



МИНОБРНАУКИ РОССИИ

*Федеральное государственное бюджетное образовательное учреждение
высшего образования*

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий

Кафедра Вычислительной Техники (ВТ)

Отчет по выполнению лабораторной работы
По дисциплине «Архитектуры вычислительных машин и систем»

Выполнил студент группы ИНБО-04-20

Ло В.Х.

Принял старший преподаватель

Гололобов А.А.

Лабораторные работы

« » ____ 2021 г

выполнены

«Зачтено»

«__» ____ 2021 г

Москва 2021

Оглавление

Лабораторная работа № 1.....	3
Лабораторная работа № 2.....	4
Лабораторная работа № 3.....	5
Лабораторная работа № 4.....	7
Лабораторная работа № 5.....	9
Лабораторная работа № 6.....	10
Лабораторная работа № 7.....	15
Лабораторная работа № 8.....	21
Лабораторная работа № 9.....	37
Лабораторная работа №10.....	44

Лабораторная работа №1

Графический ввод схемы и симуляция в САПР MAX+PLUS II

Цель работы:

Спроектировать логическую схему при помощи графического редактора САПР MAX+PLUS II. Исследовать работу схемы с использованием сигнального редактора САПР MAX+PLUS II.

$$Y = AD + \bar{C} + BD + \bar{D} \oplus ACB$$

Схема:

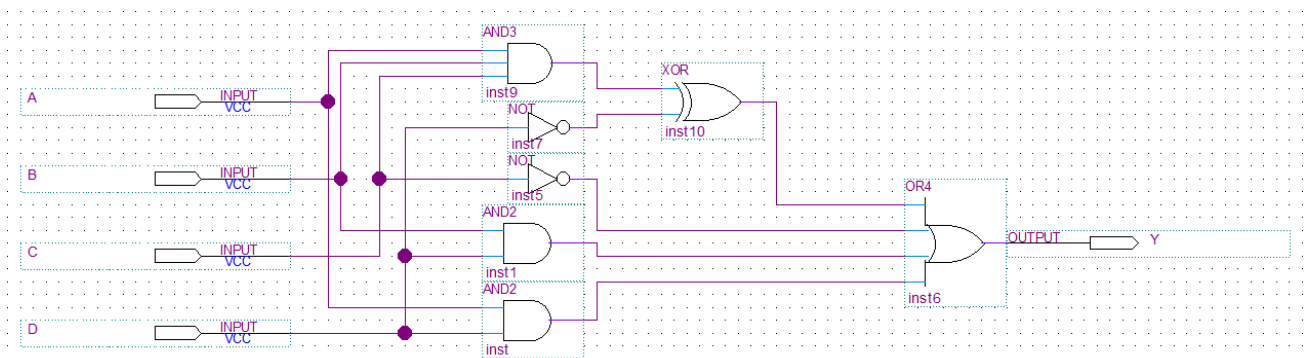


Рис. 1 Схема

Временная диаграмма:

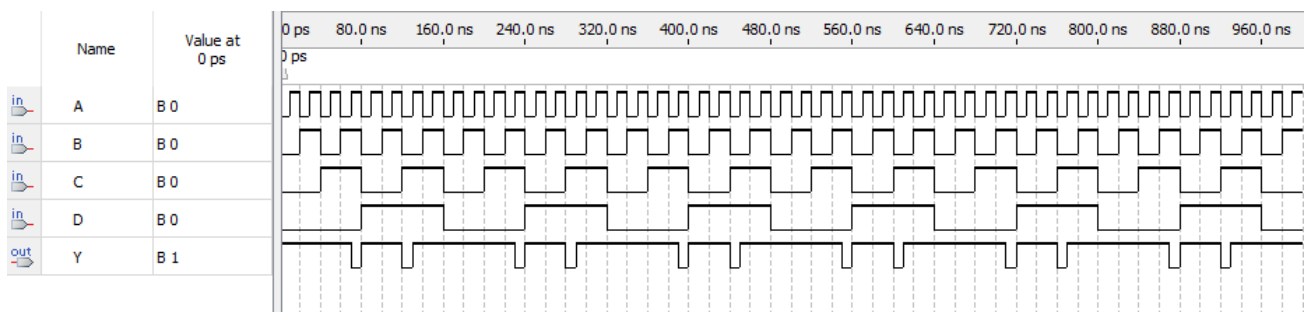


Рис. 2 Временная диаграмма

Вывод: были изучены правила построения и принцип работы логических схем. Была спроектирована логическая схема при помощи графического редактора САПР MAX+PLUS II.

Лабораторная работа №2

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 MAX+PLUS II. На языке AHDL изменить выражение

$$Y = AD + \bar{C} + BD + \bar{D} \oplus ACB$$

Код программы:

```
1  SUBDESIGN 'lab12'
2  (
3      A, B, C, D : INPUT;
4      Y : OUTPUT;
5  )
6  BEGIN
7      Y = A AND D OR NOT C OR B AND D OR NOT D $ A AND B AND C;
8  END;
9
```

Временная диаграмма:

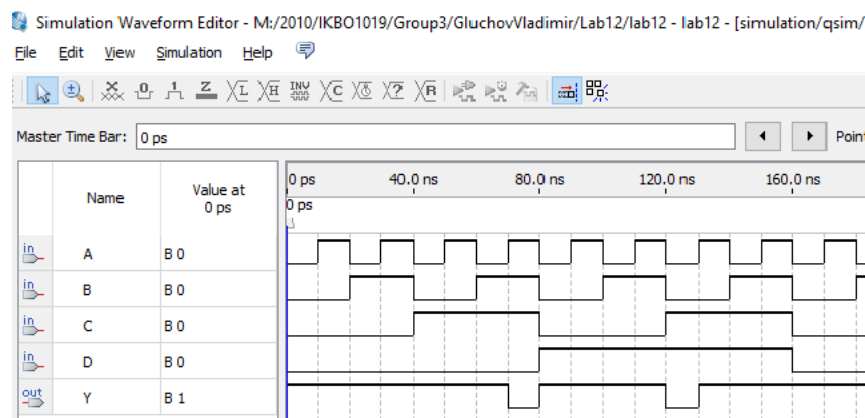


Рис. 3 Временная диаграмма

Вывод: Временная диаграмма в Практическая 1.1 и Практическая 1.2 совпали, значит задания выполнены верно. В ходе работы был написан код логической схемы варианта. Сравнение результатов работы подтвердило правильность построения.

Лабораторная работа №3

Моделирование цифровых схем с использованием параметрических элементов

Цель работы:

Приобретение навыков использования параметрических элементов (LPM function) в САПР MAX+PLUS II, экспериментальное исследование счетчиков и регистров, построенных на их основе.

Описание:

Шифратором (англ. *Coder*) называется комбинационное устройство, преобразующее номер входного сигнала в двоичный код, десятичный эквивалент которого равен номеру возбужденного входа. Шифратор формирует двоичный код номера входной линии, на которую поступает сигнал. В результате шифратор производит преобразование десятичных чисел в двоичную систему счисления.

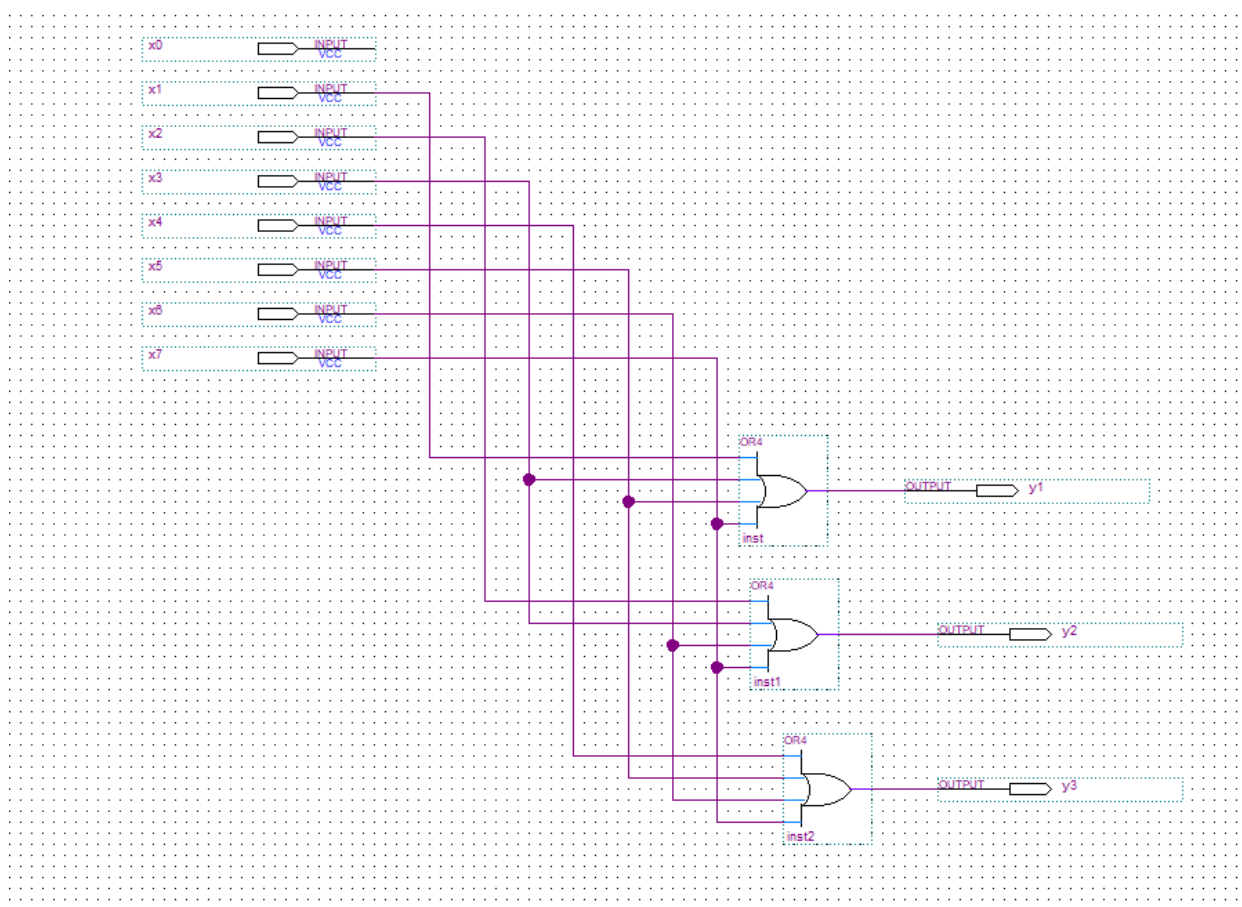


Рис. 4 Схема дешифратора

Временная диаграмма:

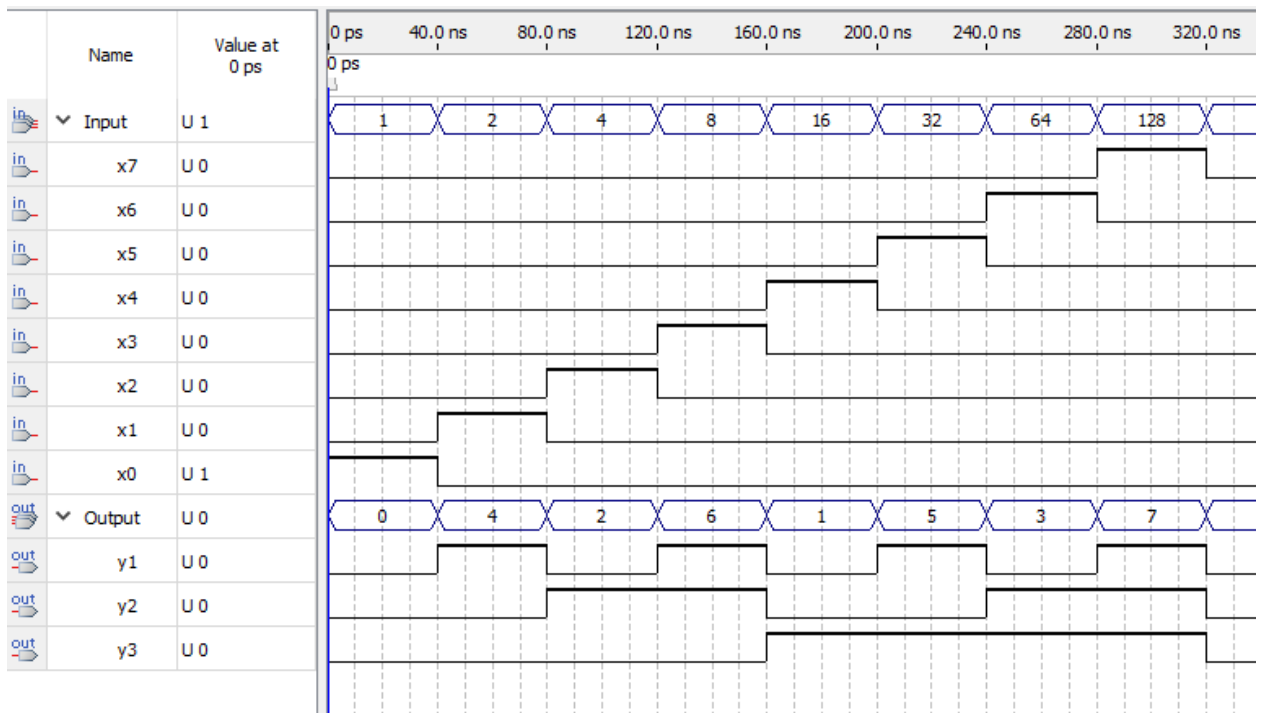


Рис. 5 Временная диаграмма

Вывод: в ходе работы были приобретены навыки использования параметрических элементов (LPM function) в САПР MAX+PLUS II.

Лабораторная работа №4

Моделирование цифровых схем с использованием языка AHDL

Цель работы:

Ознакомиться с САПР Max+Plus II фирмы Altera, получить практические навыки создания проектов по схемотехнике ЭВМ в САПР (ввод схем, компиляция и моделирование).

Описание:

Шифратором (англ. *Coder*) называется комбинационное устройство, преобразующее номер входного сигнала в двоичный код, десятичный эквивалент которого равен номеру возбужденного входа. Шифратор формирует двоичный код номера входной линии, на которую поступает сигнал. В результате шифратор производит преобразование десятичных чисел в двоичную систему счисления.

Код программы:

```
1  SUBDESIGN 'Lap_3'
2  (
3      x0,x1,x2,x3,x4,x5,x6,x7: INPUT;
4      y1,y2,y3: OUTPUT;
5  )
6  BEGIN
7      y1 = x1 # x3 # x5 # x7;
8      y2 = x2 # x3 # x6 # x7;
9      y3 = x4 # x5 # x6 # x7;
10 END;
```

Временная диаграмма:

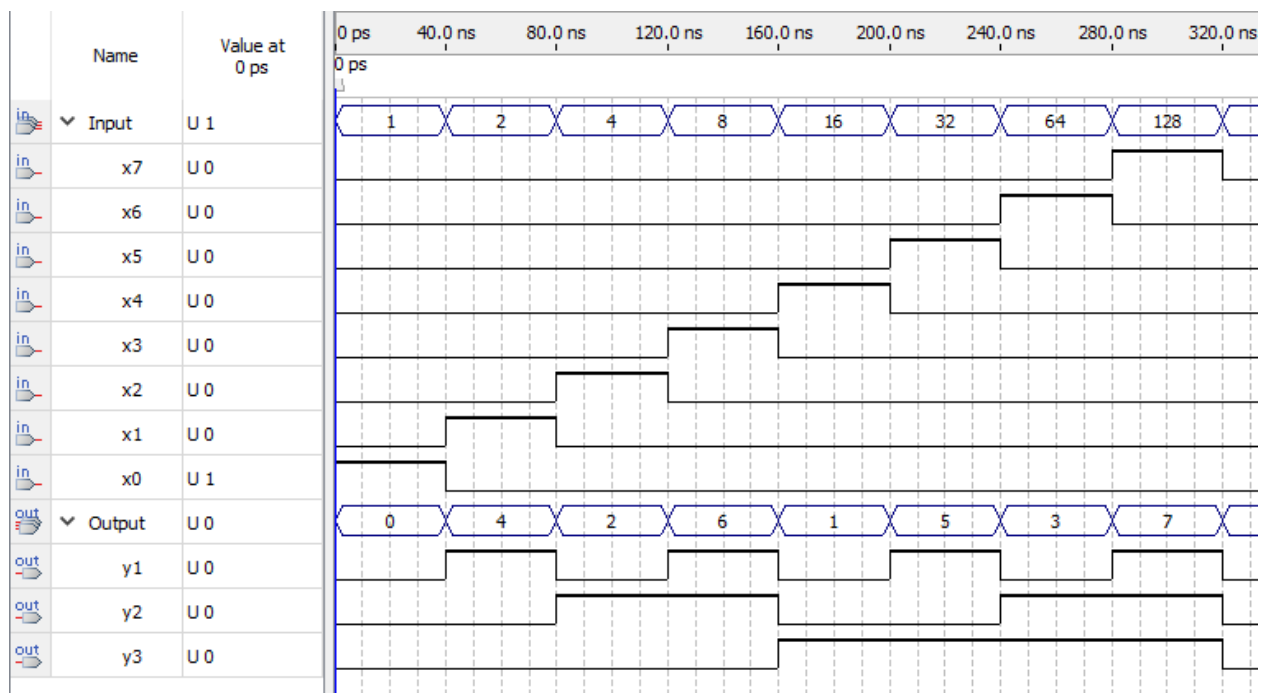


Рис. 6 Временная диаграмма

Вывод: Временная диаграмма в Практическая 3 и Практическая 4 совпали, значит задания выполнены верно. В ходе работы был написан код схемы компаратора. Сравнение с результатами предыдущей работы подтвердили правильность выполнения.

Лабораторная работа № 5

1. Постановка задачи

Цель работы:

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

Вариант: Прямой счётчик по модулю 25

2. Схема

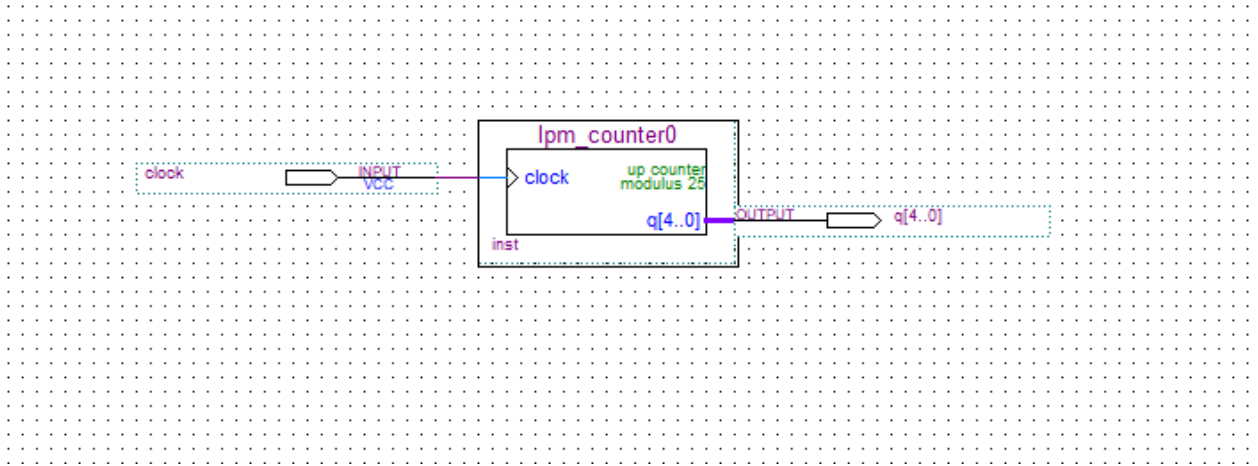


Рисунок 1-Синтезированная схема в графическом редакторе САПР QUARTUS II

3. Временная диаграмма

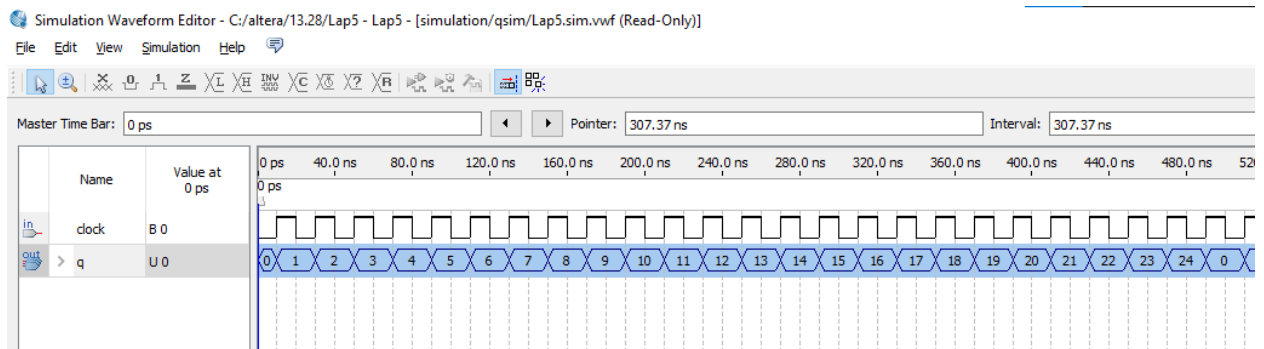


Рисунок 2-Результаты моделирования работы программы в сигнальном редакторе

ВЫВОДЫ

В ходе работы получены знания и практические навыки использования параметрических элементов (LPM function) в САПР QUARTUS II, экспериментальное исследование счетчиков и регистров, построенных на их основе по построению схемы и диаграммы по выражению.

Лабораторная работа №6

1. Постановка задачи

Цель: Ознакомиться с САПР QUARTUS II фирмы Altera, получить практические навыки создания проектов по схемотехнике ЭВМ в САПР (вариант 14)

2. Таблица перекодировки состояний автомата

Таблица 1: Таблица перекодировки состояний автомата

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

3. Граф состояний

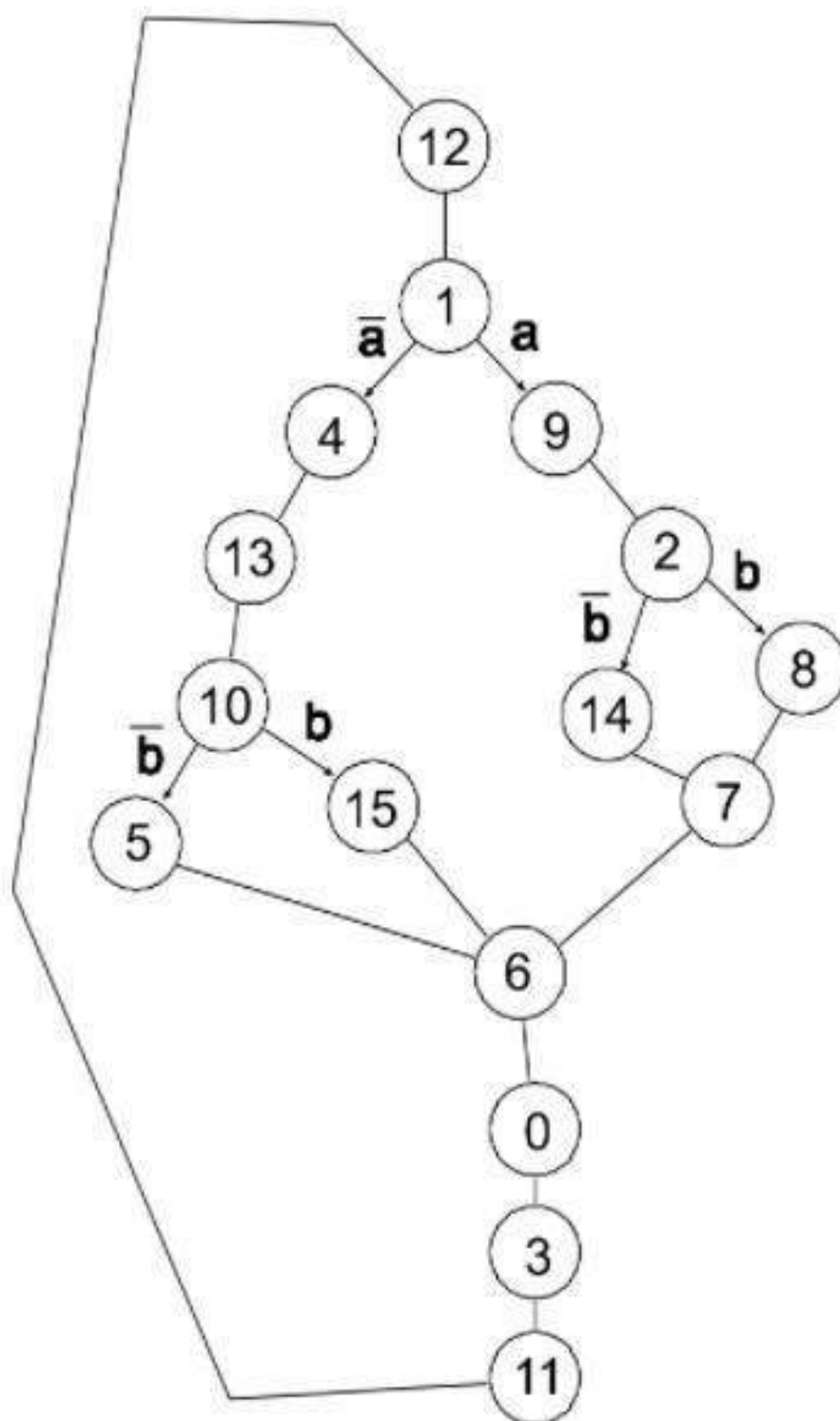


Рис.1 Граф состояний

4. Таблица истинности автомата

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

Таблица 2: Таблица истинности автомата

5. Схеми

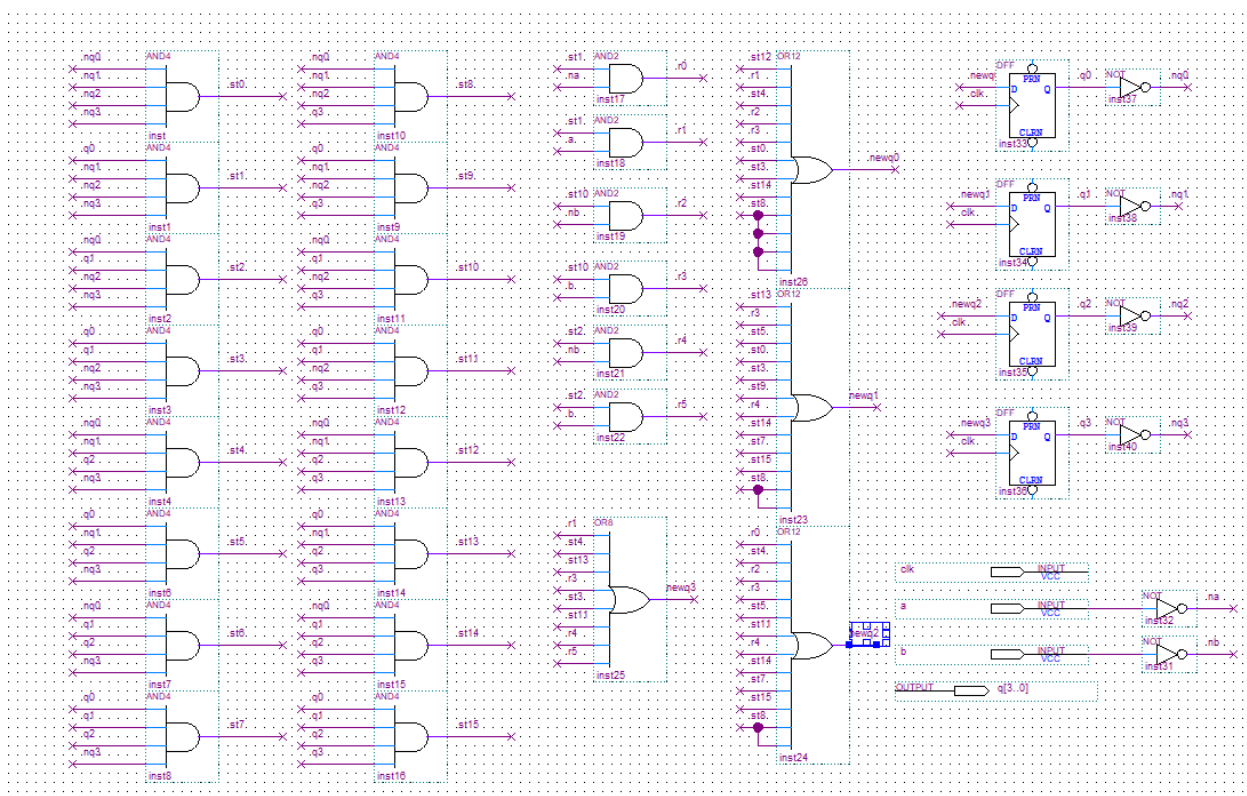


Рисунок 2. Синтезированная схема в графическом редакторе САПР QUARTUS II.

6. Временная диаграмма

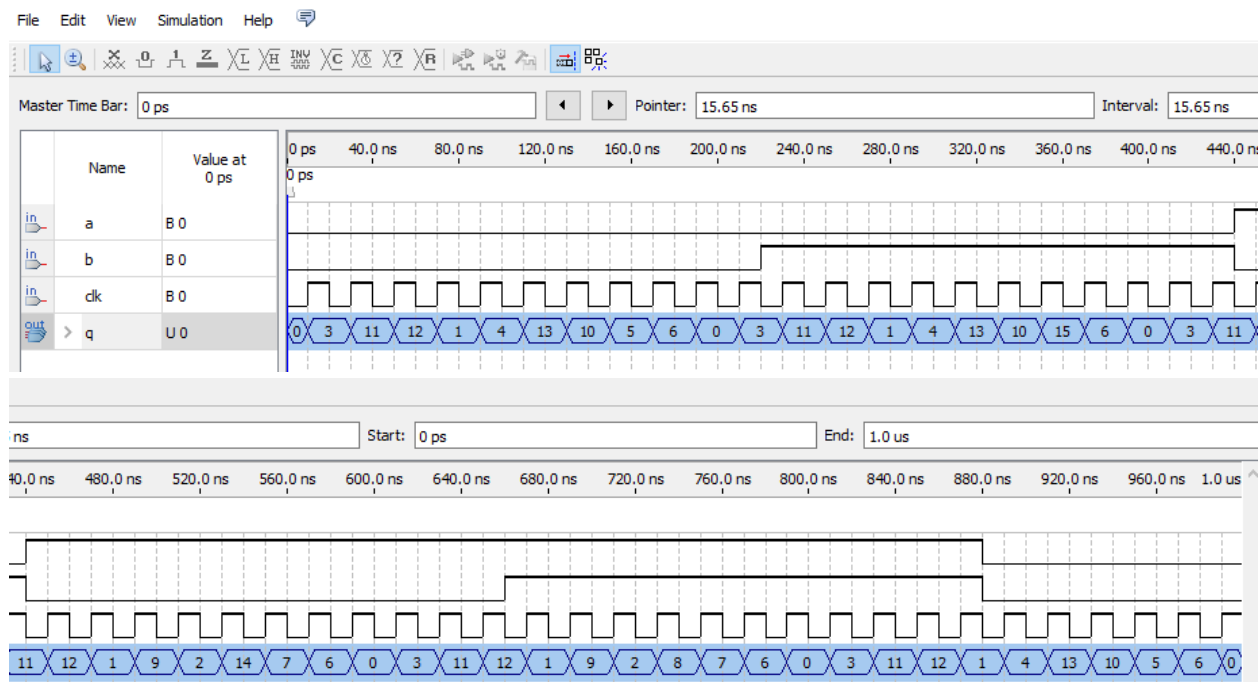


Рис.3 Временные диаграммы

7. Заключение

В ходе работы получены знания и практические навыки по построению схемы счетчика.

Лабораторная работа №7

1. Постановка задачи

Цель: Ознакомиться с САПР QUARTUS II фирмы Altera, получить практические навыки создания проектов по схемотехнике ЭВМ в САПР (вариант 14)

2. Таблица перекодировки состояний автомата

Таблица 1: Таблица перекодировки состояний автомата

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

3. Граф состояний

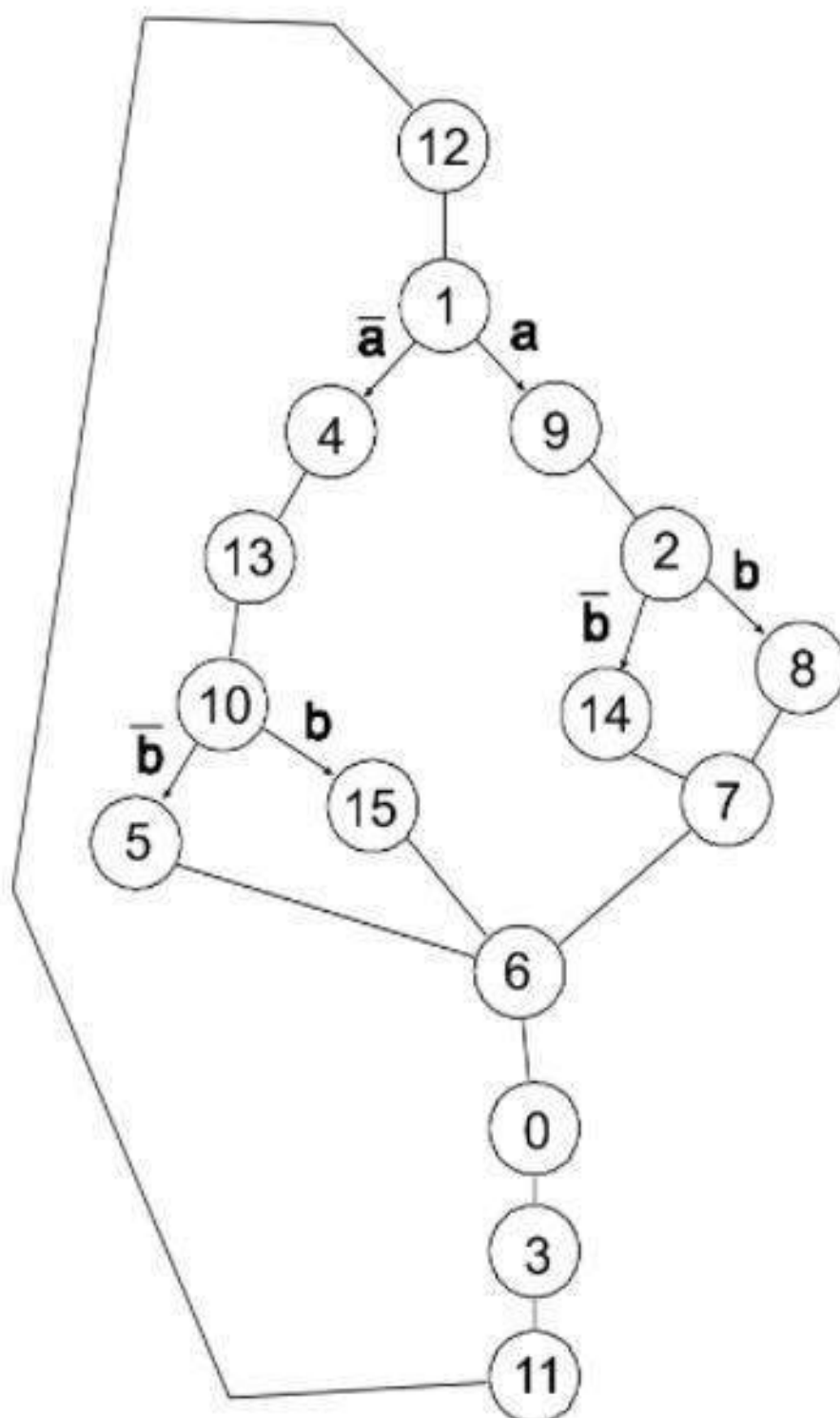


Рис.1 Граф состояний

4. Таблица истинности автомата

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

Таблица 2: Таблица истинности автомата

5. Код

```
SUBDESIGN 'Lap7'
(
  clock, a, b :INPUT;
  q[3..0] :OUTPUT;
)
VARIABLE
newq[3..0], st[15..0], r[5..0] :NODE;
BEGIN
  st0 = !q0 and !q1 and !q2 and !q3;
  st1 = q0 and !q1 and !q2 and !q3;
  st2 = !q0 and q1 and !q2 and !q3;
  st3 = q0 and q1 and !q2 and !q3;
  st4 = !q0 and !q1 and q2 and !q3;
  st5 = q0 and !q1 and q2 and !q3;
  st6 = !q0 and q1 and q2 and !q3;
  st7 = q0 and q1 and q2 and !q3;
  st8 = !q0 and !q1 and !q2 and q3;
  st9 = q0 and !q1 and !q2 and q3;
  st10 = !q0 and q1 and !q2 and q3;
  st11 = q0 and q1 and !q2 and q3;
  st12 = !q0 and !q1 and q2 and q3;
  st13 = q0 and !q1 and q2 and q3;
  st14 = !q0 and q1 and q2 and q3;
  st15 = q0 and q1 and q2 and q3;
  r0 = st1 and !a;
  r1 = st1 and a;
  r2 = st10 and !b;
  r3 = st10 and b;
  r4 = st2 and !b;
  r5 = st2 and b;
  newq0 = st12 or r1 or st4 or r2 or r3 or st0 or st3 or st14 or st8 ;
  newq1 = st13 or r3 or st5 or st0 or st3 or st9 or r4 or st14 or st7 or st15 or st8;
  newq2 = r0 or st4 or r2 or r3 or st5 or st11 or r4 or st14 or st7 or st15 or st8;
  newq3 = r1 or st4 or st13 or r3 or st3 or st11 or r4 or r5;
  q0 = dff(newq0, clock, , );
  q1 = dff(newq1, clock, , );
  q2 = dff(newq2, clock, , );
  q3 = dff(newq3, clock, , );
END;
```

SUBDESIGN 'Lap7'

(

clock, a, b :INPUT;

q[3..0] :OUTPUT;

)

VARIABLE

newq[3..0], st[15..0], r[5..0] :NODE;

BEGIN

st0 = !q0 and !q1 and !q2 and !q3;

st1 = q0 and !q1 and !q2 and !q3;

st2 = !q0 and q1 and !q2 and !q3;

st3 = q0 and q1 and !q2 and !q3;

```

st4 = !q0 and !q1 and q2 and !q3;
st5 = q0 and !q1 and q2 and !q3;
st6 = !q0 and q1 and q2 and !q3;
st7 = q0 and q1 and q2 and !q3;
st8 = !q0 and !q1 and !q2 and q3;
st9 = q0 and !q1 and !q2 and q3;
st10 = !q0 and q1 and !q2 and q3;
st11 = q0 and q1 and !q2 and q3;
st12 = !q0 and !q1 and q2 and q3;
st13 = q0 and !q1 and q2 and q3;
st14 = !q0 and q1 and q2 and q3;
st15 = q0 and q1 and q2 and q3;
r0 = st1 and !a;
r1 = st1 and a;
r2 = st10 and !b;
r3 = st10 and b;
r4 = st2 and !b;
r5 = st2 and b;
newq0 = st12 or r1 or st4 or r2 or r3 or st0 or st3 or st14 or st8 ;
newq1 = st13 or r3 or st5 or st0 or st3 or st9 or r4 or st14 or st7 or st15 or st8;
newq2 = r0 or st4 or r2 or r3 or st5 or st11 or r4 or st14 or st7 or st15 or st8;
newq3 = r1 or st4 or st13 or r3 or st3 or st11 or r4 or r5;
q0 = dff(newq0, clock, , );
q1 = dff(newq1, clock, , );
q2 = dff(newq2, clock, , );
q3 = dff(newq3, clock, , );
END;

```

6. Временная диаграмма

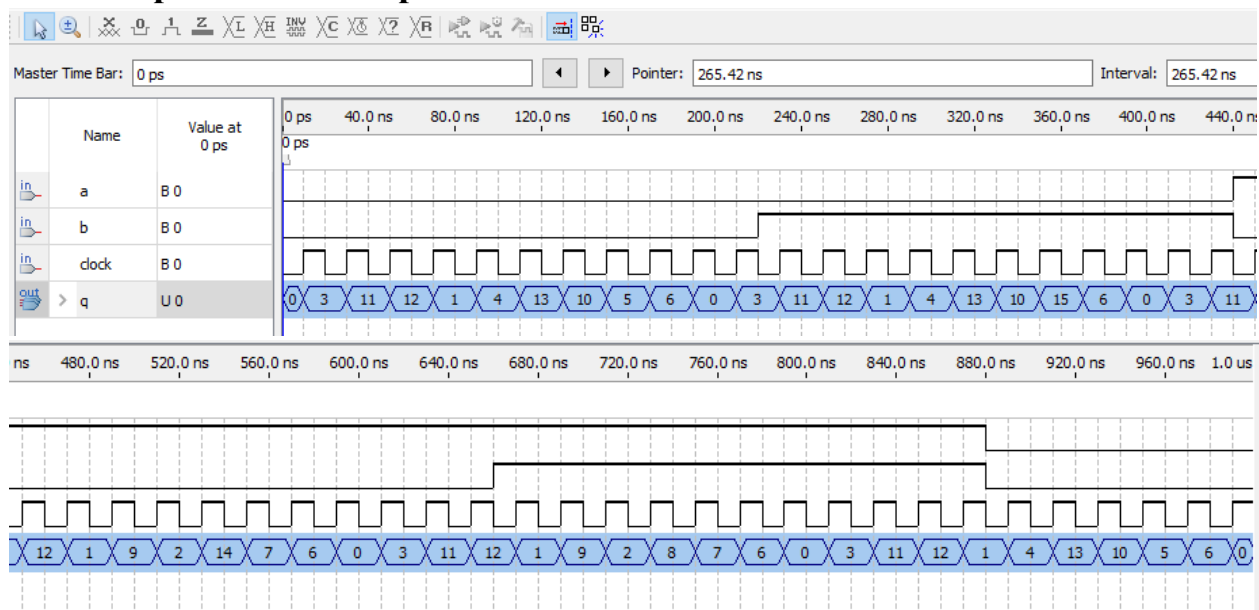


Рис.2 Временная диаграмма

7. Заключение

В ходе работы получены знания и практические навыки по построению схемы счетчика.

Лабораторная работа №8

1. Цель лабораторной работы

Целью данной лабораторной 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 лабораторной работы студенты познакомятся с процессом установки на персональный компьютер виртуальной машины OracleVirtualBox, получают представление о процессе создания и настройки виртуального окружения. На примере операционной системы UbuntuLinux будет выполнен процесс установки и базовой настройки операционной системы.

Задание

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

Для выполнения лабораторной 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, видео-памяти, 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.

2. Отчет по заданию

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.

2. Установить Oracle VirtualBox на компьютер

Для установки OracleVirtualBox на компьютере необходимо запустить скачанный на первом шаге установочный файл дистрибутива.

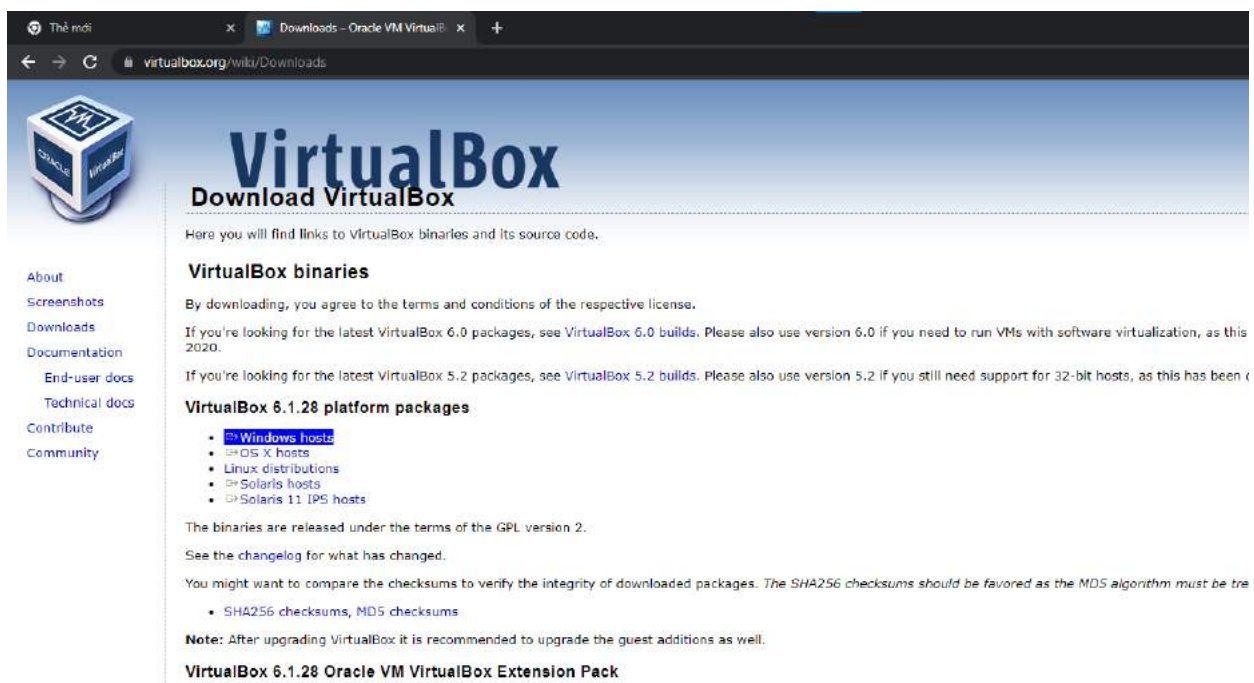


Рисунок 1 – Выбор дистрибутива Oracle VirtualBox



Рисунок 2 – Окно программы установки Oracle VirtualBox

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

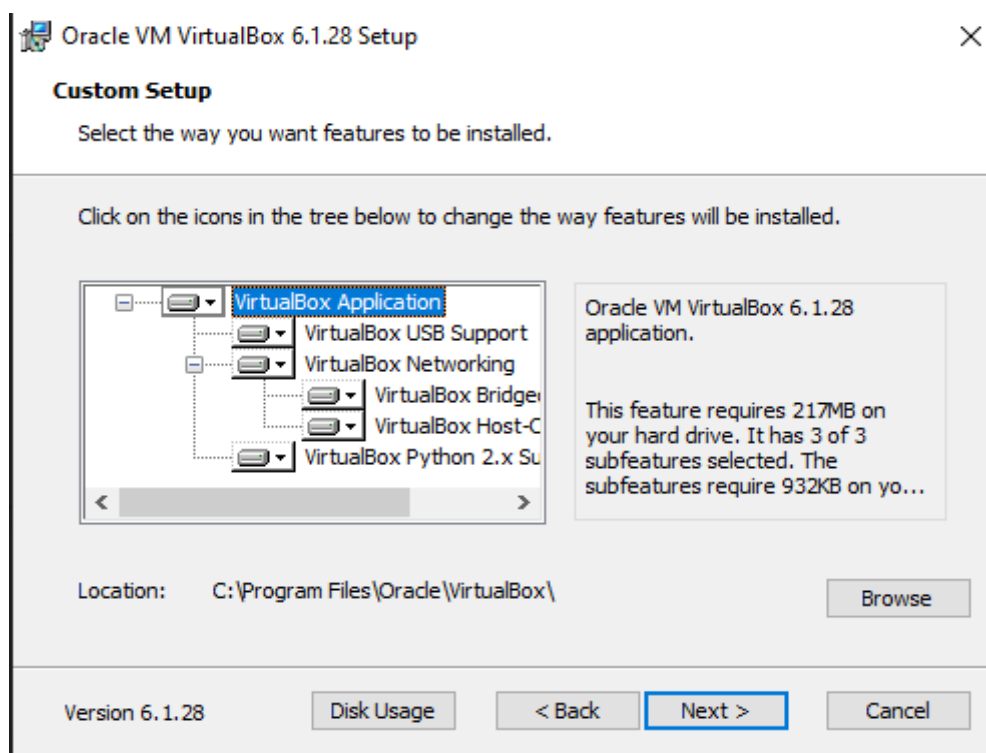


Рисунок 3 – Диалог выбора компонентов установк

По завершении установки на рабочем столе компьютера будет создан ярлык для запуска виртуальной машины OracleVirtualBox



Рисунок 4 – Окно завершения 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.ru/> На открывшейся странице переходим по ссылке для скачки.



Рисунок 5 – Выбор дистрибутива Ubuntu

Сохраняем образ установочного диска на жесткий диск.

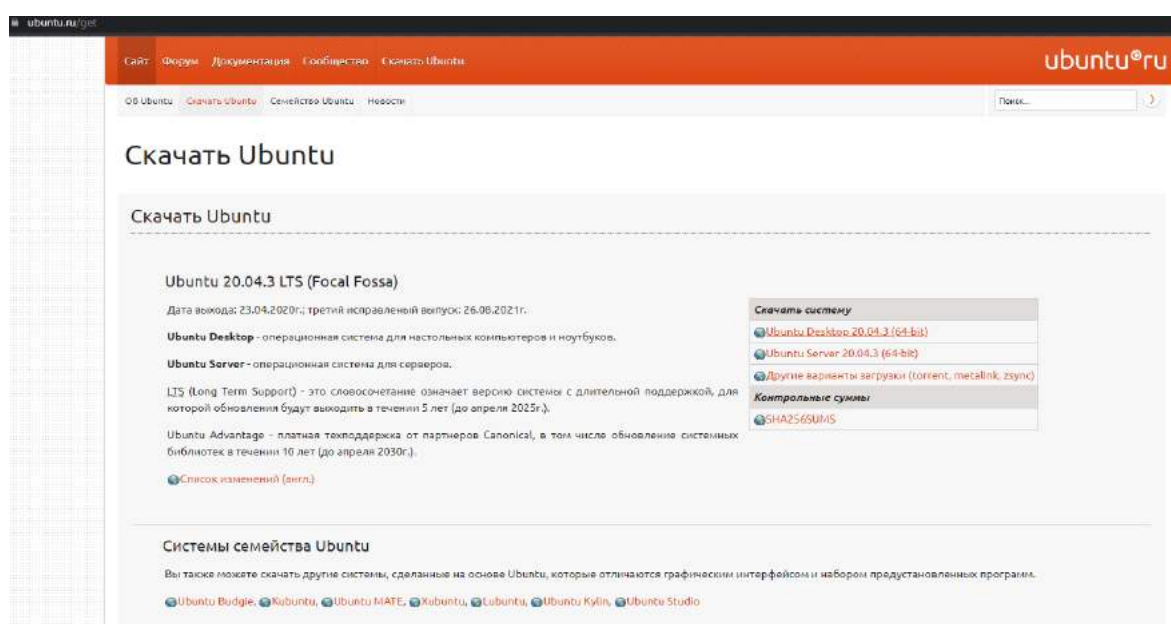


Рисунок 6 – Выбор дистрибутива Ubuntu

1. Создать и настроить виртуальную машину

Запускаем установленное приложение OracleVirtualBox. В появившемся окне в левой части перечислены все созданные виртуальные машины. В правой части отображаются свойства выбранной в списке виртуальной машины. Сразу после установки список виртуальных машин пуст. Для создания новой виртуальной машины необходимо в верхней части окна нажать на кнопку

«Создать».

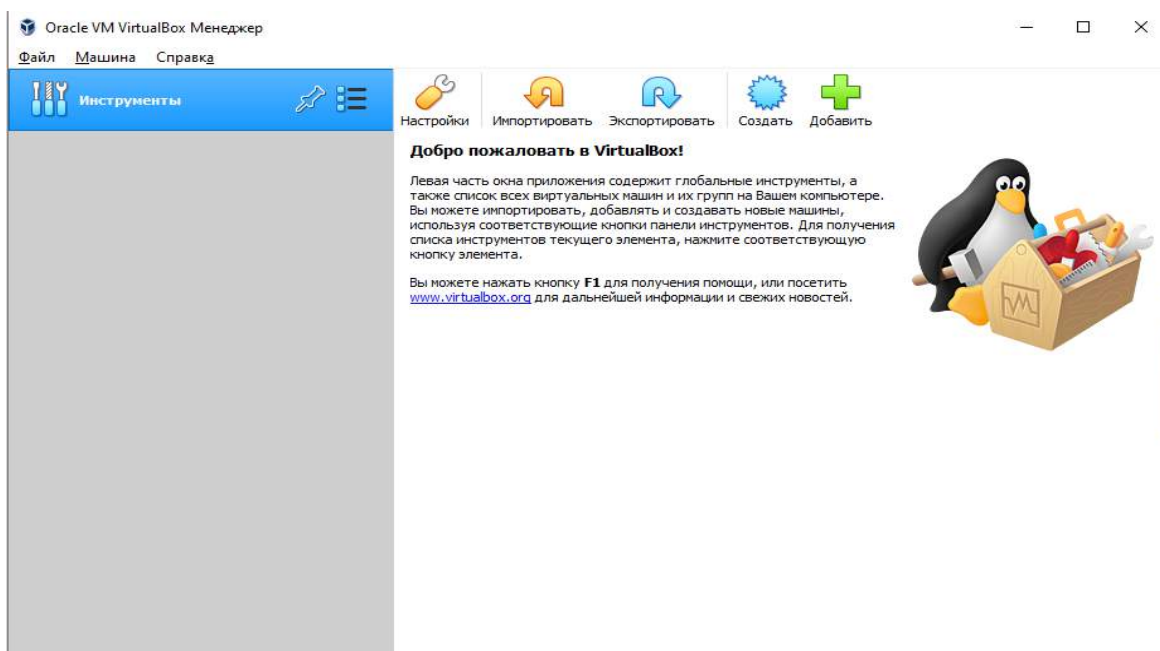


Рисунок 7 – Окно программы Oracle VirtualBox

При создании для новой виртуальной машины необходимо задать оригинальное имя, сопоставимое с ФИО студента, тип и версию операционной системы, которая планируется для установки.

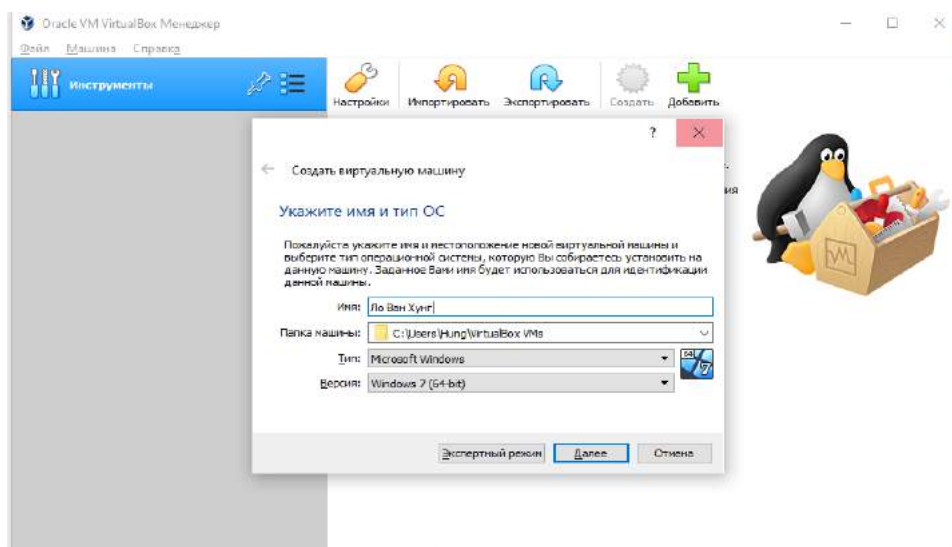


Рисунок 8 – Задание имени и типа новой виртуальной машины

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. Внимание! Не выделяйте более 50% объема физической памяти компьютера для виртуальной машины, так как это может привести к нестабильной работе компьютера.

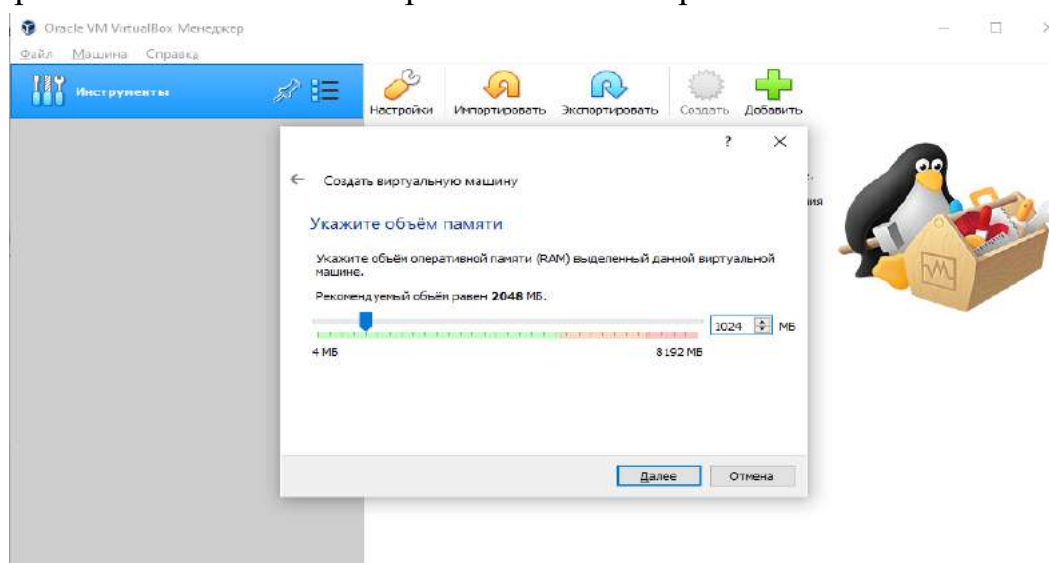


Рисунок 9 – Выделение оперативной памяти для виртуальной машины

После выделения оперативной памяти для виртуальной машины необходимо задать размер жесткого диска. Жесткий диск виртуальной машины представляет собой один или несколько файлов особого формата, содержащие всю информацию виртуальной машины.

Все изменения, вносимые на жесткий диск виртуальной машины, не могут причинить вред информации, хранящейся на жестком диске основного компьютера.

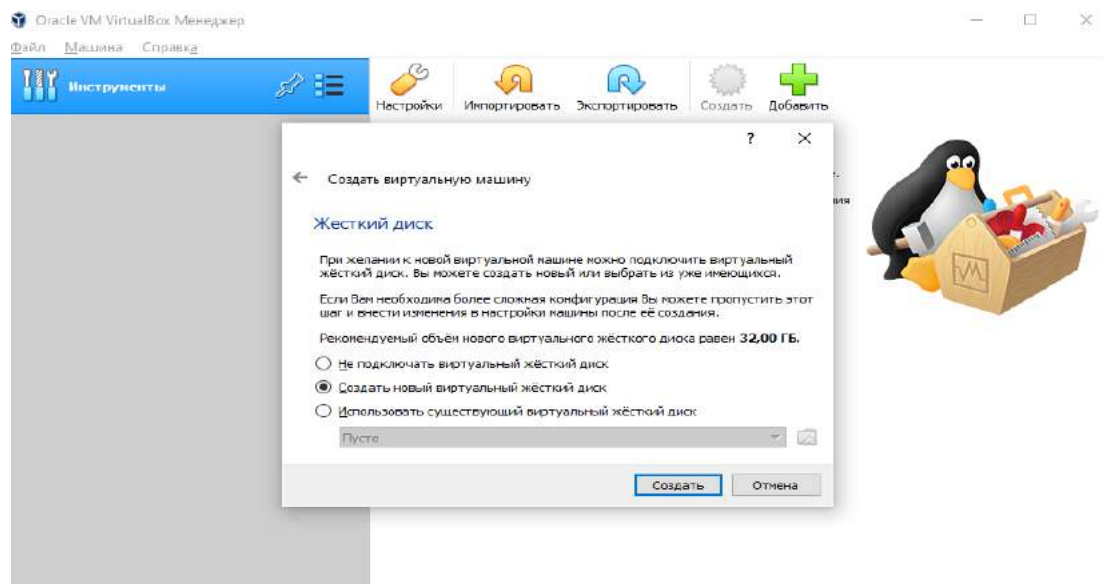


Рисунок 10 – Создание виртуального жёсткого диска

В OracleVirtualBox поддерживается несколько форматов файлов жестких дисков.

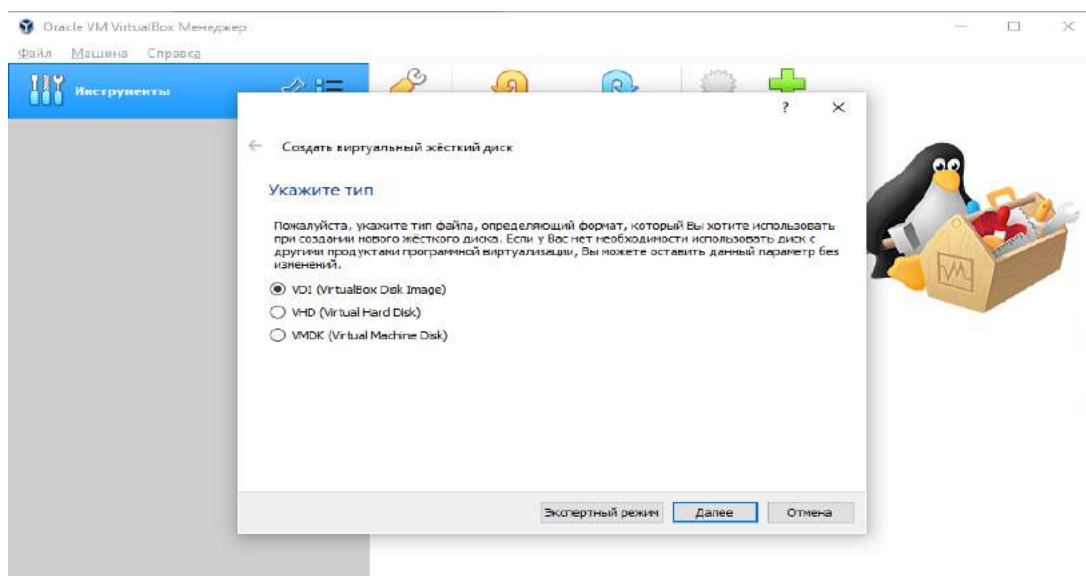


Рисунок 11 – Выбор типа виртуального жёсткого диска

Существует два поддерживаемых формата хранения файлов жесткого диска виртуальной машины: динамический и фиксированный. Для экономии места на жестком диске можно выбрать динамический формат хранения данных жесткого диска виртуальной машины. Однако, из практики лучше выбирать фиксированный формат виртуального диска.

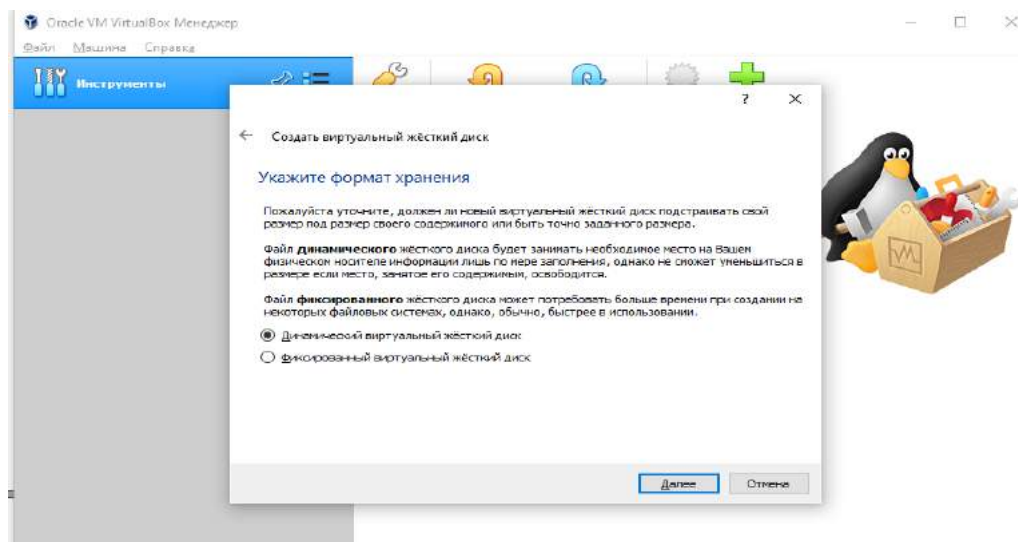


Рисунок 12 – Выбор формата хранения виртуального жёсткого диска

Далее необходимо задать имя файла жесткого диска и максимальный размер файловых данных, который виртуальная машина сможет хранить. Внимание! При выборе фиксированного формата хранения жесткого диска заданный объем сразу будет занят файлом. Таким образом, при задании размера необходимо оценивать требуемый для работы виртуальной машины объем дискового пространства и наличие свободного места на жестком диске основного компьютера.

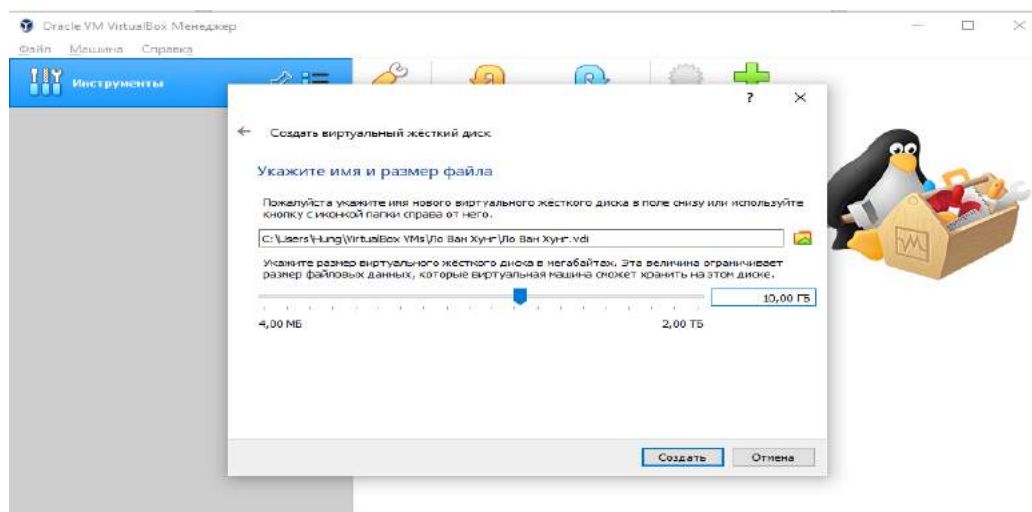


Рисунок 13 –Задание размера виртуального жёсткого диска

Подготовить виртуальную машину к установке операционной системы Для установки операционной системы необходимо в виртуальный привод подключить скаченный на 3-м шаге образ установочного диска. Для этого необходимо в списке виртуальных машин выделить созданную нами виртуальную машину и в верхней части окна нажать на кнопку «Настроить».

В появившемся окне необходимо перейти во вкладку «Носители», выбрать виртуальный привод и подключить скаченный образ установочного диска.

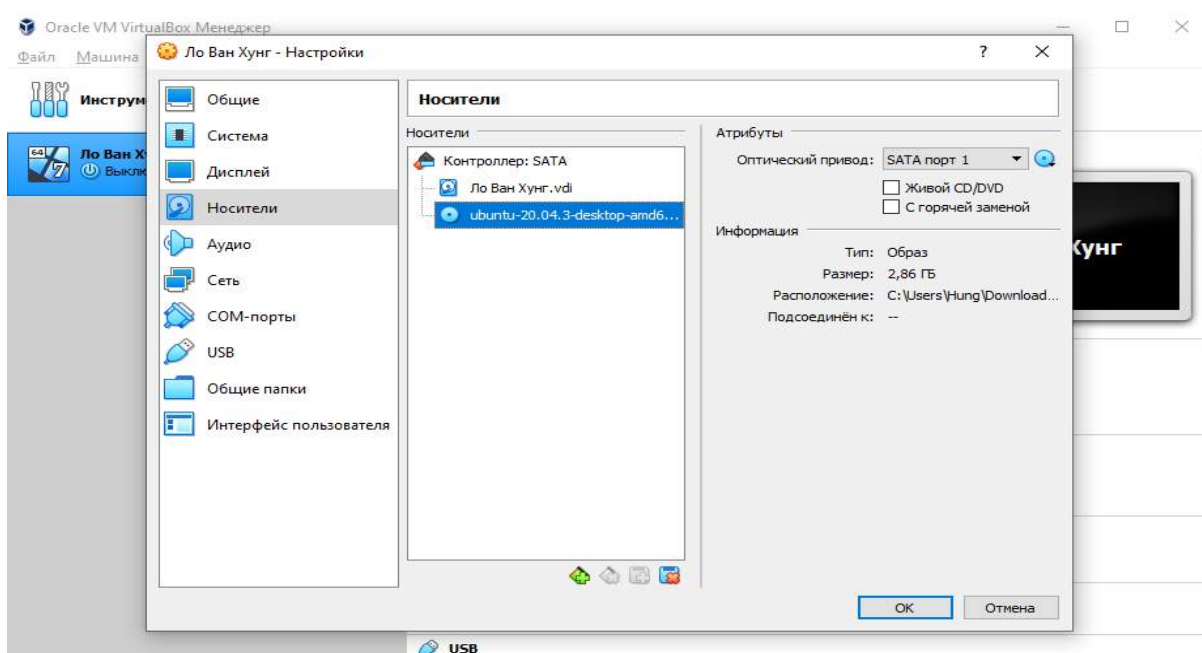


Рисунок 14 – Выбор виртуального привода

1. Установить операционную систему Ubuntu Linux на

виртуальную машину

Для начала установки операционной системы необходимо запустить созданную виртуальную машину. Для этого надо выбрать ее в списке виртуальных машин и нажать на кнопку «Запустить», расположенную в верхней части окна.

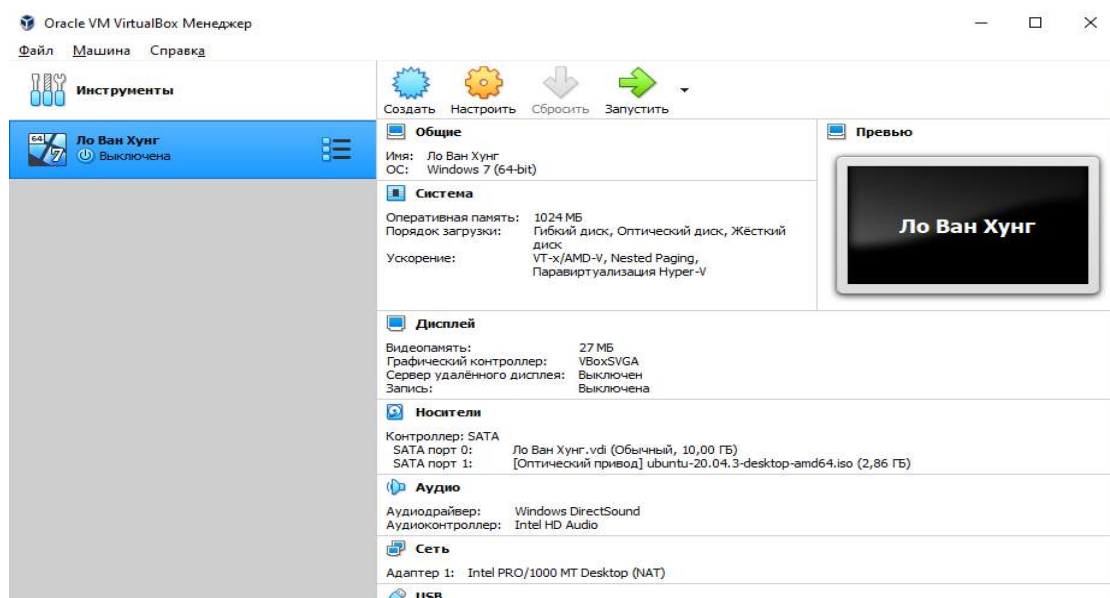


Рисунок 15 – Запуск виртуальной машины

Виртуальная машина начнет загружаться с установочного образа диска, смонтированного в привод. Автоматически будет запущен процесс установки операционной системы.

Первым шагом необходимо выбрать язык, который будет использоваться в процессе установки.

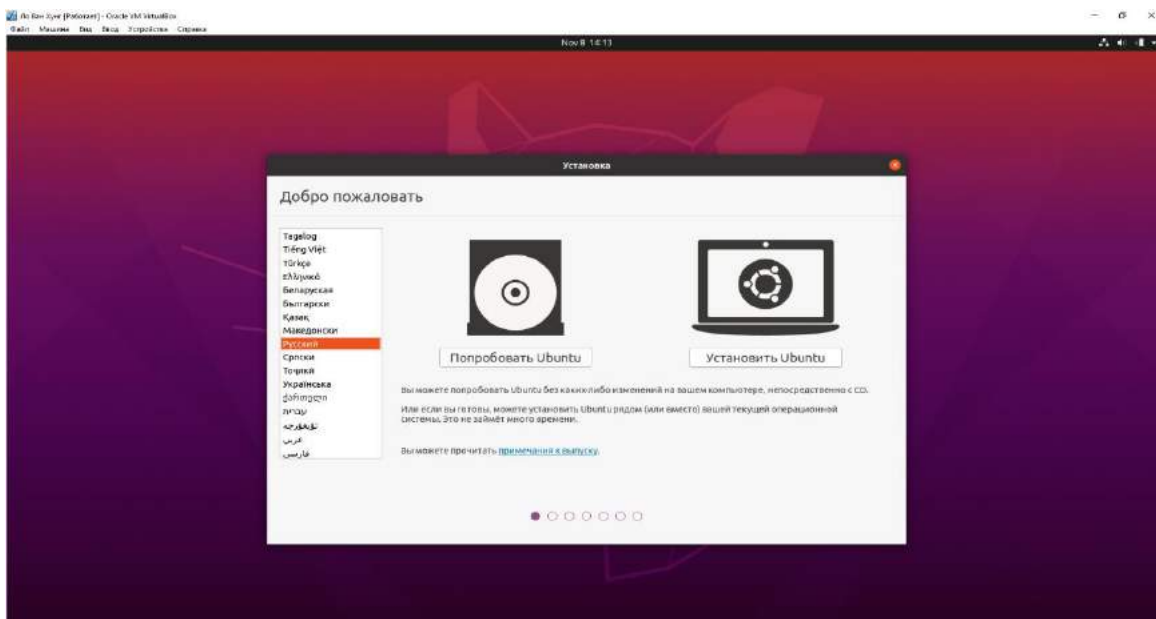


Рисунок 16 – Выбор языка программы установки

После выбора языка необходимо выбрать один из предложенных сценариев: использовать операционную систему без установки или установить операционную систему на жесткий диск виртуальной машины. Для выполнения задания лабораторной работы необходимо выбрать пункт «Установить Ubuntu».

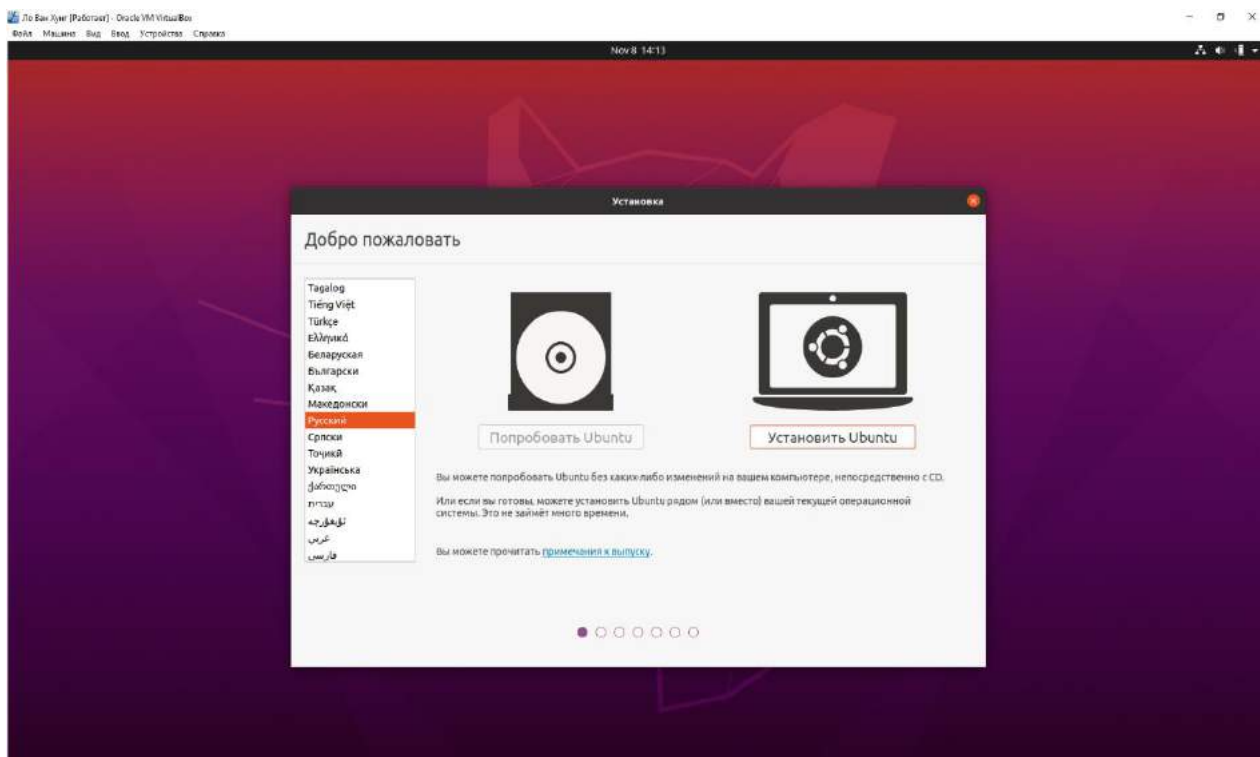


Рисунок 17 – Выбор режима запуска виртуальной машины

При установке можно затребовать установку последних обновлений компонентов операционной системы, а также установку компонент, использование которых может быть ограничено условиями лицензионных соглашений, не подпадающих под условия лицензии GPL, под которой распространяется операционная система UbuntuLinux.

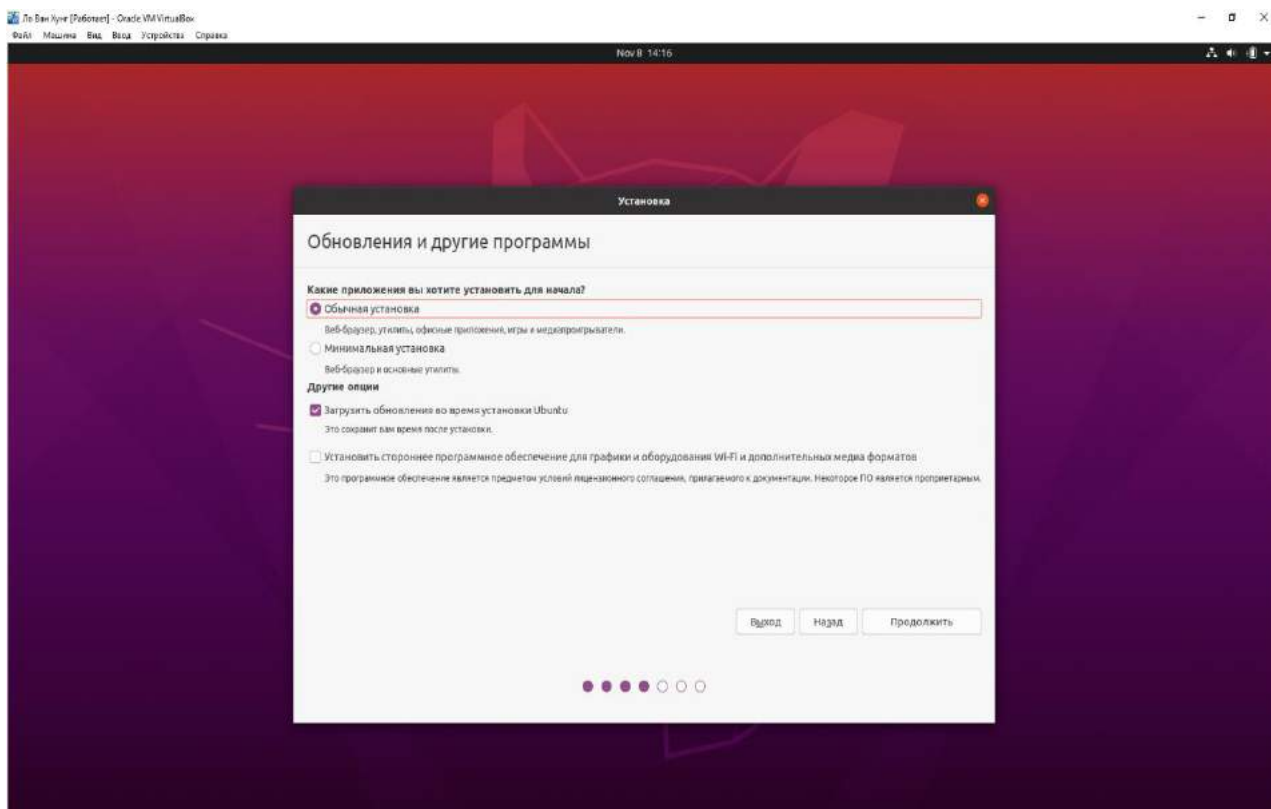


Рисунок 18 – Настройка параметров обновлен

Выбираем тип установки.

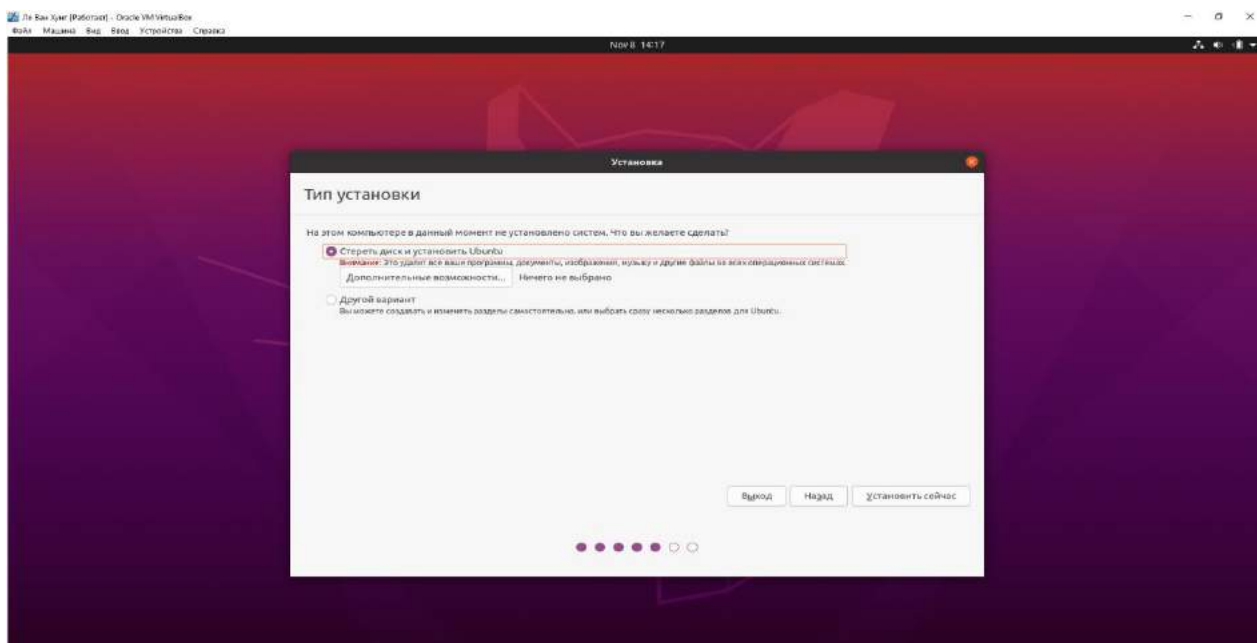


Рисунок 19 – Выбираем тип установки.

Далее необходимо задать часовой пояс. В зависимости от выбранного часового пояса будет рассчитываться смещение времени относительно нулевого меридиана.

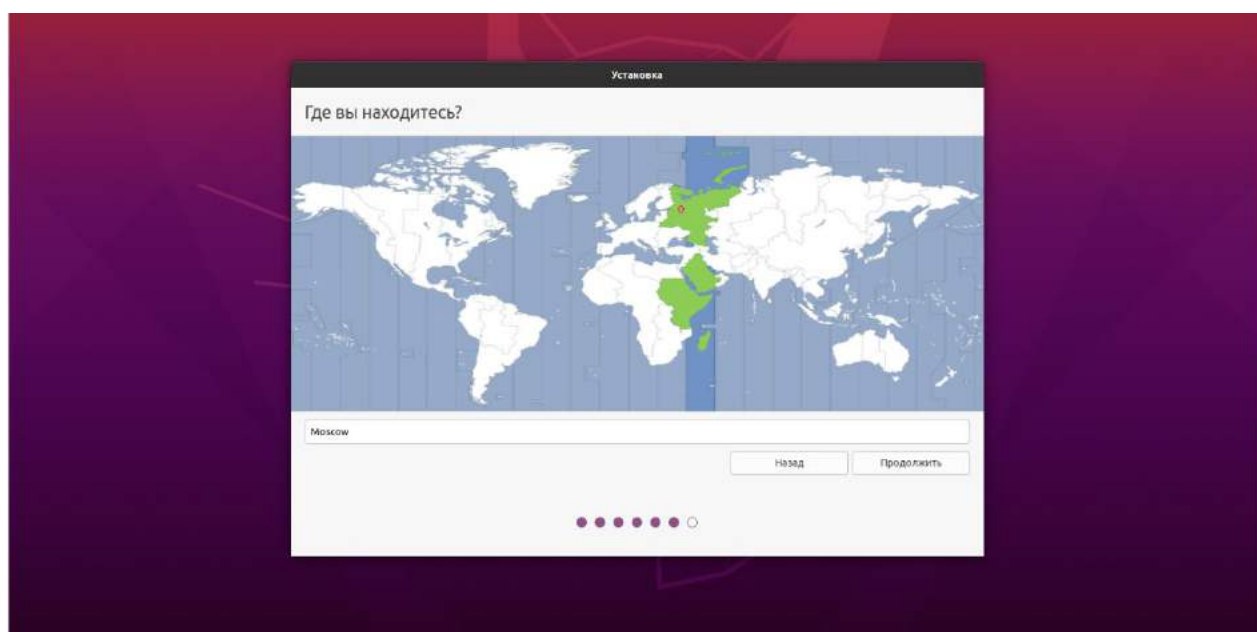


Рисунок 20 – Выбор часового п

Для комфортной работы с операционной системой необходимо выбрать раскладку клавиатуры, используемую по умолчанию. Дополнительную раскладку клавиатуры можно будет задать после установки операционной системы.

Для представления в системе необходимо создать учетную запись, от имени которой будет осуществляться работа.

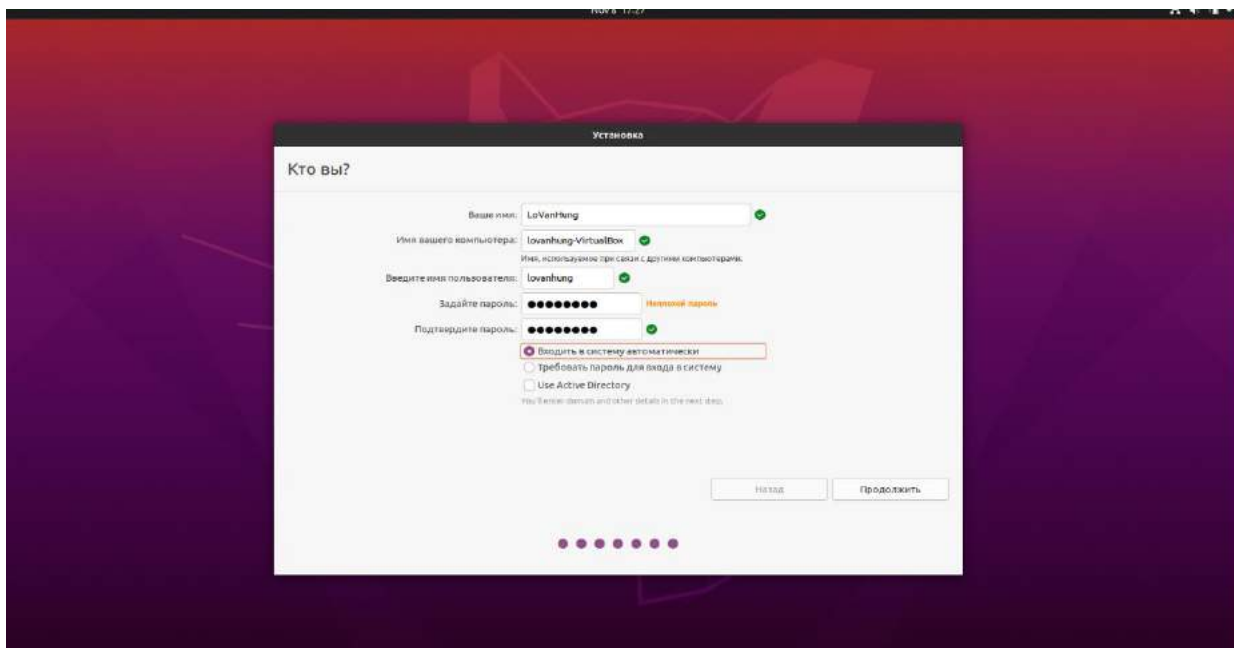


Рисунок 21 – Создание учётной записи пользователя системы

После задания всех параметров начнется установка операционной системы на жесткий диск виртуальной машины. По завершении установки будет предложено перезагрузить виртуальную машину. Процесс перезагрузки может продолжаться длительное время, в зависимости от производительности компьютера.

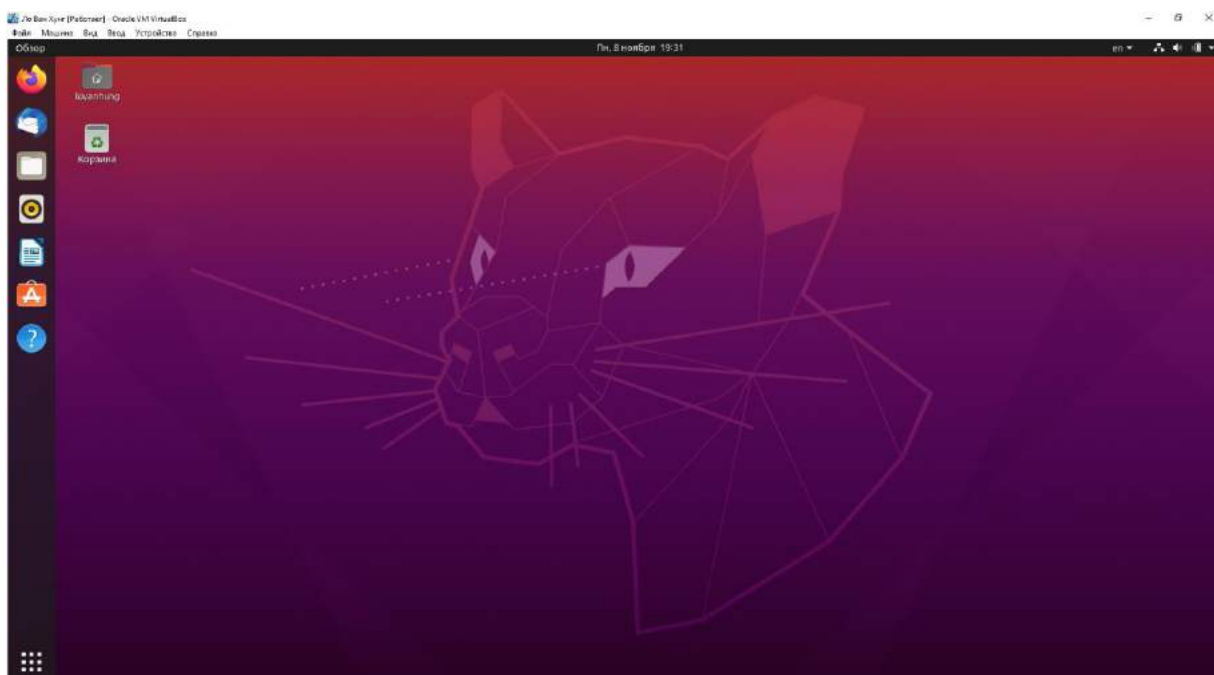


Рисунок 22 – Операционная система после перезагрузки

ВЫВОДЫ

В ходе выполнения данной лабораторной работы были получены навыки установки OracleVirtualBox. Также был изучен процесс создания виртуальной машины. Более того была установлена и настроена операционная система UbuntuLinux.

Лабораторная работа №9

Индивидуальный вариант: 14-Футбольные команды (название команды, ФИО тренера, количество забитых мячей, количество набранных очков). Поиск по названию команды. Сортировка по ФИО тренера.

1. Цель лабораторной работы

Целью данной лабораторной 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.

На примере созданной в процессе лабораторной работы базы данных на основе текстовых файлов будут рассмотрены вопросы сортировки и фильтрации информации, вывод требуемых данных на экран и в файл.

2 Выполнение работы

После запуска системы от имени своей учетной записи и переключения на текстовую консоль, создадим родительский каталог, используя команду *mkdir* (см. Рис.1).

```
lovanhung@lovanhung-VirtualBox:~$ mkdir lovanhung  
lovanhung@lovanhung-VirtualBox:~$ cd lovanhung
```

Рисунок 1 – Создание родительского каталога

После этого внутри каталога, созданного выше создадим структуру каталогов, представленную на рисунке 2, для этого также используем команду *mkdir*. Вывести на экран содержимое текущего каталога и убедиться, что все созданные каталоги созданы без ошибок можно с помощью утилиты *tree*, которую необходимо установить (см. Рисунки 3-4).

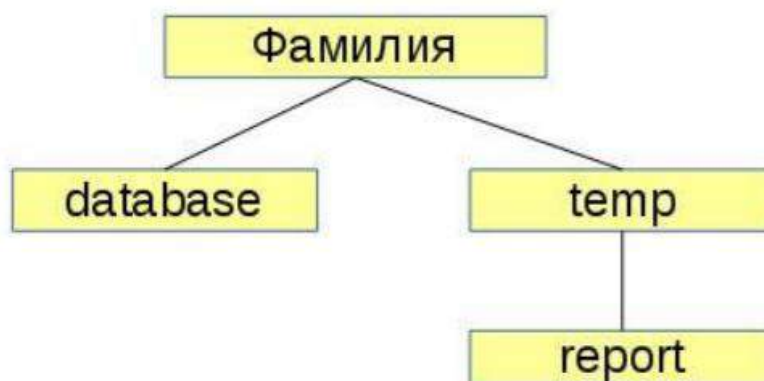


Рисунок 2 – Структура каталогов

```
lovanhung@lovanhung-VirtualBox:~$ cd lovanhung
lovanhung@lovanhung-VirtualBox:~/lovanhung$ mkdir database
lovanhung@lovanhung-VirtualBox:~/lovanhung$ mkdir temp
lovanhung@lovanhung-VirtualBox:~/lovanhung$ cd temp
lovanhung@lovanhung-VirtualBox:~/lovanhung/temp$ mkdir report
```

Рисунок 3 – Создание папок database, temp и report

```
lovanhung@lovanhung-VirtualBox:~/lovanhung$ sudo snap install tree
tree 1.8.0+pkg-3fd6 от 林博仁(Buo-ren, Lin) (brlin) установлен
lovanhung@lovanhung-VirtualBox:~/lovanhung$ tree
locales-launch: Data of ru_RU locale not found, generating, please wait...
├── database
├── temp
│   └── report
3 directories, 0 files
lovanhung@lovanhung-VirtualBox:~/lovanhung$
```

Рисунок 4 – Установка команды tree и проверка правильности создания папок

Перейдем в каталог temp с помощью команды *cd*. Убедимся, что он является текущим с помощью команды *pwd*. Выведем на экран содержимое каталога с помощью команды *ls* (см. Рис.5).

```
lovanhung@lovanhung-VirtualBox:~$ cd lovanhung/temp
lovanhung@lovanhung-VirtualBox:~/lovanhung/temp$ pwd
/home/lovanhung/lovanhung/temp
lovanhung@lovanhung-VirtualBox:~/lovanhung/temp$ ls
report
```

Рисунок 5 – Переход в каталог temp, вывод его содержимого в консоль

Внутри каталога temp создадим файл базы данных dataset1.txt с помощью команды *touch*. Заполним файл данными в соответствии с номером варианта задания с помощью команды *nano*, сохраним введенные записи с помощью *Ctrl+O*, вернемся к консоли с помощью *Ctrl+X* (см. Рис.6). В качестве разделителя столбцов данных в файле используем символ “;” без пробелов. Файл содержит 5 строк (см. Рис.7).

```
lovanhung@lovanhung-VirtualBox:~/lovanhung/temp$ touch database1.txt
lovanhung@lovanhung-VirtualBox:~/lovanhung/temp$ nano database1.txt
```

Рисунок 6 – Создание и открытие в текстовом редакторе файла database1

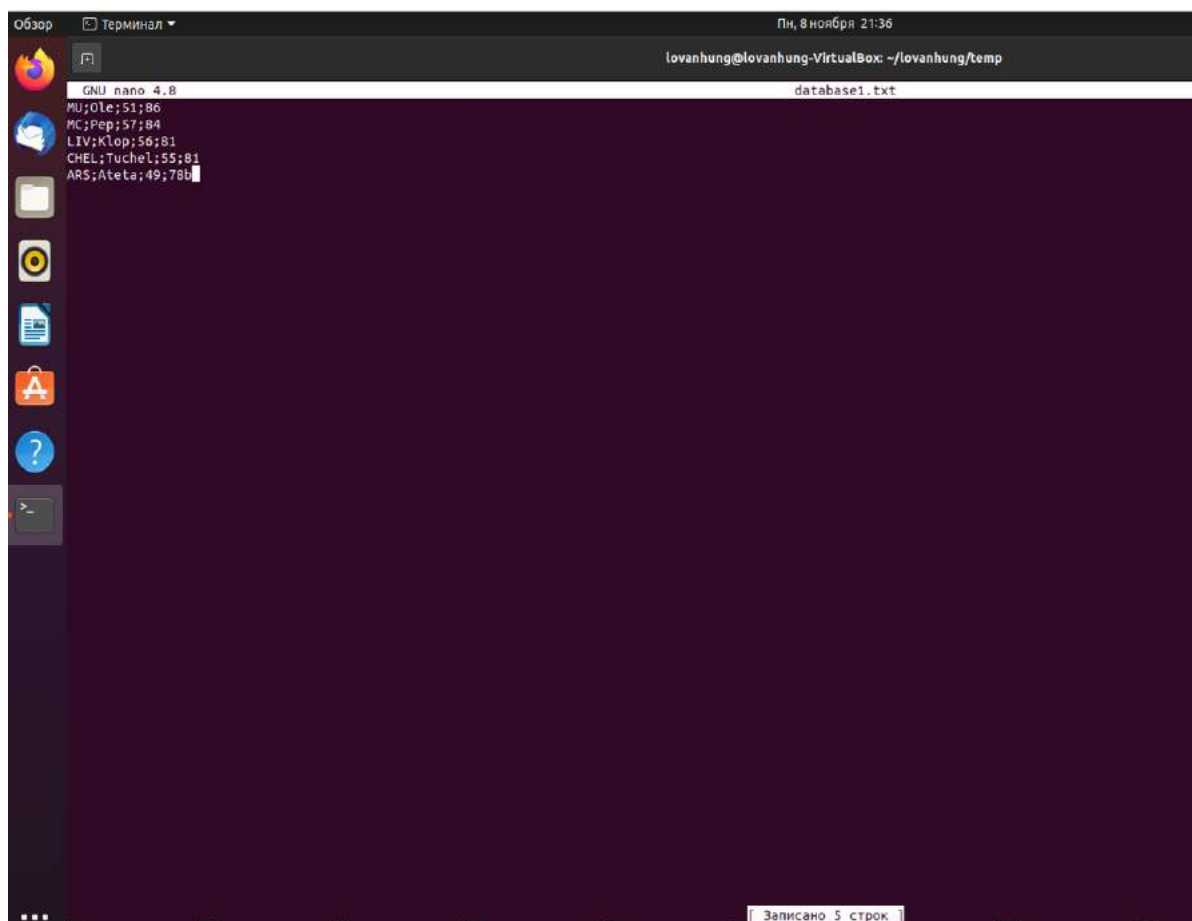


Рисунок 7 – Заполнение файла

С помощью конвейера команд внутри каталога temp создадим файл базы данных dataset2.txt. Заполним файл данными в соответствии с номером варианта задания. В качестве разделителя столбцов данных в файле также используем символ “;” без пробелов. Файл содержит 5 строк. Данные должны отличаться от введенных ранее (см. Рис. 8).



Рисунок 8 – Создание и заполнение файла database2

С помощью перенаправления вывода в файл создадим файл базы данных dataset3.txt. Заполним файл данными в соответствии с номером варианта задания. В качестве разделителя столбцов данных в файле используем символ “;” без пробелов. Файл содержит 3 строки. Данные отличаются от введенных ранее (см.Рис.9).


```

lovanhung@lovanhung-VirtualBox:~/lovanhung/temp$ echo "WOLVES;Nuno;36;68" >> database3.txt
lovanhung@lovanhung-VirtualBox:~/lovanhung/temp$ echo "BH;Marti;42;68" >> database3.txt
lovanhung@lovanhung-VirtualBox:~/lovanhung/temp$ echo "AV;Tele;36;67" >>database3.txt
lovanhung@lovanhung-VirtualBox:~/lovanhung/temp$ echo "CP;Ali;35;65" >>database3.txt
lovanhung@lovanhung-VirtualBox:~/lovanhung/temp$ echo "NEW;Unai;40;60" >>database3.txt

```

Рисунок 9 – Создание и открытие текстового редактора для файла database3

Выведем на экран содержимое всех созданных файлов базы данных (см.Рис.10).

```

lovanhung@lovanhung-VirtualBox:~/lovanhung/temp$ cat database1.txt
MU;Ole;51;86
MC;Pep;57;84
LIV;Klop;56;81
CHEL;Tuchel;55;81
ARS;Ateta;49;78
lovanhung@lovanhung-VirtualBox:~/lovanhung/temp$ cat database2.txt
Tot;Konte;46;75
WH;David;52;75
LEI;Moye;40;74
LEEDS;Cheisa;60;72
EVER;Holding;41;70
lovanhung@lovanhung-VirtualBox:~/lovanhung/temp$ cat database3.txt
cat database3.txt: команда не найдена
lovanhung@lovanhung-VirtualBox:~/lovanhung/temp$ cat database3.txt
WOLVES;Nuno;36;68
BH;Marti;42;68
AV;Tele;36;67
CP;Ali;35;65
NEW;Unai;40;60

```

Рисунок 10 – Вывод содержимого файлов database1, database2, database3

Объединим содержимое всех созданных файлов базы данных в один файл data.txt и поместим его в каталог /database (см.Рис.11).

```

lovanhung@lovanhung-VirtualBox:~/lovanhung/temp$ cat database1.txt database2.txt database3.txt > ../database/data.txt
lovanhung@lovanhung-VirtualBox:~/lovanhung/temp$ cat ../database/data.txt
MU;Ole;51;86
MC;Pep;57;84
LIV;Klop;56;81
CHEL;Tuchel;55;81
ARS;Ateta;49;78
Tot;Konte;46;75
WH;David;52;75
LEI;Moye;40;74
LEEDS;Cheisa;60;72
EVER;Holding;41;70
WOLVES;Nuno;36;68
BH;Marti;42;68
AV;Tele;36;67
CP;Ali;35;65
NEW;Unai;40;60

```

Рисунок 11 – Объединение содержимого в один файл и проверка объединения

Перейдем в каталог /database. Убедимся, что он является текущим. Выведем на экран содержимое каталога (см. Рис.12).

```

lovanhung@lovanhung-VirtualBox:~/lovanhung/temp$ cd ../database
lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ ls
data.txt

```

Рисунок 12 – Переход в каталог database

Подсчитаем количество строк файла data.txt. Результат подсчета выведем на экран и в файл отчета output.txt, расположенный в каталоге report (см. Рис.13).

```
lovanhung@lovanhung-VirtualBox:~/lovanhung/temp$ cd ../database
lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ ls
data.txt
lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ wc -l data.txt
15 data.txt
lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ wc -l data.txt | cat > ../temp/report/output.txt
lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ nano data.txt
```

Рисунок 13 – Подсчет количества строк

Дополним файл data.txt 2-я строками данных в соответствии с номером варианта задания. В качестве разделителя столбцов данных в файле используем символ “;” без пробелов (см Рис.14).

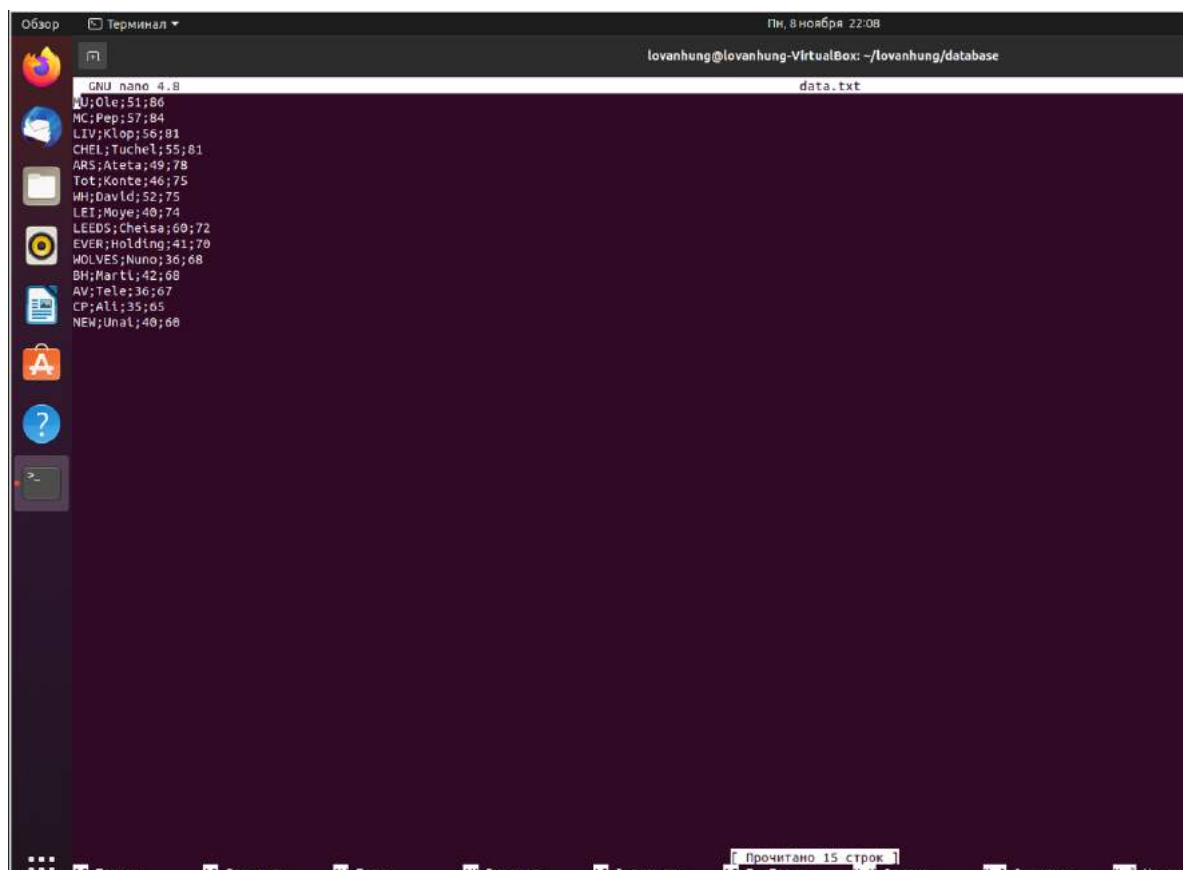


Рисунок 14 – Добавление строк

Повторно подсчитаем количество строк файла data.txt. Результат подсчета выведем на экран и допишем в конец файла отчета output.txt, расположенного в каталоге report (см. Рис.15).

```
lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ nano data.txt
lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ wc -l data.txt
15 data.txt
lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ wc -l data.txt | cat > ../temp/report/output.txt
bash: ../temp/report/output.txt: Нет такого файла или каталога
lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ wc -l data.txt | cat > ../temp/report/output.txt
```

Рисунок 15 – Подсчет количества строк и запись подсчета количества строк в файл

Осуществим фильтрацию данных файла data.txt в соответствии с номером варианта задания. Результат фильтрации выведем на экран и в файл отчета filtered.txt, расположенный в каталоге report. Повторим фильтрацию с различными значениями фильтра. Результаты фильтрации выведем на экран и допишем в файл отчета filtered.txt (см.Рис.16).

```
lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ grep "MU" data.txt
MU;Ole;51;86
lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ grep "LIV" data.txt
LIV;Klop;56;81
lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ grep "LEEDS" data.txt
LEEDS;Cheisa;60;72
lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ grep "AV" data.txt
AV;Tele;36;67
lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ grep "LEI" data.txt
LEI;Moye;40;74
lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ grep "LIV" data.txt
LIV;Klop;56;81
lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ grep "MU" data.txt | cat >> ../temp/report/filtered.txt
с: команда не найдена
lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ grep "MU" data.txt | cat >>^C
lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ grep "MU" data.txt | cat >> ../temp/report/filtered.txt
lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ grep "LIV" data.txt | cat >> ../temp/report/filtered.txt
lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ grep "LEEDS" data.txt | cat >> ../temp/report/filtered.txt
lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ grep "AV" data.txt | cat >> ../temp/report/filtered.txt
lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ grep "LEI" data.txt | cat >> ../temp/report/filtered.txt
lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ cat ../temp/report/filtered.txt
cat: ../temp/report/filtered.txt: Нет такого файла или каталога
lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ cat ../temp/report/filtered.txt
MU;Ole;51;86
LIV;Klop;56;81
LEEDS;Cheisa;60;72
AV;Tele;36;67
LEI;Moye;40;74
```

Рисунок 16 – Вывод фильтрации в консоль и запись в файл

Выполним сортировку содержимого файла data.txt в соответствии с номером варианта задания. Результат сортировки выведем на экран и в файл отчета sorted.txt, расположенный в каталоге report (см. Рис.17).

```
lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ sort -t ';' -k2 data.txt | cat > ../temp/report/sorted.txt
lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ cat ../temp/report/sorted.txt
CP;Ali;35;65
ARS;Ateta;49;78
LEEDS;Cheisa;60;72
WH;David;52;75
EVER;Holding;41;70
LIV;Klop;56;81
Tot;Konte;46;75
BH;Marti;42;68
LEI;Moye;40;74
WOLVES;Nuno;36;68
MU;Ole;51;86
MC;Pep;57;84
AV;Tele;36;67
CHEL;Tuchel;55;81
NEW;Unai;40;60
```

Рисунок 17 - Сортировка

Выполним фильтрацию содержимого файла data.txt с сортировкой результата фильтрации. Фильтрацию и сортировку выполним в соответствии с номером варианта задания. Результат выведем на экран и в файл отчета filteredsorted.txt, расположенный в каталоге report (см. Рис.18).

```

lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ cat data.txt | grep -v "EVER" | sort -t ";" -k2 > ../temp/report/filteredsorted.txt
lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ cat ../temp/report/filteredsorted.txt
CP;Ali;35;65
ARS;Ateta;49;78
LEEDS;Chelsea;60;72
WH;David;52;75
LIV;Klop;56;81
Tot;Konte;46;75
BH;Martl;42;68
LEI;Moye;40;74
WOLVES;Nuno;36;68
MU;Ole;51;86
MC;Pep;57;84
AV;Tele;36;67
CHEL;Tuchel;55;81
NEW;Unai;40;60

```

Рисунок 18 – Фильтрация и сортировка

Выполним команду вывода календаря на экран и записи его значения в файл calendar.txt, находящийся в каталоге /database. Результат вывести на экран

(см.Рис.19).

```

lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ cal
  Ноябрь 2021
Вс Пн Вт Ср Чт Пт Сб
    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

lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ cal > calendar.txt
lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ cat calendar.txt
  Ноябрь 2021
Вс Пн Вт Ср Чт Пт Сб
    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

```

Рисунок 19– Отображение, запись и вывод данных календаря

ВЫВОДЫ

В данной лабораторной работе мы приобрели навыки работы с файлами и каталогами, познакомились с некоторыми командами манипулирования данными на примере текстовой базы данных.

Лабораторная работа №10

«Использование программируемого фильтра awk»

Цель работы:

Лабораторная работа выполняется в среде, установленной и настроенной в процессе выполнения лабораторной работы №8 или в среде, установленной в компьютерном классе.

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

Задание:

1. Вывести на экран из файла calendar.txt день недели и текущее число:

```
lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ cat calendar.txt
    Ноября 2021
Вс Пн Вт Ср Чт Пт Сб
   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
```

Рис. 1 – вывод файла calendar.txt

```
lovanhung@lovanhung-VirtualBox:~/lovanhung/database$ awk 'BEGIN {print "Сегодня"}' && awk 'NR == 2{print $4}' calendar.txt && awk 'NR==4 {print $4}' calendar.txt && awk 'NR==1{print $1}' calendar.txt
Сегодня
Ср
10
Ноября
```

Рис. 2 – вывод текущего числа

2. Вывести список каталогов, имена которых состоят из русских букв, без дополнительных полей:

```
lovanhung@lovanhung-VirtualBox:~$ ls -l | awk '$9~/[а-я]/ {print $9}'
Hunglo
LoHung
lovanhung
snap
Видео
Документы
Загрузки
Изображения
Музыка
Общедоступные
Рабочий
Шаблоны
```

Рис. 3 – вывод списка каталогов

3. Определить количество(сумму) байтов, занятых всеми вашими текстовыми файлами (txt) в каталогах и подкаталогах;

```
lovanhung@lovanhung-VirtualBox:~/lovanhung$ find . -type f -exec du -ch {} + -exec {} \;
find: './temp/database1.txt': Отказано в доступе
find: './temp/database2.txt': Отказано в доступе
find: './temp/database3.txt': Отказано в доступе
find: './temp/report/output.txt': Отказано в доступе
find: './temp/report/filteredsorted.txt': Отказано в доступе
find: './temp/report/filtered.txt': Отказано в доступе
find: './temp/report/sorted.txt': Отказано в доступе
find: './database/calendar.txt': Отказано в доступе
find: './database/data.txt': Отказано в доступе
4,0K    ./temp/database1.txt
4,0K    ./temp/database2.txt
4,0K    ./temp/database3.txt
4,0K    ./temp/report/output.txt
4,0K    ./temp/report/filteredsorted.txt
4,0K    ./temp/report/filtered.txt
4,0K    ./temp/report/sorted.txt
4,0K    ./database/calendar.txt
4,0K    ./database/data.txt
36K     итого
```

Рис. 4 – вывод кол-ва байтов, занятых .txt файлами

4. Определить количество блоков, содержащих ваш текущий каталог:

```
lovanhung@lovanhung-VirtualBox:~$ ls -l | awk '{sum+=$2} END {print sum}'
76
```

Рис. 5 – кол-во блоков, содержащих текущий каталог

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

```
lovanhung@lovanhung-VirtualBox:~$ cd lovanhung/temp
lovanhung@lovanhung-VirtualBox:~/lovanhung/temp$ ls -l
итого 16
-rw-rw-r-- 1 lovanhung lovanhung 75 ноя 8 21:38 database1.txt
-rw-rw-r-- 1 lovanhung lovanhung 84 ноя 8 21:48 database2.txt
-rw-rw-r-- 1 lovanhung lovanhung 75 ноя 8 21:56 database3.txt
drwxrwxr-x 2 lovanhung lovanhung 4096 ноя 8 23:15 report
lovanhung@lovanhung-VirtualBox:~/lovanhung/temp$ chmod g+rx database1.txt
lovanhung@lovanhung-VirtualBox:~/lovanhung/temp$ chmod g+s database3.txt
lovanhung@lovanhung-VirtualBox:~/lovanhung/temp$ ls -l
итого 16
-rw-rwxr-- 1 lovanhung lovanhung 75 ноя 8 21:38 database1.txt
-rw-rw-r-- 1 lovanhung lovanhung 84 ноя 8 21:48 database2.txt
-rw-rwSr-- 1 lovanhung lovanhung 75 ноя 8 21:56 database3.txt
drwxrwxr-x 2 lovanhung lovanhung 4096 ноя 8 23:15 report
lovanhung@lovanhung-VirtualBox:~/lovanhung/temp$ sort -k1 | ls -l | awk '{print $9}'
database1.txt
database2.txt
database3.txt
report
```

Рис. 6 – изменение прав доступа файлов

6. Напечатать список каталогов, в которых обнаружены файлы с именами data*.txt:

```
lovanhung@lovanhung-VirtualBox:~$ find . -name data*.txt
./lovanhung/temp/database1.txt
./lovanhung/temp/database2.txt
./lovanhung/temp/database3.txt
./lovanhung/database/data.txt
```

Рис. 7 – список файлов data*.txt

7. Подсчитать, сколько раз пользователь входил в систему:

```
lovanhung@lovanhung-VirtualBox:~$ last | awk 'BEGIN {count = 0} {if($1=="lovanhung")} {count++} END {print count}'
6
```

Рис. 8 – кол-во входов в систему

8. Напечатать список пользователей, отсортированный по времени:

```
lovanhung@lovanhung-VirtualBox:~$ ls -l | grep lovanhung | awk '{print $4, $5, $6, $7, $8, $9}' | sort -nk4
lovanhung 4096 ноя 8 19:26 Видео
lovanhung 4096 ноя 8 19:26 Документы
lovanhung 4096 ноя 8 19:26 Загрузки
lovanhung 4096 ноя 8 19:26 Музыка
lovanhung 4096 ноя 8 19:26 Общедоступные
lovanhung 4096 ноя 8 19:26 Рабочий
lovanhung 4096 ноя 8 19:26 Шаблоны
lovanhung 4096 ноя 8 20:42 Hunglo
lovanhung 4096 ноя 8 20:44 LoHung
lovanhung 4096 ноя 8 20:49 lovanhung
lovanhung 4096 ноя 8 21:05 snar
lovanhung 4096 ноя 15 21:50 Изображения
```

Рис. 9 – отсортированный список пользователей

Вывод:

В данной лабораторной работе мною были использованы и частично освоены возможности программируемого фильтра `awk`. Фильтр широко применяется для обработки данных и формирования различного вида отчетов.