

# **Лабораторная работа №1.**

**Установка ОС Linux.**

Кучеренко София

# Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Домашнее задание.	20
5	Выводы	27

# Список иллюстраций

3.1	Создание Linux . . . . .	7
3.2	Название и тип ОС . . . . .	8
3.3	Объём оперативной памяти . . . . .	9
3.4	Жёсткий диск . . . . .	10
3.5	Тип файла . . . . .	11
3.6	Формат хранения . . . . .	12
3.7	Имя и размер файла . . . . .	13
3.8	Запуск . . . . .	14
3.9	Установка Linux . . . . .	15
3.10	Выбор устройств . . . . .	16
3.11	Носители . . . . .	17
3.12	Окно регистрации . . . . .	18
3.13	Команда «sudo dnf install mc» . . . . .	18
3.14	Команда «sudo dnf install git» . . . . .	18
3.15	Команда «sudo dnf install nasm» . . . . .	19

## Список таблиц

# 1 Цель работы

Приобретение практических навыков установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов.

## 2 Задание

Необходимо установить операционную систему Linux на виртуальную машину, запустить минимально необходимые для дальнейшей работы сервисы и программы. 1. Запустить виртуальную машину, добавить ОС Linux 2. Установить объем оперативной памяти 3. Создать виртуальный жесткий диск и установить его объем 4. Установить Linux, выбрать язык и носитель 5. Создать логин и пароль 6. Установить программы через терминал 7. Выполнить домашнюю работу и ответить на вопросы

## 3 Выполнение лабораторной работы

1. Запустим виртуальную машину и нажмём кнопку «создать»:

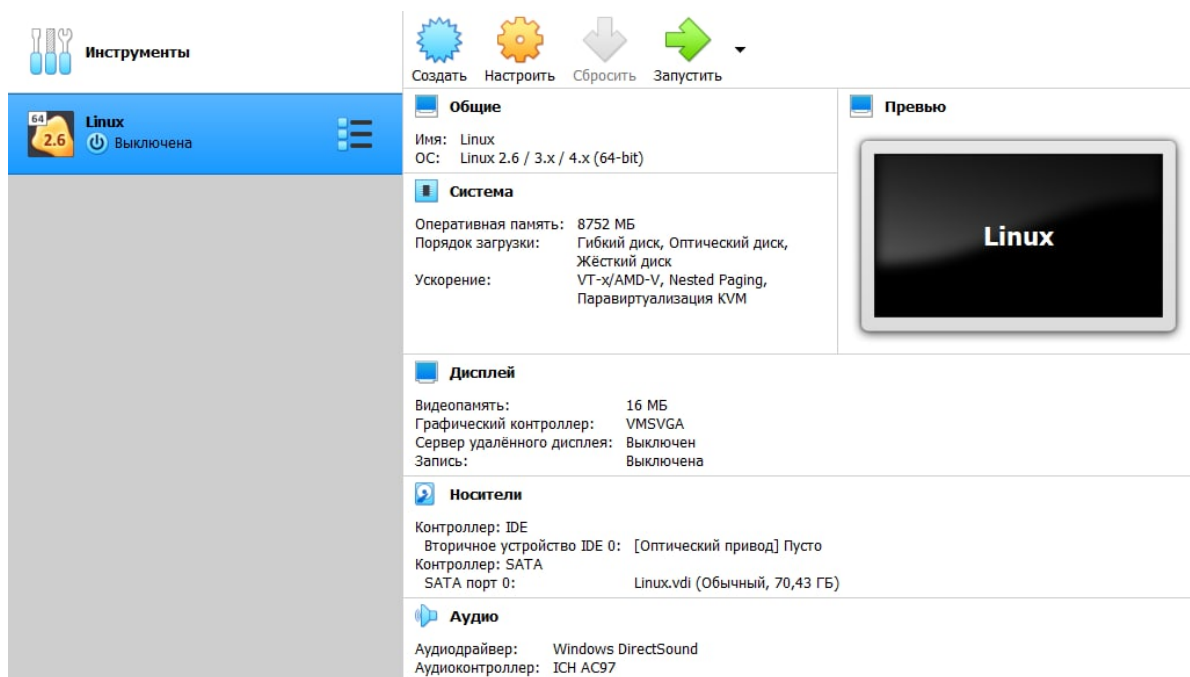


Рис. 3.1: Создание Linux

Введём название нашей ОС и выберем тип:

Создать виртуальную машину

### Укажите имя и тип ОС

Пожалуйста укажите имя и местоположение новой виртуальной машины и выберите тип операционной системы, которую Вы собираетесь установить на данную машину. Заданное Вами имя будет использоваться для идентификации данной машины.

Имя:

Папка машины:

Тип:

Версия:

Экспертный режим **Далее** Отмена

Рис. 3.2: Название и тип ОС

Укажем объём оперативной памяти, выделенный виртуальной машине:



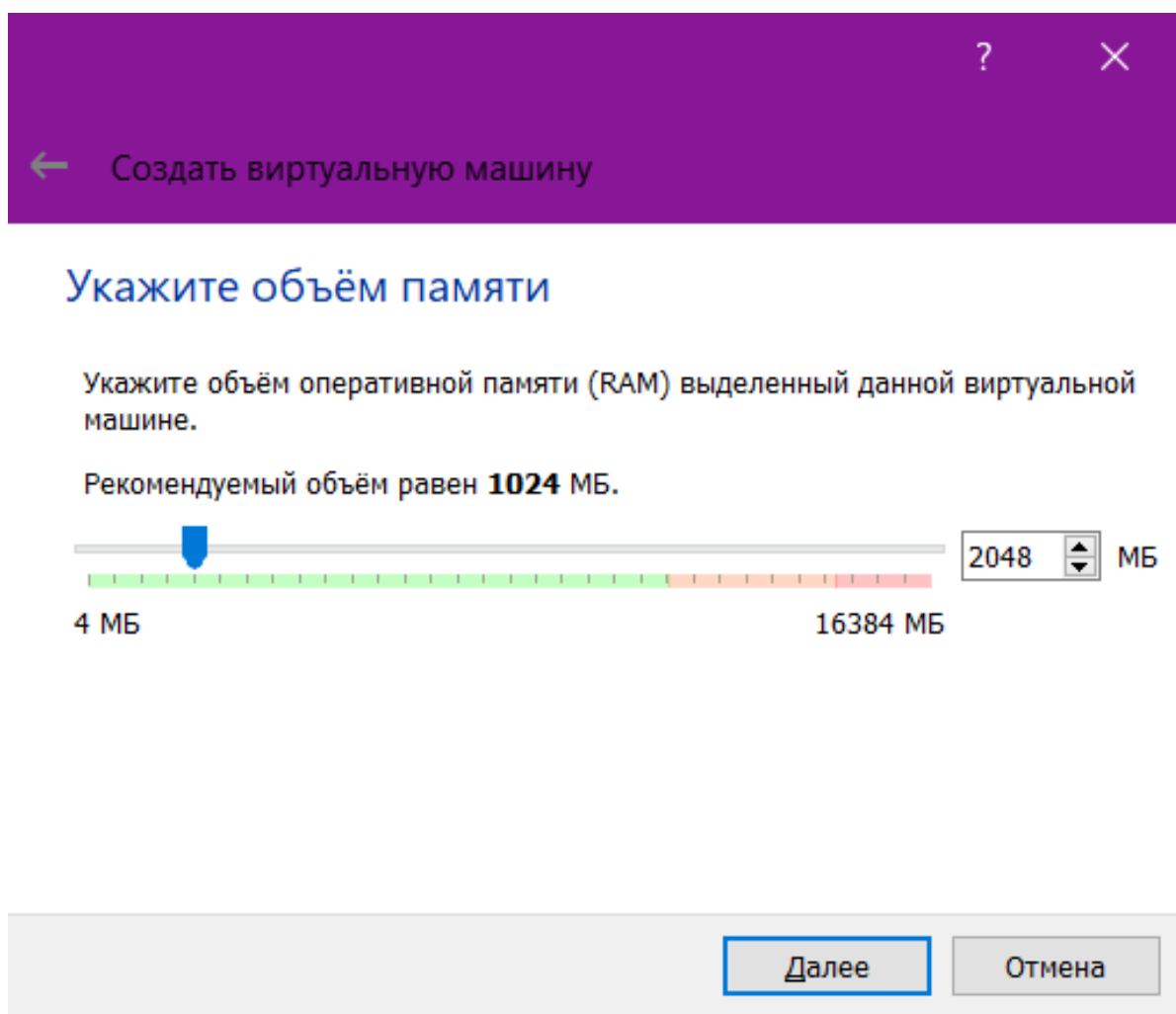


Рис. 3.3: Объем оперативной памяти

Создадим новый виртуальный диск:

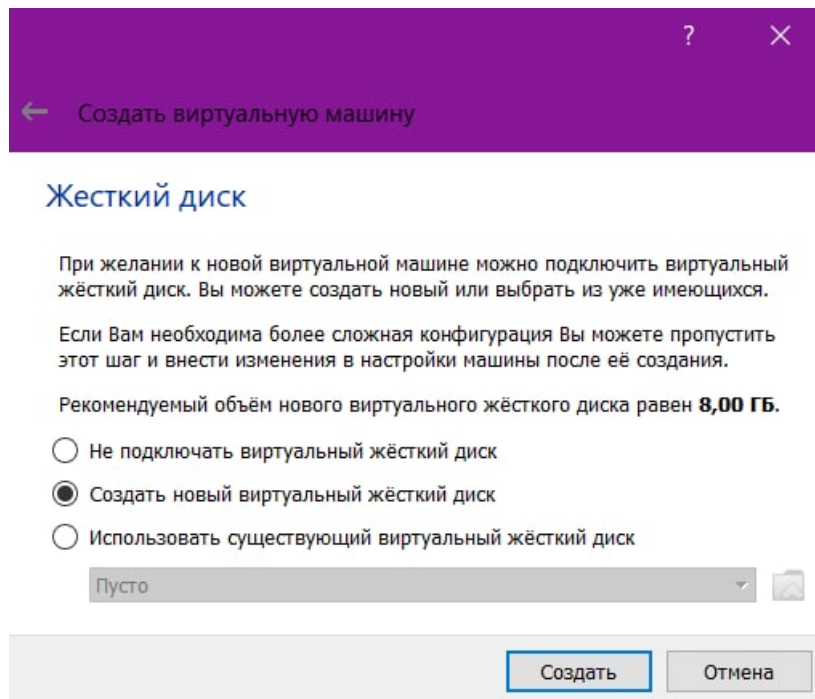


Рис. 3.4: Жёсткий диск

Выбираем формат жёсткого диска:

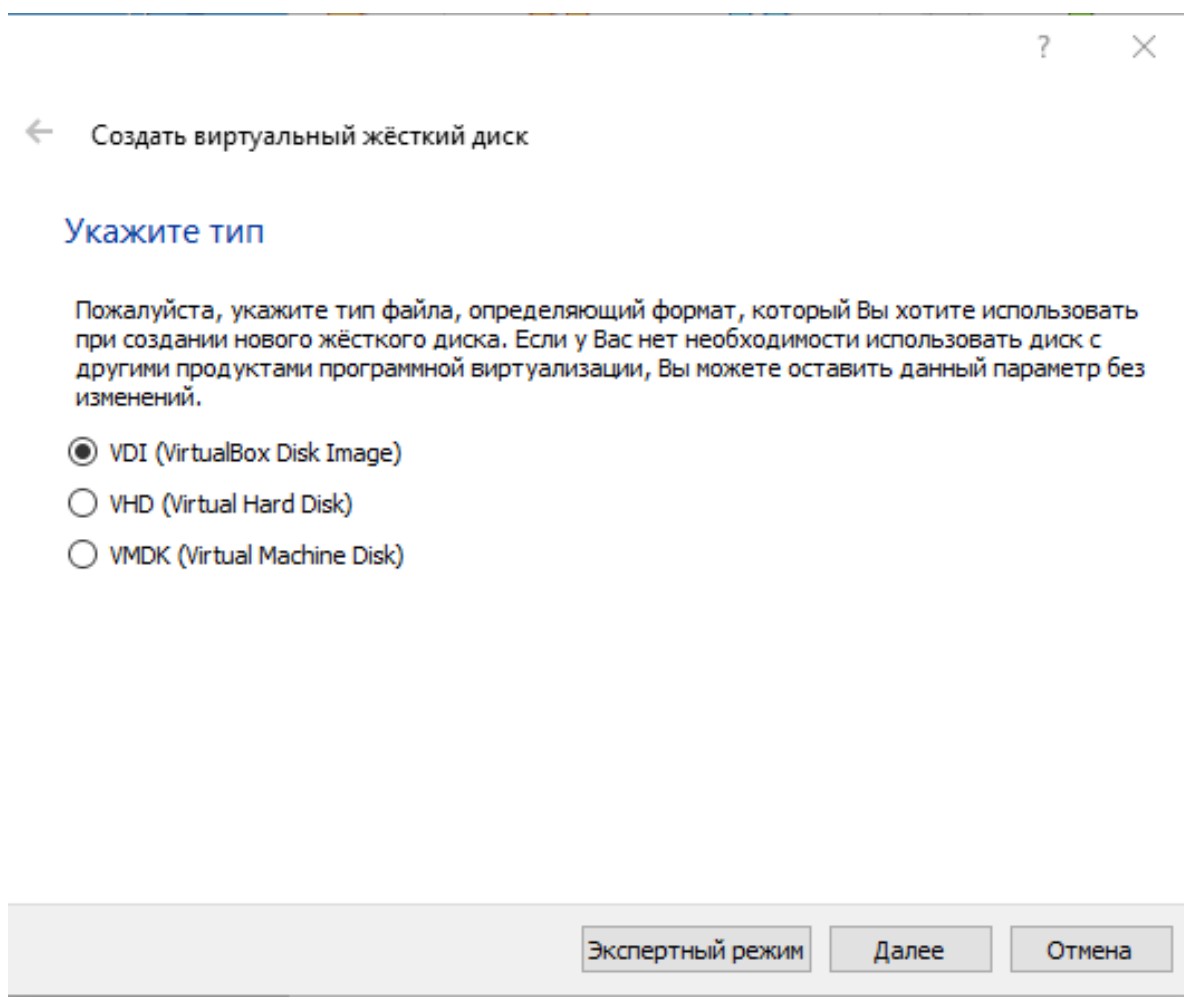


Рис. 3.5: Тип файла

Выбираем «динамический виртуальный жесткий диск»:

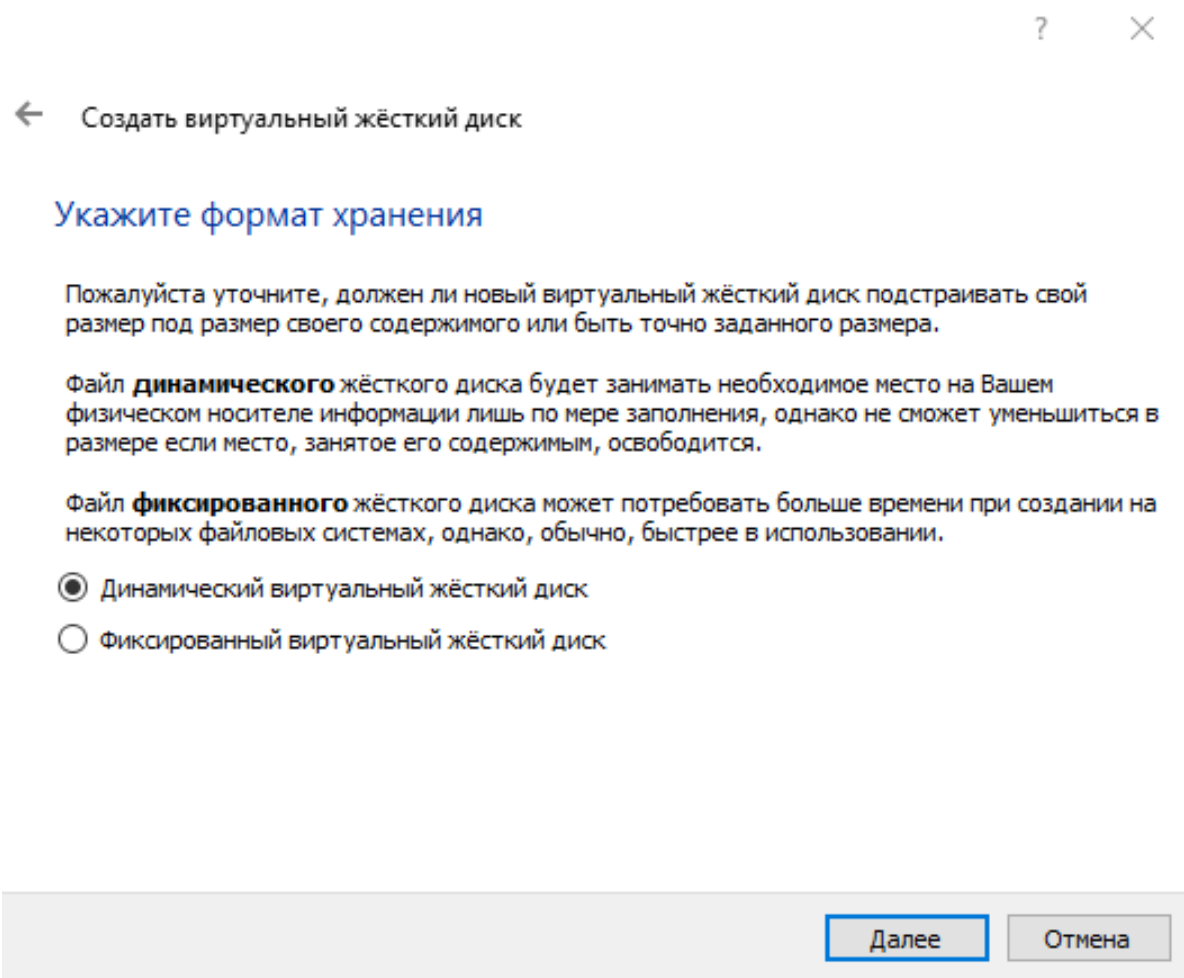


Рис. 3.6: Формат хранения

Указываем название жёсткого диска и указываем его размер:

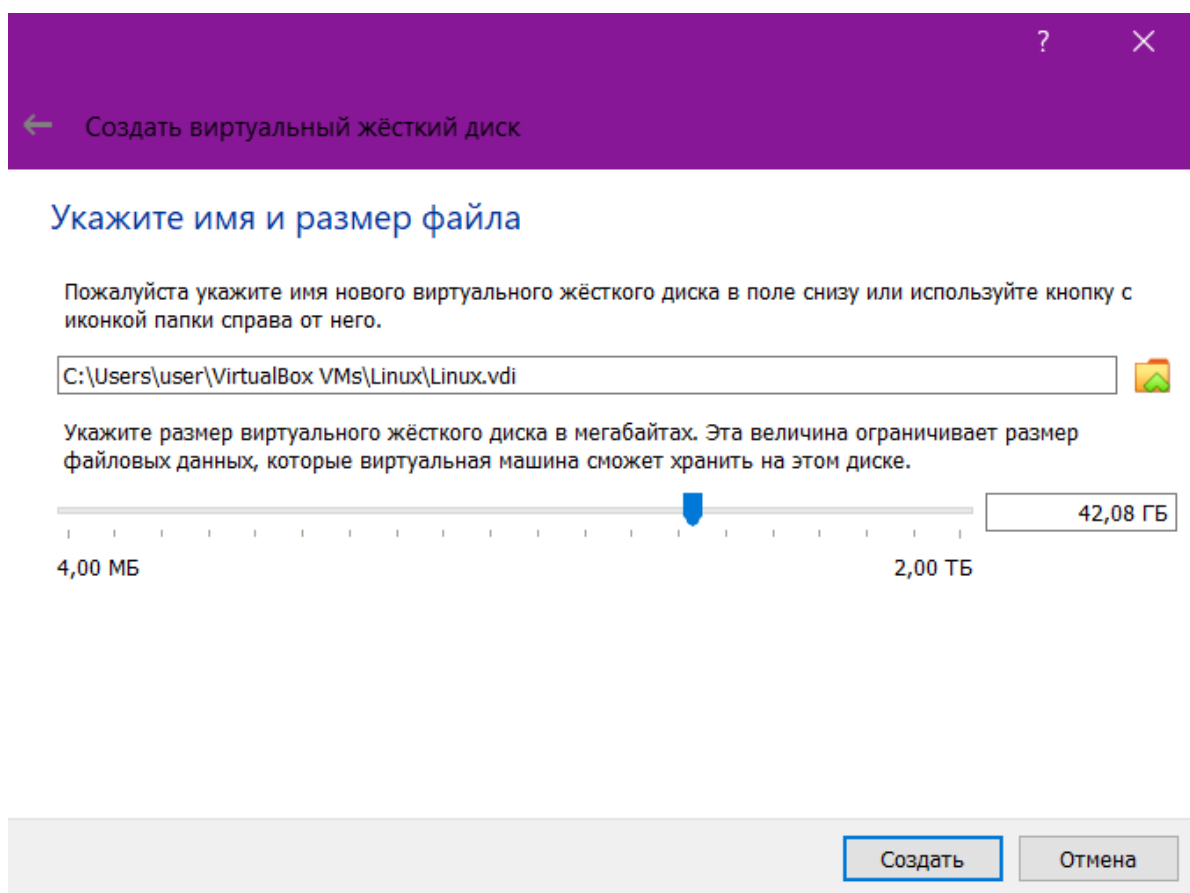


Рис. 3.7: Имя и размер файла

Запускаем виртуальную машину:

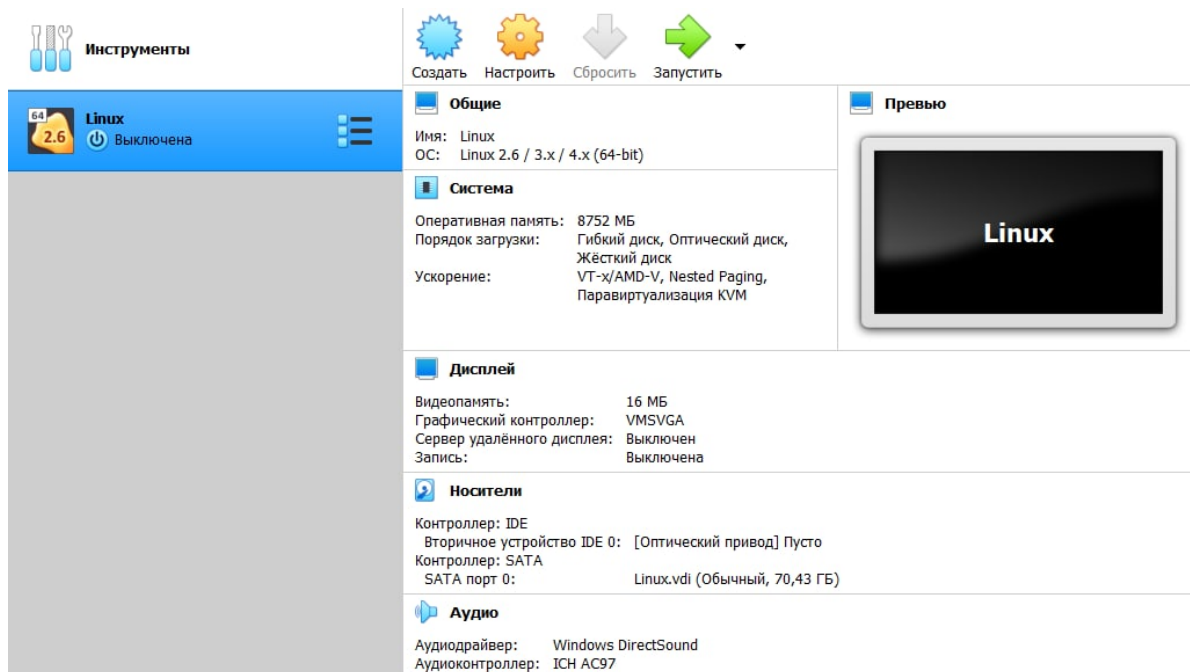


Рис. 3.8: Запуск

Устанавливаем Linux:

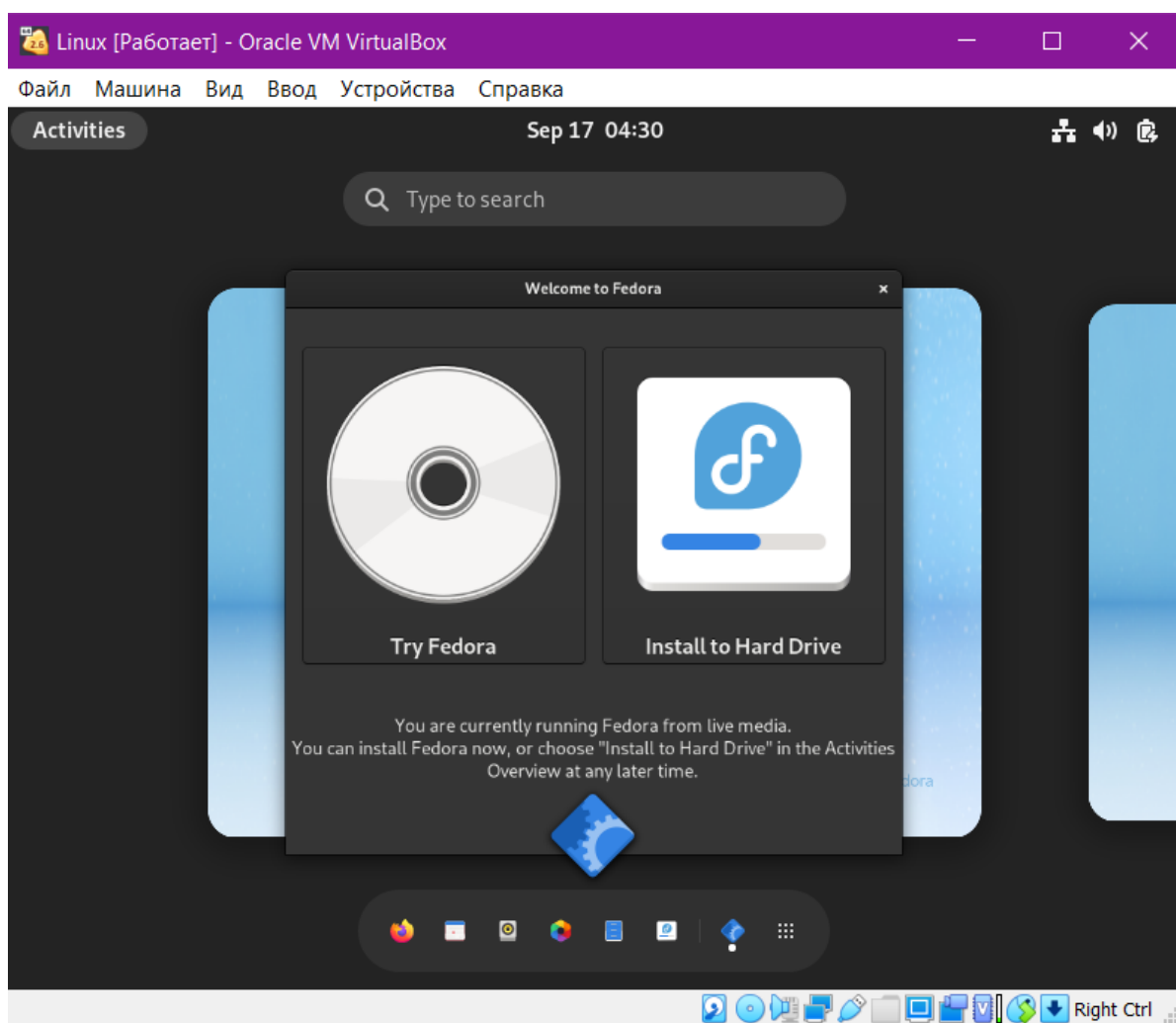


Рис. 3.9: Установка Linux

Выбираем устройство для хранения виртуальной систем (отмечаем, чтобы появилась галочка):

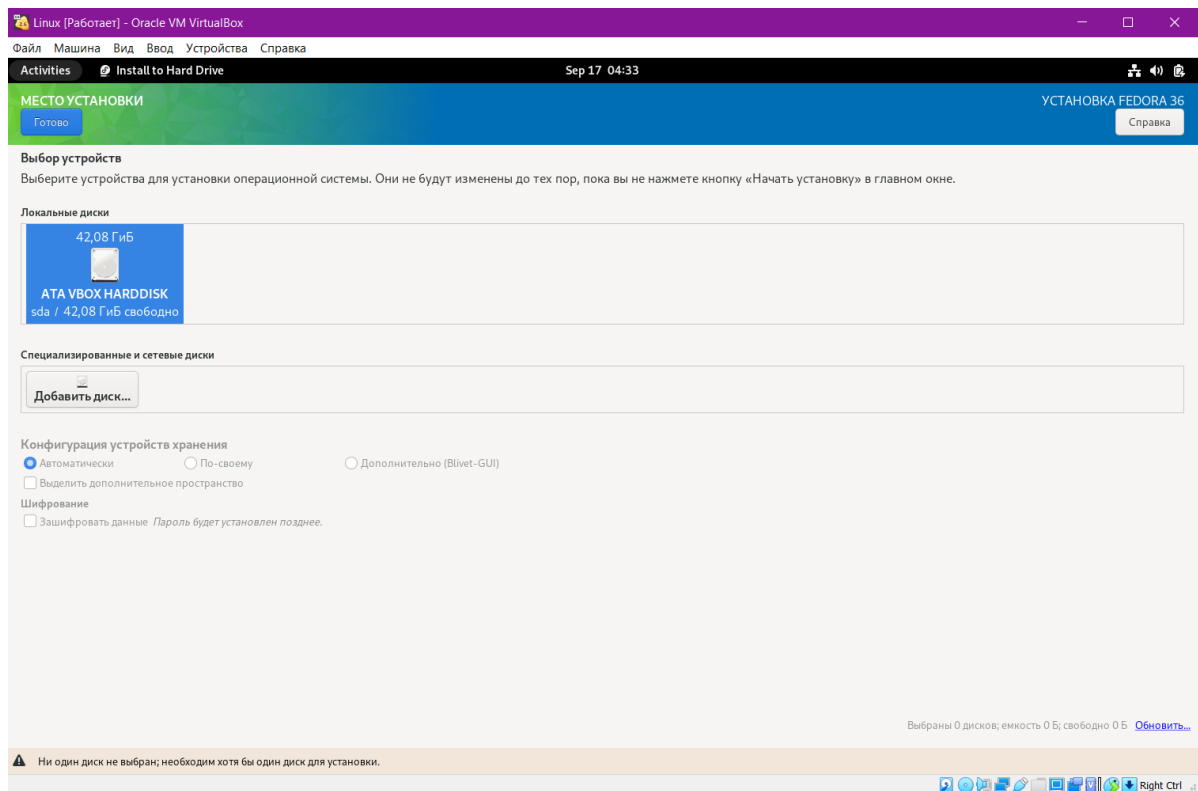


Рис. 3.10: Выбор устройств

Удаляем носитель “Fedora”:



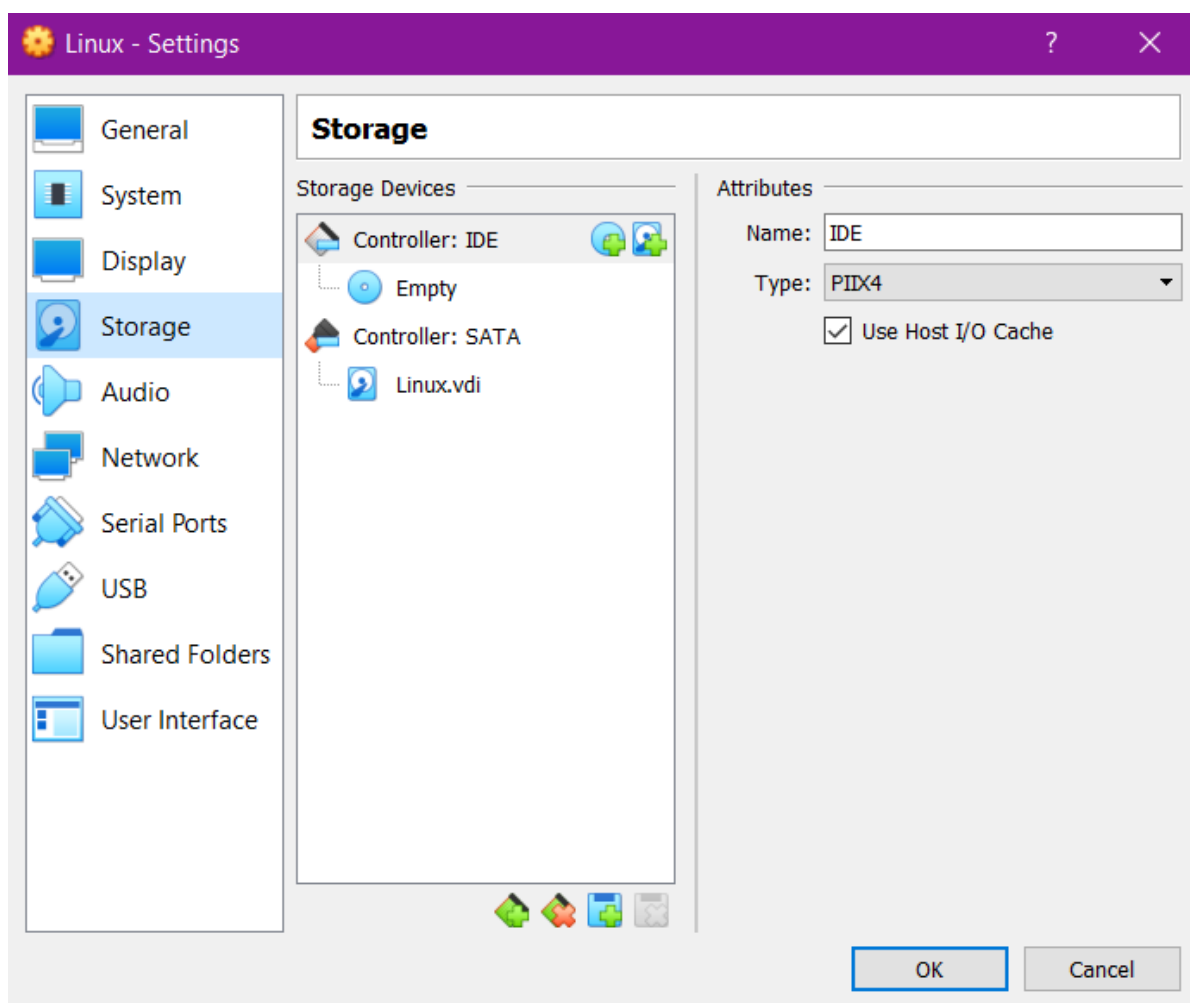
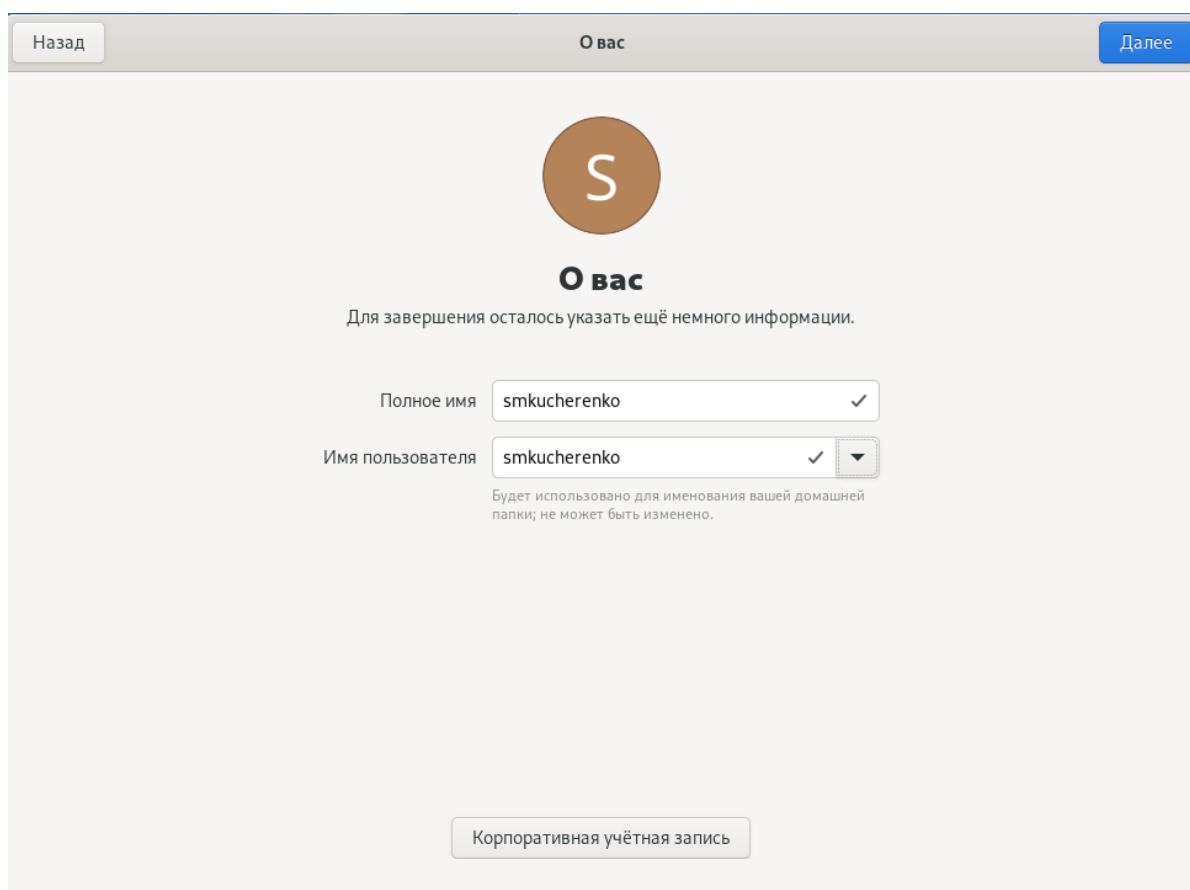


Рис. 3.11: Носители

Создаём логин и пароль:



The image shows a registration window with a light gray background. At the top, there is a navigation bar with three buttons: "Назад" (Back) on the left, "О вас" (About you) in the center, and "Далее" (Next) on the right. Below the navigation bar, there is a large circular profile picture placeholder with a white letter 'S' on a brown background. Underneath the profile picture, the text "О вас" is displayed in bold. Below this, a message states: "Для завершения осталось указать ещё немного информации." (To complete, you still need to specify a little more information). There are two input fields: "Полное имя" (Full name) and "Имя пользователя" (Username). Both fields contain the text "smkucherenko" and have a checkmark icon on the right. Below the "Имя пользователя" field, there is a small note: "Будет использовано для именования вашей домашней папки; не может быть изменено." (Will be used for naming your home folder; cannot be changed). At the bottom of the form, there is a button labeled "Корпоративная учётная запись" (Corporate account).

Рис. 3.12: Окно регистрации

Устанавливаем Midnight Commander, git, nasm командой при помощи терминала:

```
[smkucherenko@fedora ~]$ sudo dnf install mc
```

Рис. 3.13: Команда «sudo dnf install mc»

```
[smkucherenko@fedora ~]$ sudo dnf install git
```

Рис. 3.14: Команда «sudo dnf install git»

```
[smkucherenko@fedora ~]$ sudo dnf install nasm
```

Рис. 3.15: Команда «sudo dnf install nasm»

## 4 Домашнее задание.

1. Версия ядра Linux: 5.2.0-kali2-amd64
2. Частота процессора: 1995.390 МГц.
3. Модель процессора: Intel(R) Core(TM) i3-5005U CPU @ 2.00GHz
4. Объем доступной оперативной памяти: 2096628 Кб.
5. Тип обнаруженного гипервизора: Vmware.
6. Тип файловой системы корневого раздела и последовательность монтирования файловых систем: Тип файловой системы корневого раздела – EXT4.

### #Контрольные вопросы

- 1) Учетная запись пользователя – это необходимая для системы. Информация о пользователе, хранящаяся в специальных файлах. Информация используется Linux для аутентификации пользователя и назначения ему прав доступа. Аутентификация – системная процедура, позволяющая Linux определить, какой именно пользователь осуществляет вход. Вся информация о пользователе обычно хранится в файлах /etc/passwd и /etc/group. Учётная запись пользователя содержит:
  - Имя пользователя (user name)
  - Идентификационный номер пользователя (UID)
  - Идентификационный номер группы (GID).
  - Пароль (password)
  - Полное имя (full name)

- Домашний каталог (home directory)
- Начальную оболочку (login shell)

## 2) Команды терминала:

- Для получения справки по команде: `man [команда]`. Например, команда «`man ls`» выведет справку о команде «`ls`».
- Для перемещения по файловой системе: `cd [путь]`. Например, команда «`cd newdir`» осуществляет переход в каталог `newdir`
- Для просмотра содержимого каталога: `ls [опции] [путь]`. Например, команда «`ls -a ~/newdir`» отобразит имена скрытых файлов в каталоге `newdir`
- Для определения объёма каталога: `du [опция] [путь]`. Например, команда «`du -k ~/newdir`» выведет размер каталога `newdir` в килобайтах
- Для создания / удаления каталогов / файлов: `mkdir [опции] [путь] / rmdir [опции] [путь] / rm [опции] [путь]`. Например, команда «`mkdir -p ~/newdir1/newdir2`» создаст иерархическую цепочку подкаталогов, создав каталоги `newdir1` и `newdir2`; команда «`rmdir -v ~/newdir`» удалит каталог `newdir`; команда «`rm -r ~/newdir`» так же удалит каталог `newdir`
- Для задания определённых прав на файл / каталог: `chmod [опции] [путь]`. Например, команда «`chmod g+r ~/text.txt`» даст группе право на чтение файла `text.txt`
- Для просмотра истории команд: `history [опции]`. Например, команда «`history 5`» покажет список последних 5 команд

## 3) Файловая система имеет два значения: с одной стороны – это архитектура хранения битов на жестком диске, с другой – это организация каталогов в соответствии с идеологией Unix. Файловая система (англ. «file system») – это архитектура хранения данных в системе, хранение данных в оперативной памяти и доступа к конфигурации ядра. Файловая система устанавливает физическую и логическую структуру файлов, правила их создания и управления ими. В физическом смысле файловая система Linux представляет

собой пространство раздела диска, разбитое на блоки фиксированного размера. Их размер кратен размеру сектора: 1024, 2048, 4096 или 8120 байт. Существует несколько типов файловых систем:

- XFS – начало разработки 1993 год, фирма Silicon Graphics, в мае 2000 года предстала в GNU GPL, для пользователей большинства Linux систем стала доступна в 2001-2002 гг. Отличительная черта системы – прекрасная поддержка больших файлов и файловых томов, 8 эксбибайт ( $8 \cdot 260$  байт) для 64-х битных систем.
- ReiserFS (Reiser3) – одна из первых журналируемых файловых систем под Linux, разработана Namesys, доступна с 2001 г. Максимальный объём тома для этой системы равен 16 тебибайт ( $16 \cdot 240$  байт).
- JFS (Journaled File System) – файловая система, детище IBM, явившееся миру в далёком 1990 году для ОС AIX (Advanced Interactive eXecutive). В виде первого стабильного релиза, для пользователей Linux, система стала доступна в 2001 году. Из плюсов системы – хорошая масштабируемость. Из минусов – не особо активная поддержка на протяжении всего жизненного цикла. Максимальный размер тома 32 пэбибайта ( $32 \cdot 250$  байт).
- ext (extended filesystem) – появилась в апреле 1992 года, это была первая файловая система, изготовленная специально под нужды Linux ОС. Разработана Remy Card с целью преодолеть ограничения файловой системы Minix.
- ext2 (second extended file system) – была разработана Remy Card в 1993 году. Не журналируемая файловая система, это был основной её недостаток, который исправит ext3.
- ext3 (third extended filesystem) – по сути расширение исконной для Linux ext2, способное к журналированию. Разработана Стивеном Твиди (Stephen Tweedie) в 1999 году, включена в основное ядро Linux в ноябре 2001 года. На фоне других своих сослуживцев обладает более скромным размером пространства, до 4 тебибайт ( $4 \cdot 240$  байт) для 32-х разрядных систем. На дан-

ный момент является наиболее стабильной и поддерживаемой файловой системой в среде Linux.

- Reiser4 – первая попытка создать файловую систему нового поколения для Linux. Впервые представленная в 2004 году, система включает в себя такие передовые технологии как транзакции, задержка выделения пространства, а так же встроенная возможность кодирования и сжатия данных. Ханс Рейзер (Hans Reiser) – главный разработчик системы.
- ext4 – попытка создать 64-х битную ext3 способную поддерживать больший размер файловой системы (1 эксбибайт). Позже добавились возможности – непрерывные области дискового пространства, задержка выделения пространства, онлайн дефрагментация и прочие. Обеспечивается прямая совместимость с системой ext3 и ограниченная обратная совместимость при недоступной способности к непрерывным областям дискового пространства.
- Btrfs (B-tree FS или Butter FS) – проект изначально начатый компанией Oracle, впоследствии поддержанный большинством Linux систем. Ключевыми особенностями данной файловой системы являются технологии: *copy-on-write*, позволяющая сделать снимки областей диска (снапшоты), которые могут пригодиться для последующего восстановления; контроль за целостностью данных и метаданных (с повышенной гарантией целостности); сжатие данных; оптимизированный режим для накопителей SSD (задаётся при монтировании) и прочие. Немаловажным фактором является возможность перехода с ext3 на Btrfs. С августа 2008 года данная система выпускается под GNU GPL.
- Tux2 – известная, но так и не анонсированная публично файловая система. Создатель Дэниэл Филипс (Daniel Phillips). Система базируется на алгоритме «Фазового Древа», который как и журналирование защищает файловую систему от сбоев. Организована как надстройка на ext2.
- Tux3 – система создана на основе FUSE (Filesystem in Userspace), специаль-

- ного модуля для создания файловых систем на Unix платформах. Данный проект ставит перед собой цель избавиться от привычного журналирования, взамен предлагая версионное восстановление (состояние в определённый промежуток времени). Преимуществом используемой в данном случае версионной системы, является способ описания изменений, где для каждого файла создаётся изменённая копия, а не переписывается текущая версия.
- Xiafs – задумка и разработка данной файловой системы принадлежат Frank Xia, основана на файловой системе MINIX. В настоящее время считается устаревшей и практически не используется. Наряду с ext2 разрабатывалась, как замена системе ext. В декабре 1993 года система была добавлена в стандартное ядро Linux. И хотя система обладала большей стабильностью и занимала меньше дискового пространства под контрольные структуры – она оказалась слабее ext2, ведущую роль сыграли ограничения максимальных размеров файла и раздела, а так же способность к дальнейшему расширению.
  - ZFS (Zettabyte File System) – изначально созданная в Sun Microsystems файловая система, для неизвестной операционной системы Solaris в 2005 году. Отличительные особенности – отсутствие фрагментации данных как таковой, возможности по управлению снапшотами (snapshots), пулами хранения (storage pools), варьируемый размер блоков, 64-х разрядный механизм контрольных сумм, а так же способность адресовать 128 бит информации. В Linux системах может использоваться посредством FUSE.
- 4) Команда «findmnt» или «findmnt –all» будет отображать все подмонтированные файловые системы или искать файловую систему.
  - 5) Основные сигналы (каждый сигнал имеет свой номер), которые используются для завершения процесса:
- SIGINT – самый безобидный сигнал завершения, означает Interrupt. Он отправляется процессу, запущенному из терминала с помощью сочетания



клавиш Ctrl+C. Процесс правильно завершает все свои действия и возвращает управление;

- SIGQUIT – это еще один сигнал, который отправляется с помощью сочетания клавиш, программе, запущенной в терминале. Он сообщает ей что нужно завершиться и программа может выполнить корректное завершение или проигнорировать сигнал. В отличие от предыдущего, она генерирует дампы памяти. Сочетание клавиш Ctrl+Q;
- SIGHUP – сообщает процессу, что соединение с управляющим терминалом разорвано, отправляется, в основном, системой при разрыве соединения с интернетом;
- SIGTERM – немедленно завершает процесс, но обрабатывается программой, поэтому позволяет ей завершить дочерние процессы и освободить все ресурсы;
- SIGKILL – тоже немедленно завершает процесс, но, в отличие от предыдущего варианта, он не передается самому процессу, а обрабатывается ядром. Поэтому ресурсы и дочерние процессы остаются запущенными. Также для передачи сигналов процессам в Linux используется утилита kill, её синтаксис: kill [-сигнал] [pid\_процесса] (PID – уникальный идентификатор процесса). Сигнал представляет собой один из выше перечисленных сигналов для завершения процесса. Перед тем, как выполнить остановку процесса, нужно определить его PID. Для этого используют команды ps и grep. Команда ps предназначена для вывода списка активных процессов в системе и информации о них. Команда grep запускается одновременно с ps (в канале) и будет выполнять поиск по результатам команды ps. Утилита pkill – это оболочка для kill, она ведет себя точно так же, и имеет тот же синтаксис, только в качестве идентификатора процесса ей нужно передать его имя. killall работает аналогично двум предыдущим утилитам. Она тоже принимает имя процесса в качестве параметра и ищет его PID в директории /proc. Но эта утилита обнаружит все процессы с таким именем и завершит

ИХ.

## 5 Выводы

В ходе данной лабораторной работы я изучила, как установить операционную систему на виртуальную машину и настроить минимально необходимые для дальнейшей работы сервисы, а также приобрела навыки работы с консолью.