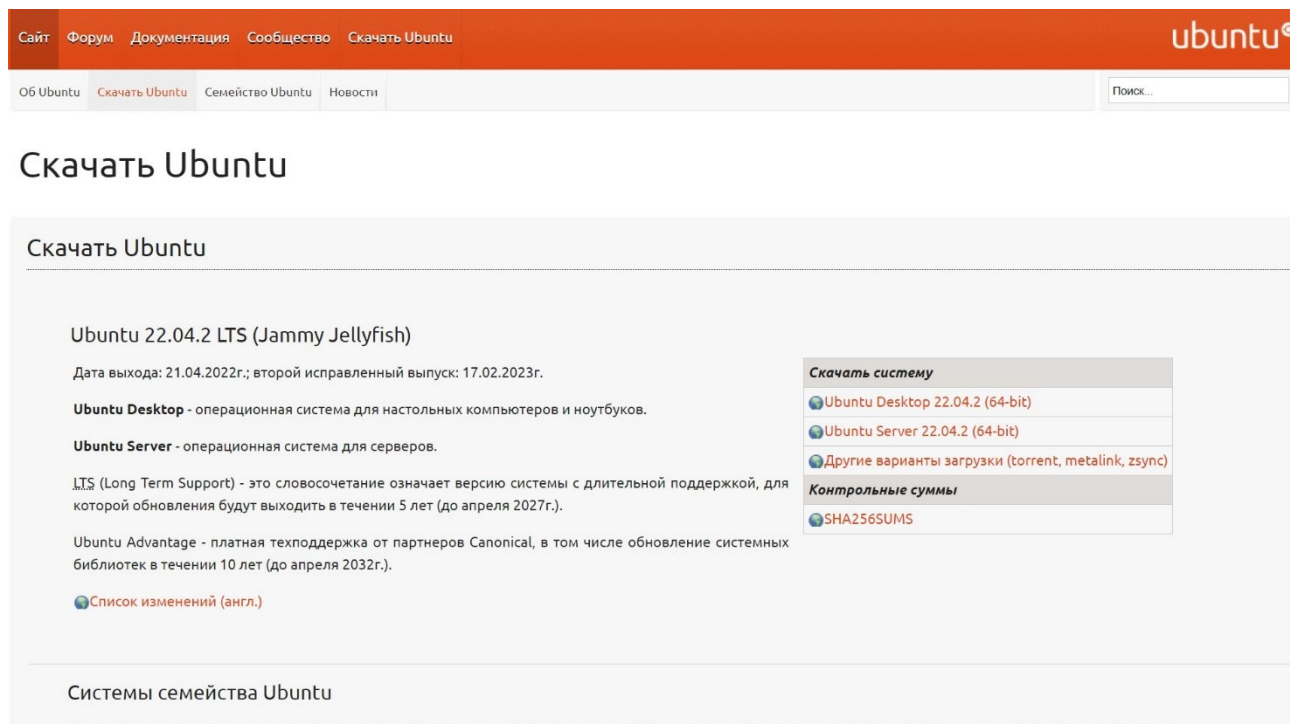


Рисунок 24 – Сайт Ubuntu 2

Далее была произведена установка и настройка операционной системы (рис. 25-31).

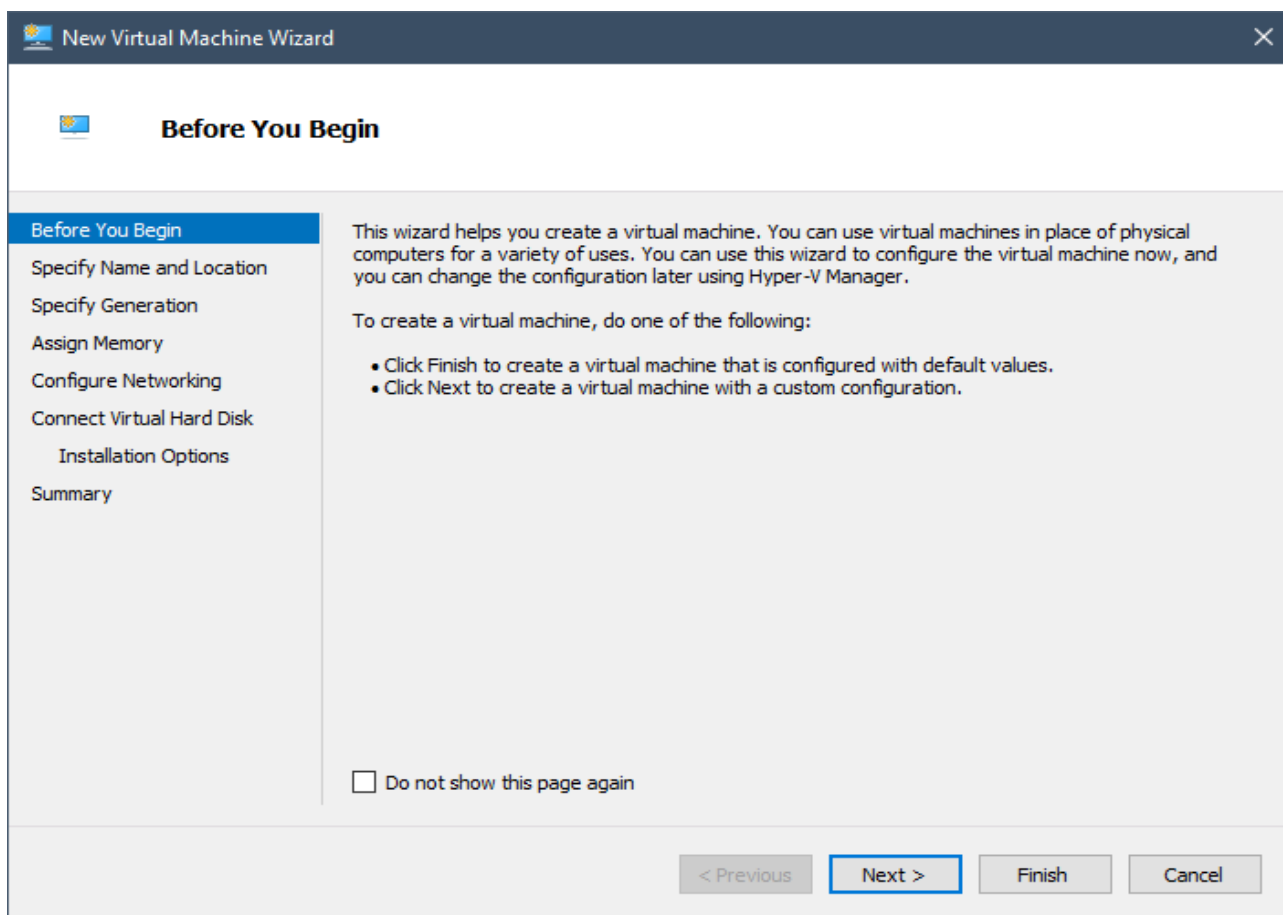


Рисунок 25 – Создание виртуальной машин

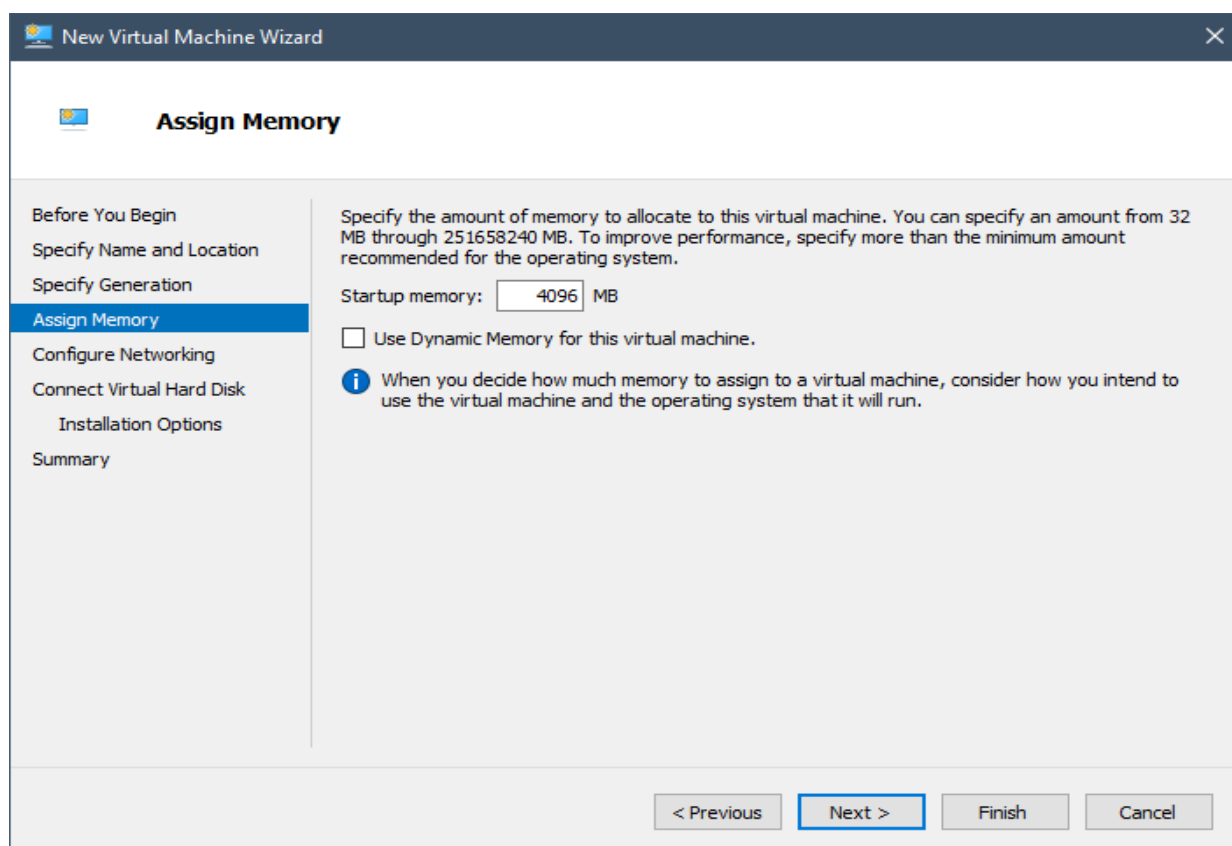


Рисунок 27 – Выделение кол-ва оперативной памяти и кол-во ЦП

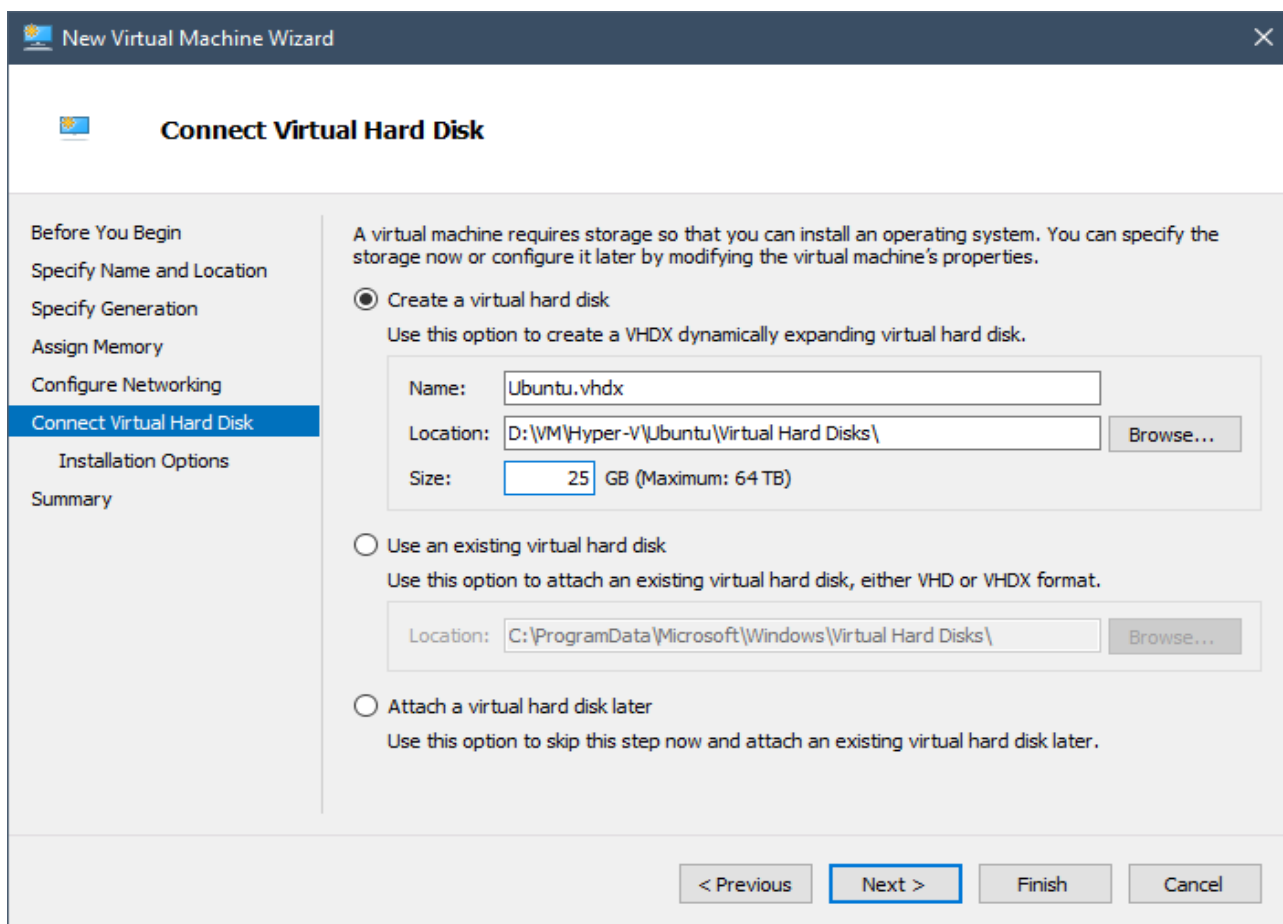


Рисунок 28 – Выделение места на твердотельном накопителе

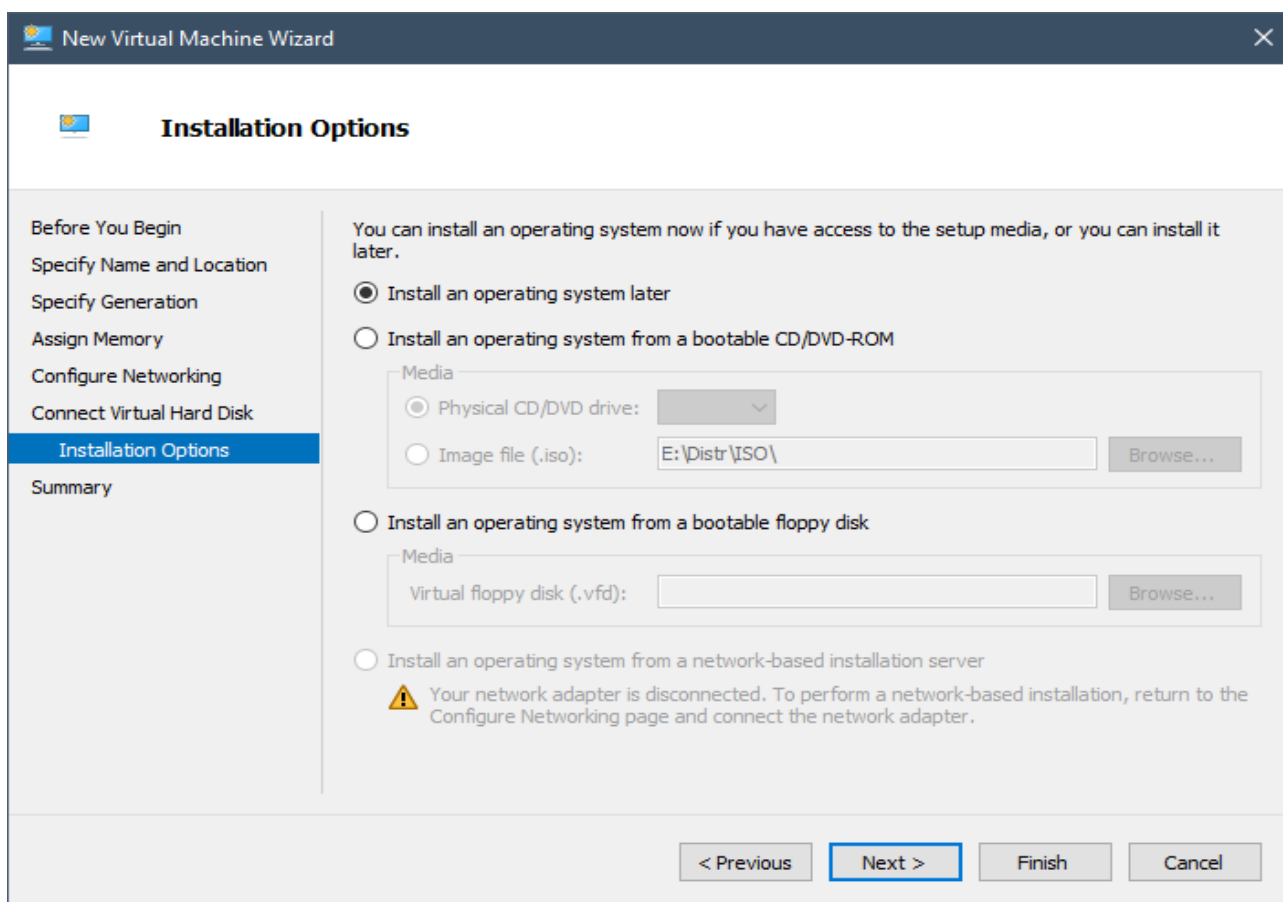


Рисунок 29 – Создание

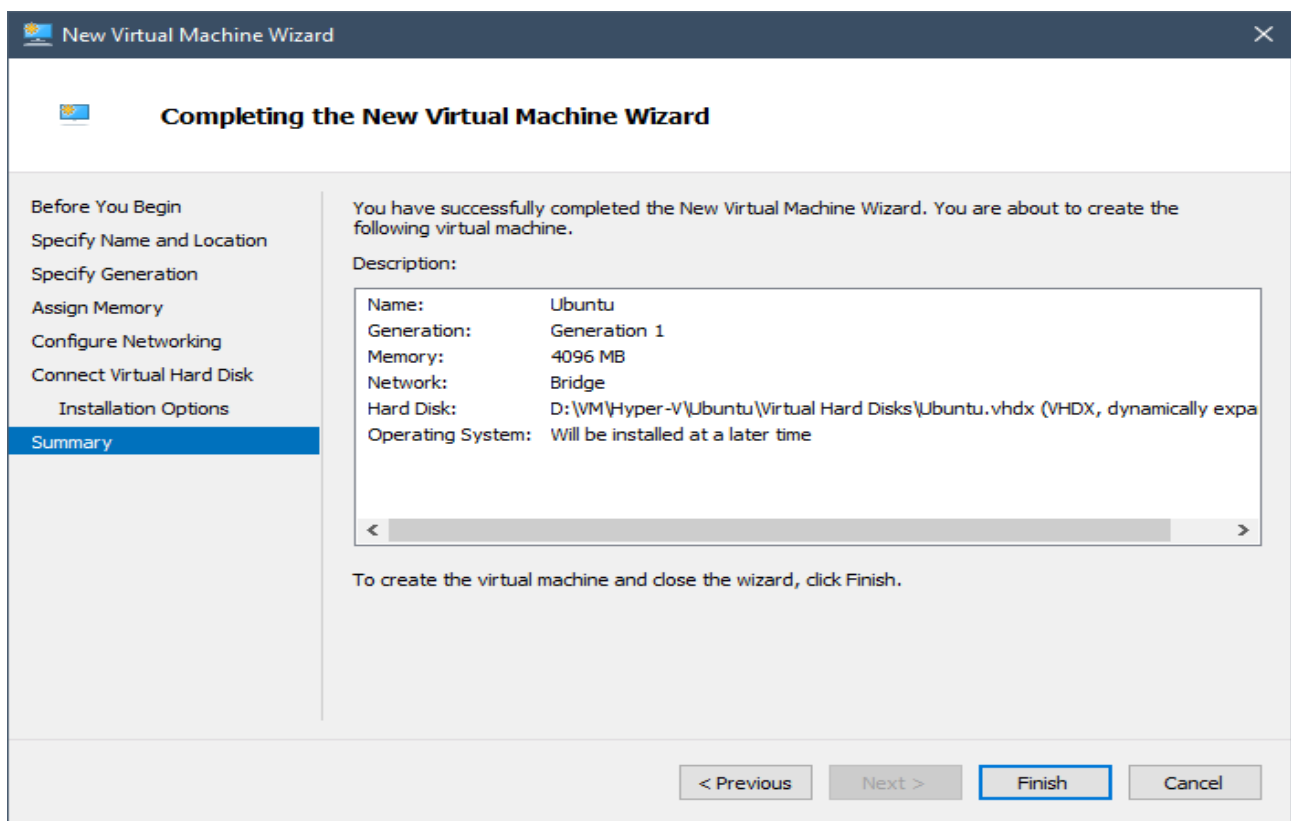


Рисунок 30 - Итог установки ОС 1

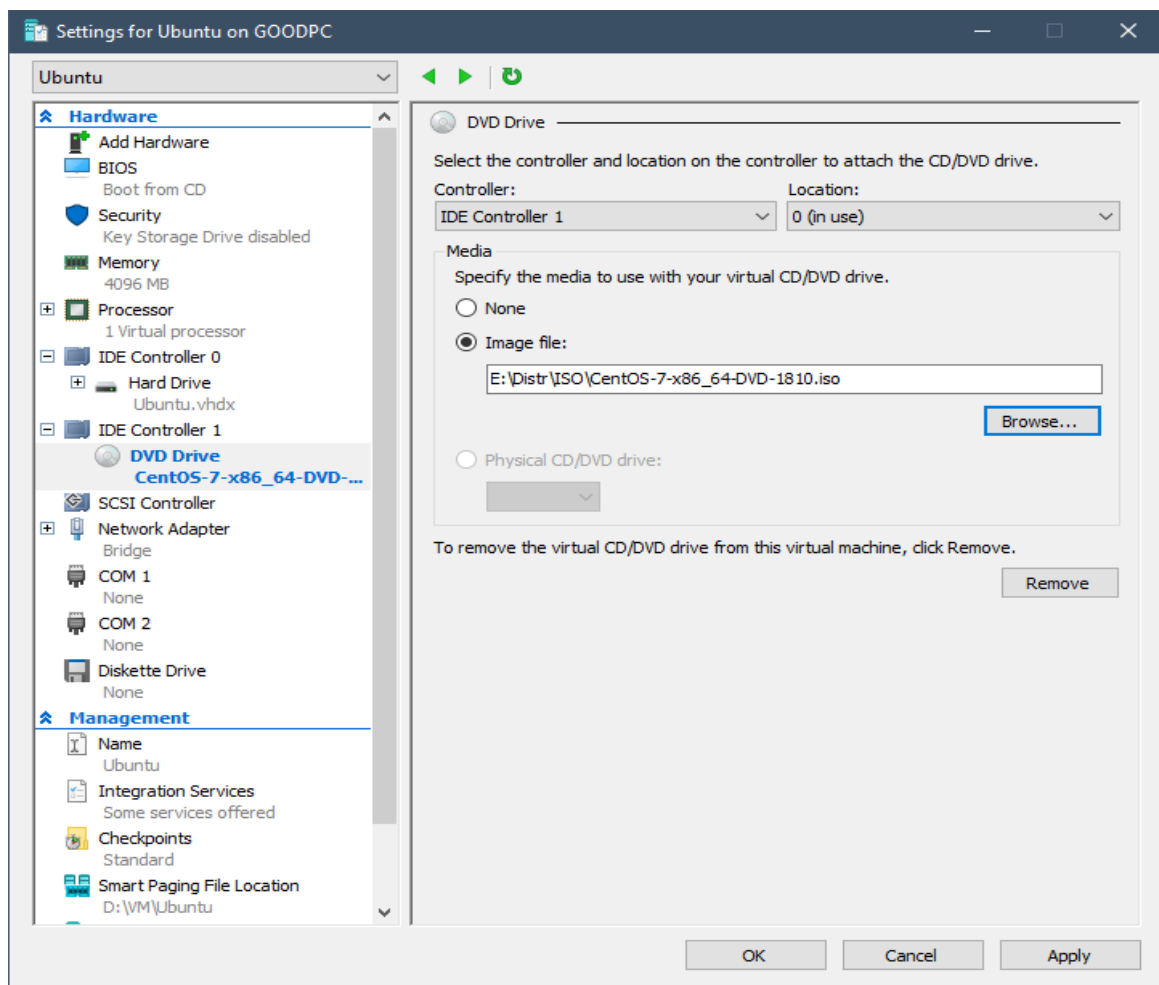


Рисунок 31 – Подключение образа диска с Ubuntu

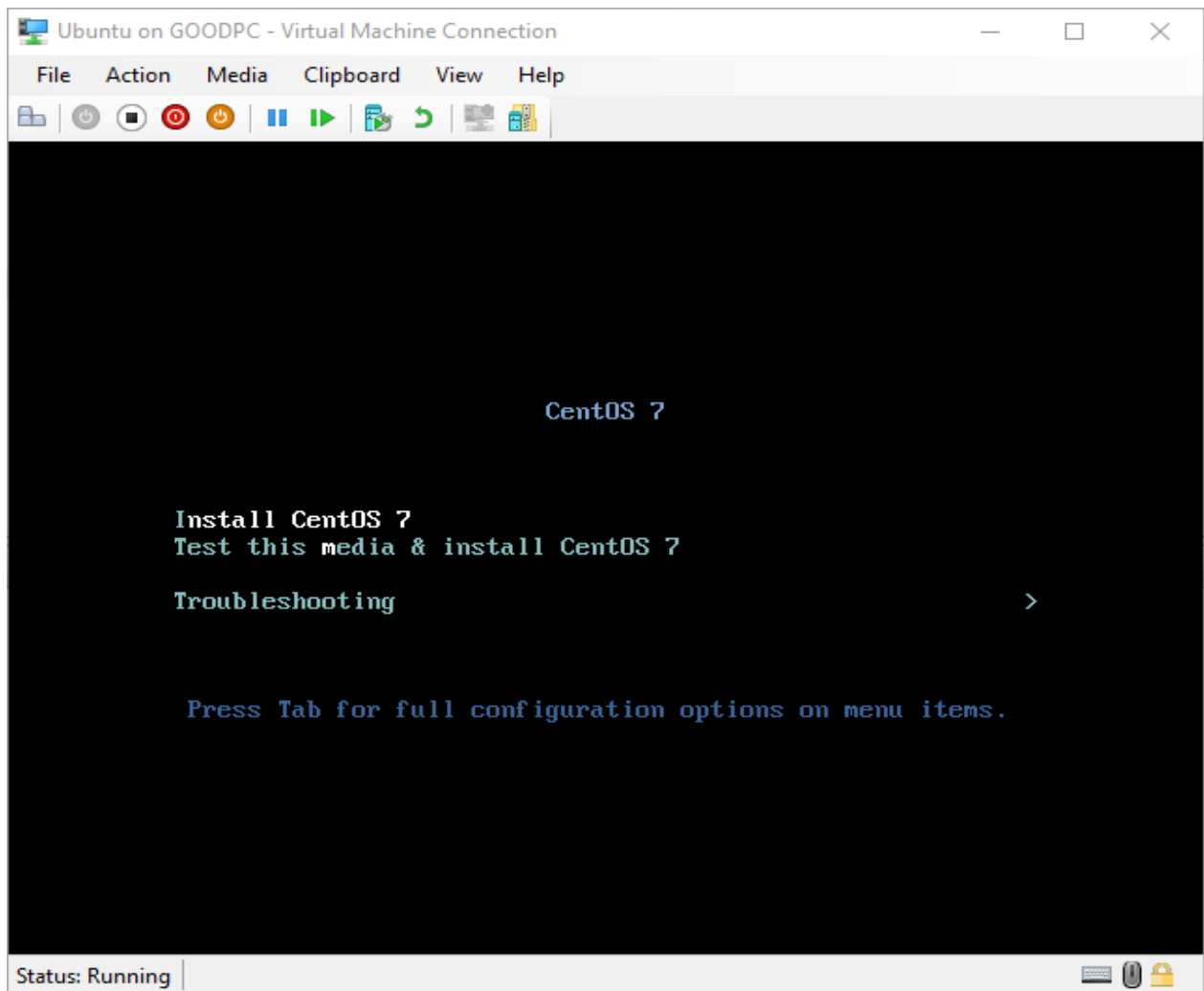


Рисунок 32 - Первый запуск системы



Рисунок 33 - Рабочий стол функционирующей системы

ВЫВОД

В ходе выполнения данной практической работы были получены практические навыки установки и создания виртуальных машин в Oracle VirtualBox. Были изучены принципы инсталляции и начальной настройки операционной системы Ubuntu Linux. Это позволило глубже понять принципы работы виртуальных машин и операционных систем, а также приобрести навыки, которые будут полезны в дальнейшей профессиональной деятельности.

Практическая работа №9

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

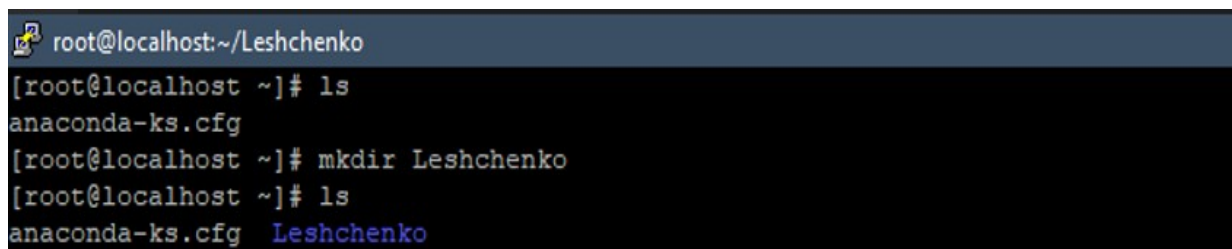
Цель работы: Практическая работа выполняется в среде, установленной и настроенной в процессе выполнения практической работы №1 или в среде, установленной в компьютерном классе. Целью данной практической работы является изучение команд операционной системы GNU Linux по работе с элементами файловой системы, а также получение практических навыков создания, изменения, манипулирования и удаления файлов и каталогов. В результате выполнения практической работы студенты познакомятся с процессом создания структуры каталогов, изучат различные способы создания и манипулирования данными. На примере созданной в процессе практической работы базы данных на основе текстовых файлов будут рассмотрены вопросы сортировки и фильтрации информации, вывод требуемых данных на экран и в файл.

Персональный вариант:

4 Фотоаппарат (название фотоаппарата, год выпуска, количество мегапикселей, характеристика зума). Поиск по

Ход работы

Создание родительского каталога названного фамилией (рис. 34).



```
root@localhost:~/Leshchenko
[root@localhost ~]# ls
anaconda-ks.cfg
[root@localhost ~]# mkdir Leshchenko
[root@localhost ~]# ls
anaconda-ks.cfg  Leshchenko
```

Рисунок 34 – Создание родительского каталога

Создание структуры каталогов в соответствии с заданной структурой (рис. 35-36).

```
[root@localhost ~]# cd Leshchenko/
[root@localhost Leshchenko]# mkdir database
[root@localhost Leshchenko]# mkdir temp
[root@localhost Leshchenko]# cd temp/
[root@localhost temp]# mkdir report
[root@localhost temp]# ls
report
```

Рисунок 35 – Создание структуры каталогов

```
[root@localhost temp]# cd ..
[root@localhost Leshchenko]# tree
.
├── database
├── temp
│   └── report
```

Рисунок 36 – Вывод дерева с помощью утилиты tree

Переходим в каталог temp, убеждаемся что он текущий и выводим содержимое (рис. 37).

```
[root@localhost temp]# ls
report
```

Рисунок 37 – Переход в каталог temp

С помощью встроенного текстового редактора внутри каталога temp создан файл базы данных dataset1.txt. Файл заполнен данными в соответствии с номером варианта задания. В качестве разделителя столбцов данных в файле использован символ “;” без пробелов (рис. 38).

```
root@localhost:~/Leshchenko/temp
GNU nano 2.3.1 File: dataset1.txt
CANON;2020;64;10
NIKON;2021;40;8
FUJIFILM;2022;48;12
CANON;2015;32;6
```

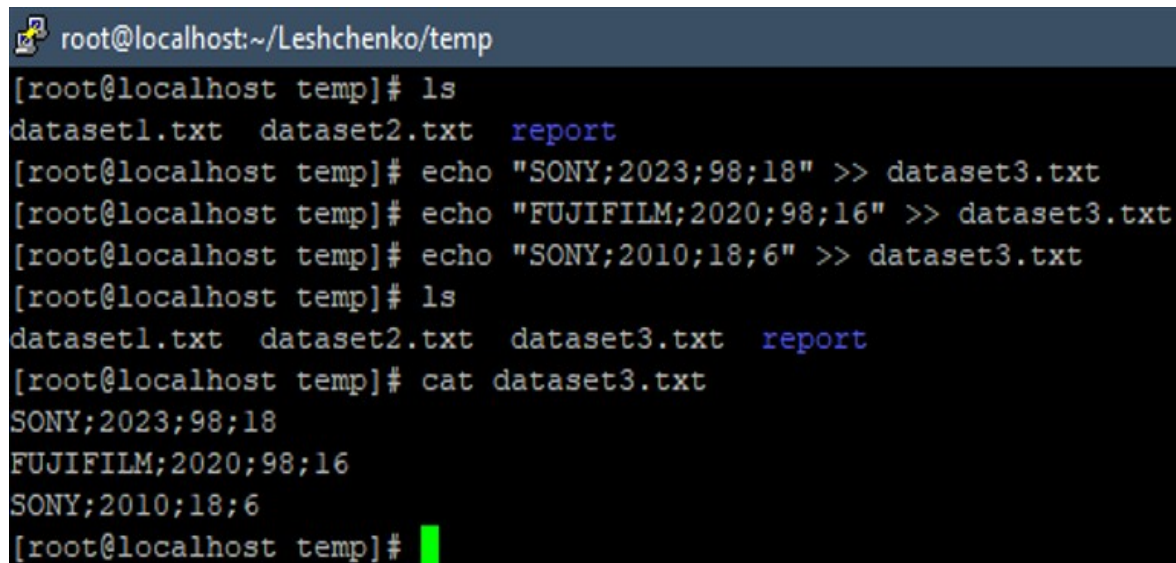
Рисунок 38 – Содержимое файла

С помощью конвейера команд внутри каталога temp создан файл базы данных dataset2.txt. Файл заполнен данными в соответствии с номером варианта задания. В качестве разделителя столбцов данных в файле использован символ “;” без пробелов (рис. 40).

```
root@localhost:~/Leshchenko/temp
[root@localhost temp]# cat > dataset2.txt << EOF
NIKON;2019;32;5
CANON;2010;12;3
FUJIFILM;2009;8;3
FUJIFILM;2018;98;12
EOF
[root@localhost temp]# ls
dataset1.txt  dataset2.txt  report
[root@localhost temp]# cat dataset2.txt
NIKON;2019;32;5
CANON;2010;12;3
FUJIFILM;2009;8;3
FUJIFILM;2018;98;12
[root@localhost temp]#
```

Рисунок 39 – создание dataset2.txt

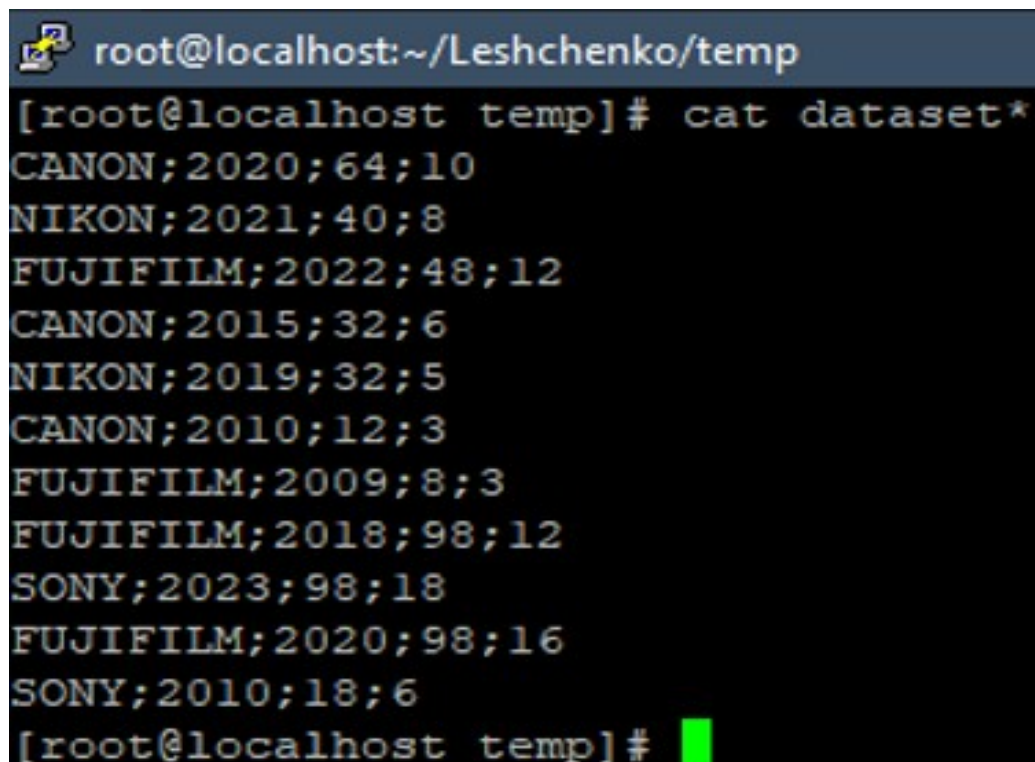
С помощью команды echo создан файл базы данных dataset3.txt. Заполнен файл данными в соответствии с номером варианта задания. В качестве разделителя столбцов данных в файле использован символ “;” без пробелов (рис. 40).



```
root@localhost:~/Leshchenko/temp
[root@localhost temp]# ls
dataset1.txt  dataset2.txt  report
[root@localhost temp]# echo "SONY;2023;98;18" >> dataset3.txt
[root@localhost temp]# echo "FUJIFILM;2020;98;16" >> dataset3.txt
[root@localhost temp]# echo "SONY;2010;18;6" >> dataset3.txt
[root@localhost temp]# ls
dataset1.txt  dataset2.txt  dataset3.txt  report
[root@localhost temp]# cat dataset3.txt
SONY;2023;98;18
FUJIFILM;2020;98;16
SONY;2010;18;6
[root@localhost temp]#
```

Рисунок 40 – Создание dataset3.txt

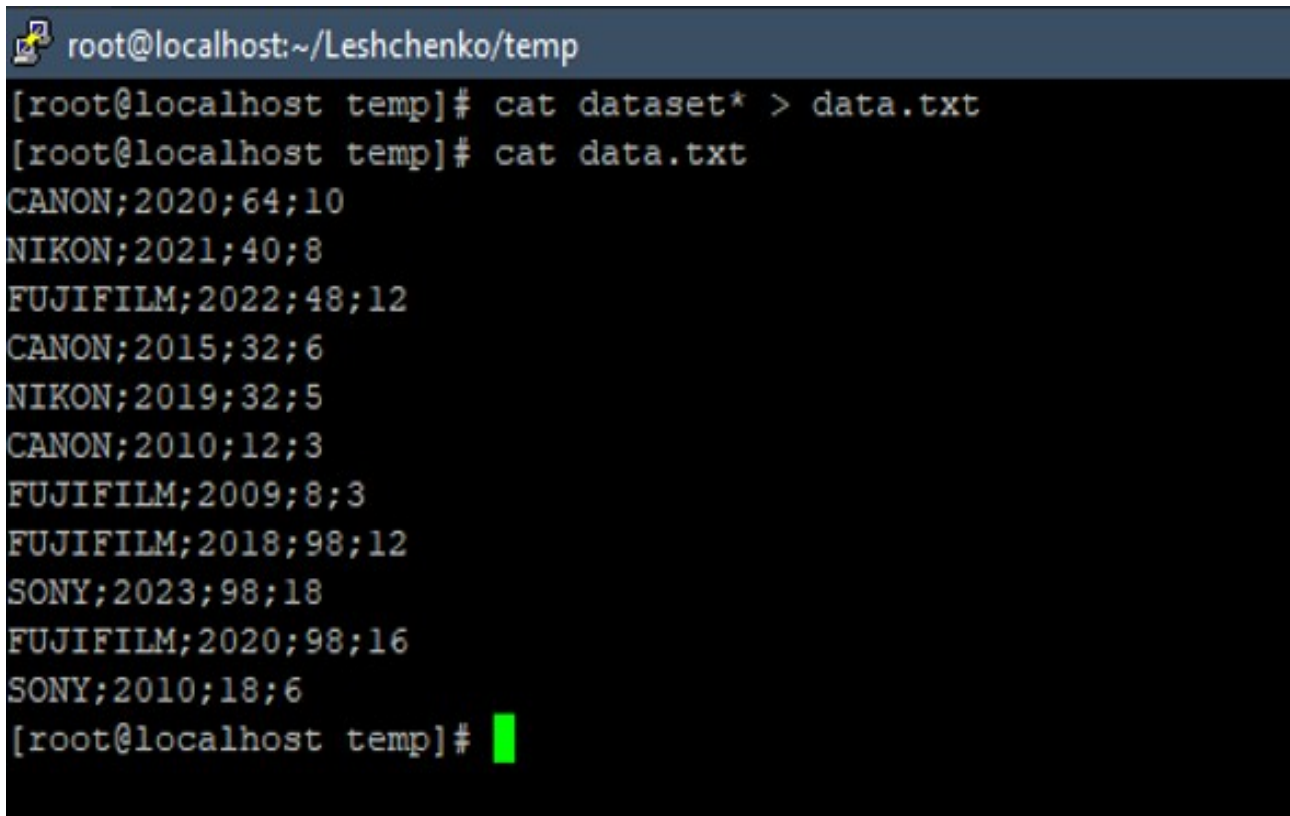
Вывод содержимого всех трёх файлов (рис. 41).



```
root@localhost:~/Leshchenko/temp
[root@localhost temp]# cat dataset*
CANON;2020;64;10
NIKON;2021;40;8
FUJIFILM;2022;48;12
CANON;2015;32;6
NIKON;2019;32;5
CANON;2010;12;3
FUJIFILM;2009;8;3
FUJIFILM;2018;98;12
SONY;2023;98;18
FUJIFILM;2020;98;16
SONY;2010;18;6
[root@localhost temp]#
```

Рисунок 41 – Вывод трёх файлов на экран

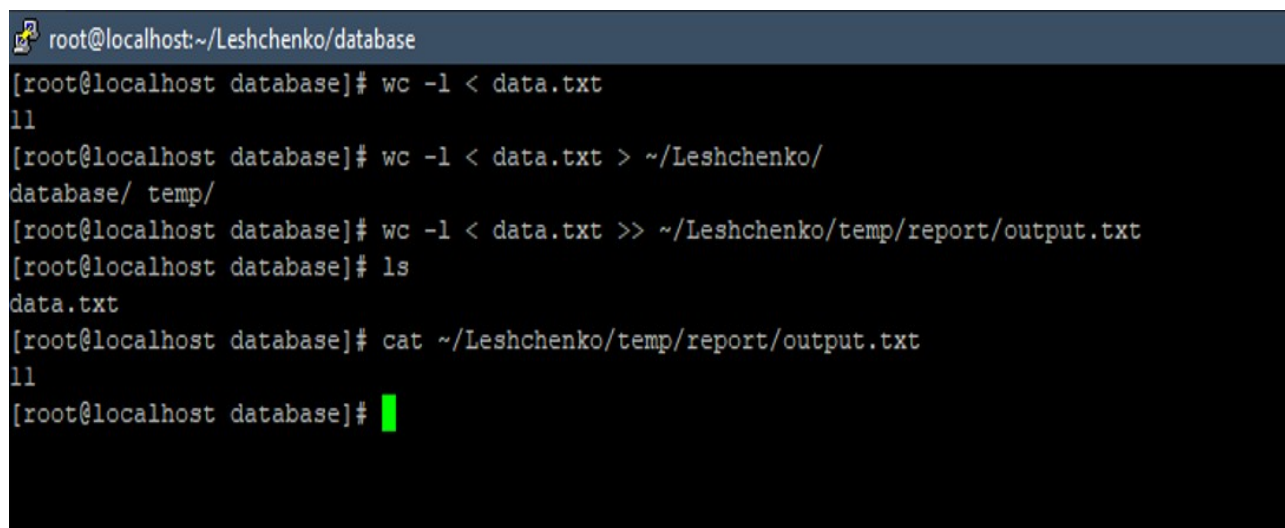
Далее объединено содержимое всех созданных файлов базы данных в один файл data.txt в каталог /database (рис. 43).

A terminal window with a dark background and light blue text. The prompt is 'root@localhost:~/Leshchenko/temp'. The user enters the command 'cat dataset* > data.txt'. The prompt changes to '[root@localhost temp]#'. The user enters 'cat data.txt'. The terminal displays the following concatenated data: CANON;2020;64;10, NIKON;2021;40;8, FUJIFILM;2022;48;12, CANON;2015;32;6, NIKON;2019;32;5, CANON;2010;12;3, FUJIFILM;2009;8;3, FUJIFILM;2018;98;12, SONY;2023;98;18, FUJIFILM;2020;98;16, SONY;2010;18;6. The prompt returns to '[root@localhost temp]#' with a red cursor.

```
root@localhost:~/Leshchenko/temp
[root@localhost temp]# cat dataset* > data.txt
[root@localhost temp]# cat data.txt
CANON;2020;64;10
NIKON;2021;40;8
FUJIFILM;2022;48;12
CANON;2015;32;6
NIKON;2019;32;5
CANON;2010;12;3
FUJIFILM;2009;8;3
FUJIFILM;2018;98;12
SONY;2023;98;18
FUJIFILM;2020;98;16
SONY;2010;18;6
[root@localhost temp]#
```

Рисунок 43 – Объединение и перемещение datasetX.txt

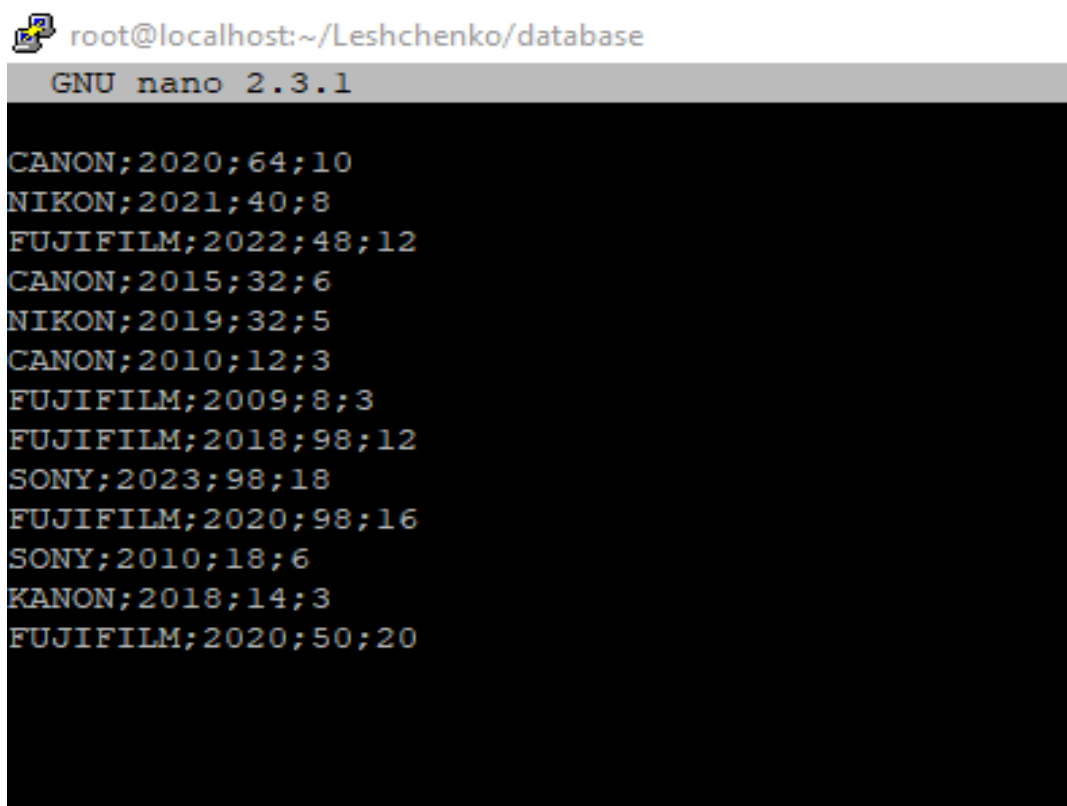
Подсчитаем количество строк файла data.txt. Результат подсчета выведен на экран и в файл отчета output.txt, расположенный в каталоге report (рис. 45).



```
root@localhost:~/Leshchenko/database
[root@localhost database]# wc -l < data.txt
11
[root@localhost database]# wc -l < data.txt > ~/Leshchenko/
database/ temp/
[root@localhost database]# wc -l < data.txt >> ~/Leshchenko/temp/report/output.txt
[root@localhost database]# ls
data.txt
[root@localhost database]# cat ~/Leshchenko/temp/report/output.txt
11
[root@localhost database]#
```

Рисунок 45 – Подсчёт строк в файле

Дополним файл data.txt двумя строками, и убедимся что файл содержит все необходимые данные (рис. 46).

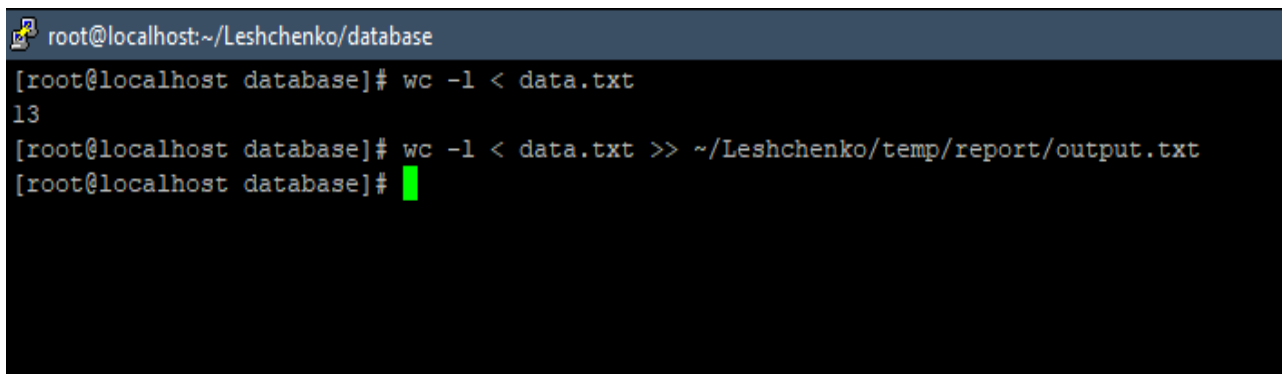


```
root@localhost:~/Leshchenko/database
GNU nano 2.3.1

CANON;2020;64;10
NIKON;2021;40;8
FUJIFILM;2022;48;12
CANON;2015;32;6
NIKON;2019;32;5
CANON;2010;12;3
FUJIFILM;2009;8;3
FUJIFILM;2018;98;12
SONY;2023;98;18
FUJIFILM;2020;98;16
SONY;2010;18;6
KANON;2018;14;3
FUJIFILM;2020;50;20
```

Рисунок 46 – Открываем файл для редактирования с помощью nano

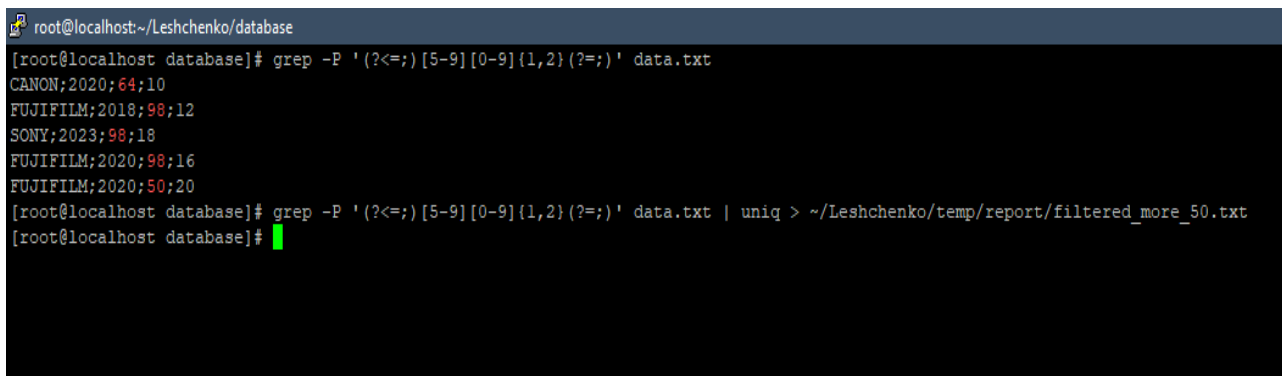
Подсчитаем количество строк файла data.txt. Результат подсчета выведен на экран и в конец файла отчета output.txt, расположенный в каталоге report (рис. 47).



```
root@localhost:~/Leshchenko/database
[root@localhost database]# wc -l < data.txt
13
[root@localhost database]# wc -l < data.txt >> ~/Leshchenko/temp/report/output.txt
[root@localhost database]#
```

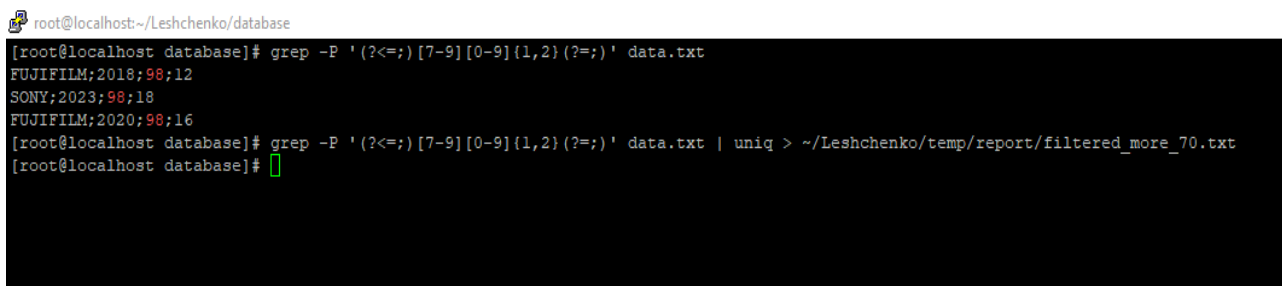
Рисунок 47 – Подсчёт количества строк

Осуществлена фильтрацию данных файла data.txt в соответствии с номером варианта задания. Результат фильтрации выведен на экран и в файл отчета filtered.txt, расположенный в каталоге report (рис. 48). Повторена фильтрация с различными значениями фильтра. Результаты фильтрации выведен на экран и дописан в файл отчета filtered.txt (рис. 49).



```
root@localhost:~/Leshchenko/database
[root@localhost database]# grep -P '(?<=;)[5-9][0-9]{1,2}(?=;)' data.txt
CANON;2020;64;10
FUJIFILM;2018;98;12
SONY;2023;98;18
FUJIFILM;2020;98;16
FUJIFILM;2020;50;20
[root@localhost database]# grep -P '(?<=;)[5-9][0-9]{1,2}(?=;)' data.txt | uniq > ~/Leshchenko/temp/report/filtered_more_50.txt
[root@localhost database]#
```

Рисунок 48 – Фильтрация 1



```
root@localhost:~/Leshchenko/database
[root@localhost database]# grep -P '(?<=;)[7-9][0-9]{1,2}(?=;)' data.txt
FUJIFILM;2018;98;12
SONY;2023;98;18
FUJIFILM;2020;98;16
[root@localhost database]# grep -P '(?<=;)[7-9][0-9]{1,2}(?=;)' data.txt | uniq > ~/Leshchenko/temp/report/filtered_more_70.txt
[root@localhost database]#
```

Рисунок 49 – Фильтрация 2

Выполнена сортировка содержимого файла data.txt в соответствии с номером варианта задания. Результат сортировки выведен на экран и в файл отчета sorted.txt, расположенный в каталоге report (рис. 50)

```
root@localhost:~/Leshchenko/database
[root@localhost database]# sort -t ';' -k 3 data.txt
CANON;2010;12;3
KANON;2018;14;3
SONY;2010;18;6
NIKON;2019;32;5
CANON;2015;32;6
NIKON;2021;40;8
FUJIFILM;2022;48;12
FUJIFILM;2020;50;20
CANON;2020;64;10
FUJIFILM;2009;8;3
FUJIFILM;2018;98;12
FUJIFILM;2020;98;16
SONY;2023;98;18
[root@localhost database]# sort -t ';' -k 3 data.txt > ~/Leshchenko/temp/report/sorted.txt
[root@localhost database]#
```

Рисунок 50 – Сортировка файла data.txt

Выполнена фильтрация содержимого файла data.txt с сортировкой результата фильтрации. Фильтрация и сортировка выполнена в соответствии с номером варианта задания. Результат выведен на экран и в файл отчета filteredsorted.txt, расположенный в каталоге report (рис. 51).

```
root@localhost:~/Leshchenko/database
[root@localhost database]# grep -P '(?<=;)[5-9][0-9]{1,2}(?>=;)' data.txt | sort -t ';' -k 3
FUJIFILM;2020;50;20
CANON;2020;64;10
FUJIFILM;2018;98;12
FUJIFILM;2020;98;16
SONY;2023;98;18
[root@localhost database]# grep -P '(?<=;)[5-9][0-9]{1,2}(?>=;)' data.txt | sort -t ';' -k 3 > ~/Leshchenko/temp/report/filtered_more_50_sorted.txt
[root@localhost database]#
```

Рисунок 51 – Сортировка и фильтрация

Выполнена команда вывода календаря на экран и в файл calendar.txt, находящийся в каталоге /database. Результат выведен на экран (рис. 52).

```
root@localhost:~/Leshchenko/database
[root@localhost database]# cal > ~/Leshchenko/database/calendar.txt
[root@localhost database]# cat ~/Leshchenko/database/calendar.txt
    December 2023
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
[root@localhost database]#
```

Рисунок 52 – Календарь

ВЫВОД

В ходе выполнения данной практической работы были изучены команды операционной системы GNU Linux для работы с элементами файловой системы. Были получены практические навыки создания, изменения, манипулирования и удаления файлов и каталогов. Получено 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. Это позволило глубже понять принципы работы с файловой системой в операционной системе GNU Linux, а также приобрести навыки, которые будут полезны в дальнейшей профессиональной деятельности.

Практическая работа №10

Тема: Использование программируемого фильтра awk

Цель работы: Практическая работа выполняется в среде, установленной и настроенной в процессе выполнения практической работы №1 или в среде, установленной в компьютерном классе. Целью данной практической работы является изучение возможностей программируемого фильтра AWK при обработке текстовой информации. В результате выполнения практической работы студенты получают практические навыки манипулирования данными средствами awk, составления правил обработки потоков информации, формирования отчетов и извлечения требуемой информации из большого массива данных.

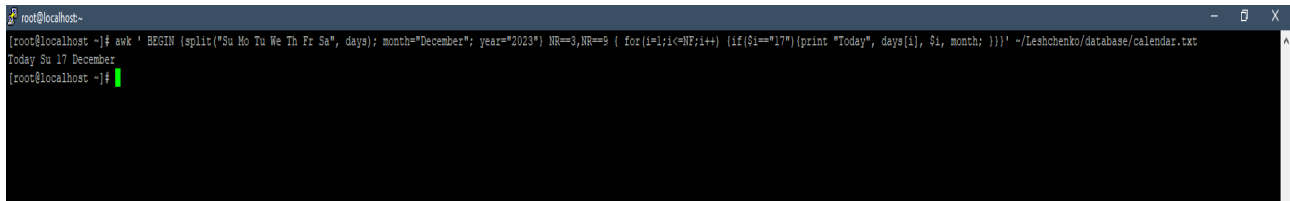
Постановка задачи:

Используя AWK:

1. вывести на экран из файла calendar.txt день недели и текущее число в виде «сегодня вторник ... августа»;
2. вывести список каталогов, имена которых состоят из русских букв, без дополнительных полей;
3. определить количество(сумму) байтов, занятых всеми вашими текстовыми файлами (txt) в каталогах и подкаталогах;
4. определить количество блоков, содержащих ваш текущий каталог; 35
5. изменить права доступа для некоторых файлов текущего каталога и провести сортировку списка по возможностям доступа;
6. напечатать список каталогов, в которых обнаружены файлы с именами data*.txt;
7. подсчитать, сколько раз пользователь входил в систему;
8. напечатать список пользователей, отсортированный по времени.

Ход работы

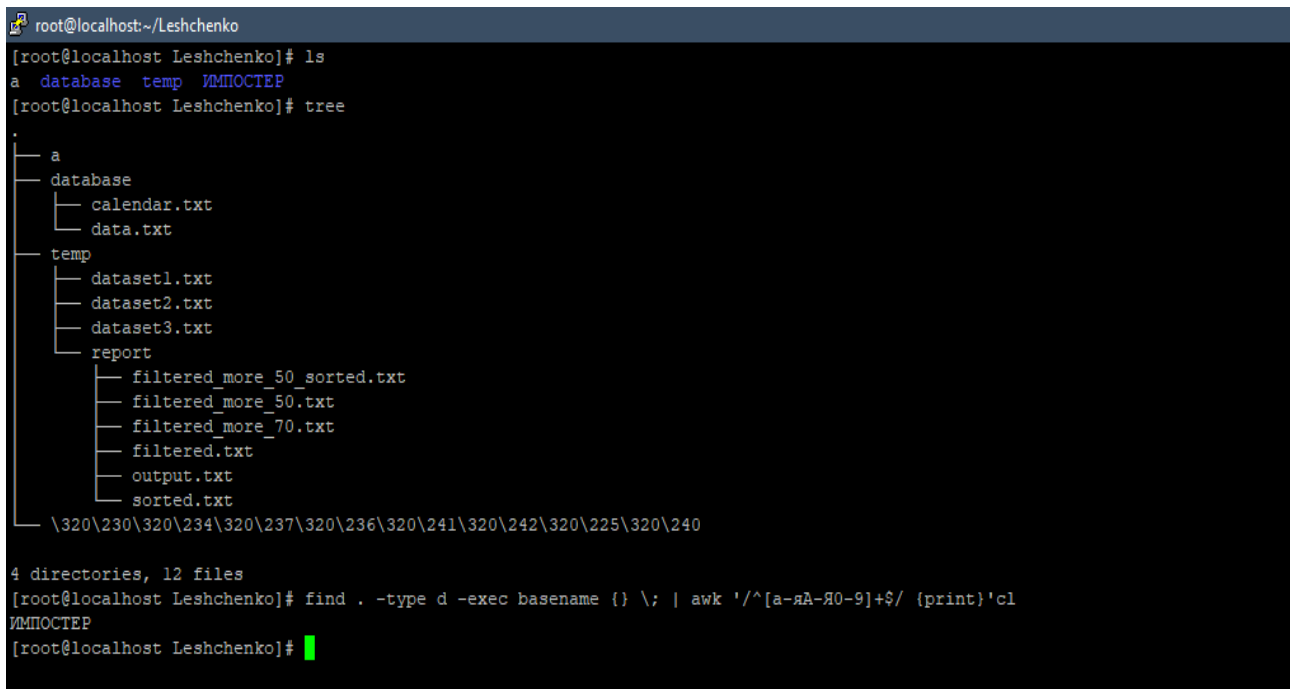
Вывод на экран из файла calendar.txt дня недели и текущего числа в виде «сегодня вторник ... августа»; (рис. 53).



```
root@localhost:~# awk 'BEGIN {split("Su Mo Tu We Th Fr Sa", days); month="December"; year="2023"} NR==3,NR==9 { for(i=1;i<NF;i++) {if($i=="17"){print "Today", days[i], $i, month; }}}' ~/Leshchenko/database/calendar.txt
Today Su 17 December
root@localhost:~#
```

Рисунок 53 – Вывод даты

Вывод списка каталогов, имена которых состоят из русских букв, без дополнительных полей (рис. 54).



```
root@localhost:~/Leshchenko
[root@localhost Leshchenko]# ls
a database temp ИМПОСТЕР
[root@localhost Leshchenko]# tree
.
├── a
├── database
│   ├── calendar.txt
│   └── data.txt
├── temp
│   ├── dataset1.txt
│   ├── dataset2.txt
│   ├── dataset3.txt
│   └── report
│       ├── filtered_more_50_sorted.txt
│       ├── filtered_more_50.txt
│       ├── filtered_more_70.txt
│       ├── filtered.txt
│       ├── output.txt
│       └── sorted.txt
└── \320\230\320\234\320\237\320\236\320\241\320\242\320\225\320\240

4 directories, 12 files
[root@localhost Leshchenko]# find . -type d -exec basename {} \; | awk '/^[а-яА-Я0-9]+$/' {print}'cl
ИМПОСТЕР
[root@localhost Leshchenko]#
```

Рисунок 54 – Вывод каталогов имена которых состоят из русских букв

Определение количества(суммы) байтов, занятых всеми текстовыми файлами (txt) в каталогах и подкаталогах (рис. 55).

```
root@localhost:~/Leshchenko
[root@localhost Leshchenko]# ls -lR
.:
итого 0
drwxr-xr-x. 2 root root 42 дек 17 16:09 database
drwxr-xr-x. 3 root root 98 ноя 26 22:28 temp
drwxr-xr-x. 2 root root 6 дек 17 16:23 ИМПОСТЕР

./database:
итого 8
-rw-r--r--. 1 root root 150 дек 17 16:09 calendar.txt
-rw-r--r--. 1 root root 226 ноя 26 22:33 data.txt

./temp:
итого 12
-rw-r--r--. 1 root root 69 ноя 26 22:20 dataset1.txt
-rw-r--r--. 1 root root 70 ноя 26 22:23 dataset2.txt
-rw-r--r--. 1 root root 51 ноя 26 22:25 dataset3.txt
drwxr-xr-x. 2 root root 153 дек 17 16:02 report

./temp/report:
итого 24
-rw-r--r--. 1 root root 93 дек 17 16:05 filtered_more_50_sorted.txt
-rw-r--r--. 1 root root 93 дек 17 15:59 filtered_more_50.txt
-rw-r--r--. 1 root root 56 дек 17 16:02 filtered_more_70.txt
-rw-r--r--. 1 root root 226 ноя 26 23:13 filtered.txt
-rw-r--r--. 1 root root 6 дек 17 15:58 output.txt
-rw-r--r--. 1 root root 226 дек 17 16:04 sorted.txt

./ИМПОСТЕР:
итого 0
[root@localhost Leshchenko]# ls -lR | awk '/\.txt$/ {sum += $5} END {print "Bytes: " sum;}'
Bytes: 1266
[root@localhost Leshchenko]#
```

Рисунок 55 – Подсчёт кол-ва байтов

Определение количества блоков, содержащихся в текущем каталоге (рис. 56).

```
root@localhost:~/Leshchenko
[root@localhost Leshchenko]# ls -ls | awk '{total += $1} END {print "Theres " total " blocks"}'
Theres 0 blocks
[root@localhost Leshchenko]# ls -ls
итого 0
0 drwxr-xr-x. 2 root root 42 дек 17 16:09 database
0 drwxr-xr-x. 3 root root 98 ноя 26 22:28 temp
0 drwxr-xr-x. 2 root root 6 дек 17 16:23 ИМПОСТЕР
[root@localhost Leshchenko]#
```

Рисунок 56 – Определение кол-ва блоков

Изменены права доступа для некоторых файлов текущего каталога и проведена сортировка списка по возможностям доступа (рис. 57).


```

root@localhost:~/Leshchenko/temp/report
[root@localhost report]# ls -l
итого 24
-rwxrw-r--. 1 root root 93 дек 17 16:05 filtered_more_50_sorted.txt
-rw-r--r--. 1 root root 93 дек 17 15:59 filtered_more_50.txt
-rw-r--r--. 1 root root 56 дек 17 16:02 filtered_more_70.txt
-rw-r--r--. 1 root root 226 ноя 26 23:13 filtered.txt
-rw-r--r--. 1 root root 6 дек 17 15:58 output.txt
-rw-r--r--. 1 root root 226 дек 17 16:04 sorted.txt
[root@localhost report]# ls -l | awk '{print $1, $9}' | sort -k 10
-rw-r--r--. filtered_more_50.txt
-rw-r--r--. filtered_more_70.txt
-rw-r--r--. filtered.txt
-rw-r--r--. output.txt
-rw-r--r--. sorted.txt
-rwxrw-r--. filtered_more_50_sorted.txt
итого
[root@localhost report]#

```

Рисунок 57 – Изменение уровня доступа

Напечатаны списки каталогов, в которых обнаружены файлы с именами data*.txt (рис. 58).

```

borzovdmityriyolegovich@ubuntu: ~/borzov
borzovdmityriyolegovich@ubuntu:~/borzov$ tree
.
├── database
│   ├── calendar.txt
│   └── data.txt
├── temp
│   ├── dataset1.txt
│   ├── dataset2.txt
│   ├── dataset3.txt
│   └── report
│       ├── filtered_200000.txt
│       ├── filtered_300000.txt
│       ├── filteredsorted.txt
│       ├── filtered.txt
│       ├── output.txt
│       └── sorted
├── Папка2
└── Папка1

5 directories, 11 files
borzovdmityriyolegovich@ubuntu:~/borzov$ find . -type f -name 'data*.txt' | awk -F/ '!seen[$2]++ {print $2}'
database
temp
borzovdmityriyolegovich@ubuntu:~/borzov$

```

Рисунок 58 - Вывод каталогов, содержащих data*.txt

Подсчёт кол-ва входов пользователей в систему (рис. 59).

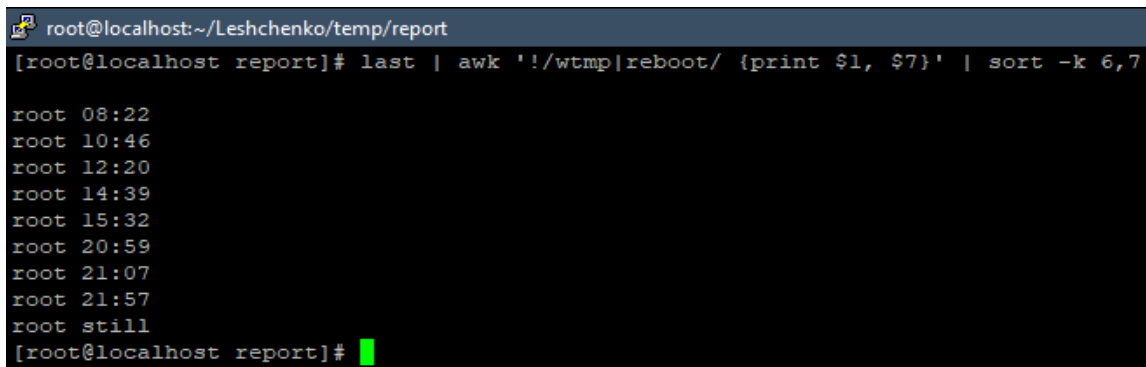
```

root@localhost:~/Leshchenko/temp/report
[root@localhost report]# last | awk '!/wtmp|reboot/ {users[$1]++} END {for (user in users) print user, users[user]}'
1
root 9
[root@localhost report]#

```

Рисунок 59 – Подсчёт входов в систему

Вывод списка пользователей, отсортированного по времени (рис. 60).



A terminal window with a dark background and a blue title bar. The title bar contains the text 'root@localhost:~/Leshchenko/temp/report'. The terminal shows a command being executed: '[root@localhost report]# last | awk '!/wtmp|reboot/ {print \$1, \$7}' | sort -k 6,7'. The output of the command is a list of user login times for the 'root' user, sorted chronologically: 'root 08:22', 'root 10:46', 'root 12:20', 'root 14:39', 'root 15:32', 'root 20:59', 'root 21:07', 'root 21:57', and 'root still'. The prompt '[root@localhost report]#' is visible at the bottom, followed by a green cursor.

```
root@localhost:~/Leshchenko/temp/report
[root@localhost report]# last | awk '!/wtmp|reboot/ {print $1, $7}' | sort -k 6,7
root 08:22
root 10:46
root 12:20
root 14:39
root 15:32
root 20:59
root 21:07
root 21:57
root still
[root@localhost report]#
```

Рисунок 60 – Пользователи отсортированные по времени

ВЫВОД

В ходе выполнения данной практической работы были изучены возможности программируемого фильтра AWK при обработке текстовой информации. Были получены практические навыки манипулирования данными средствами awk, составления правил обработки потоков информации, формирования отчетов и извлечения требуемой информации из большого массива данных. Это позволило глубже понять принципы работы с AWK и приобрести навыки, которые будут полезны в дальнейшей профессиональной деятельности.

СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. Браммер Ю.А. Пащук И.Н. Цифровые устройства
2. Архитектура_ВМиС_Учебное_пособие.pdf
3. Архитектура ВМиС_Метод.рекомендации к контрольным работам.pdf
4. Практические работы ВМ_АрхВМиС_ред_Пономарев.pdf
5. Лекции Штрекер Е.Н