

# **Лабораторная работа № 1**

Установка и конфигурация операционной системы на  
виртуальную машину

Никулина Ксения Ильинична

НММбд-02-22

## **Содержание**

- 1. Цель работы**
- 2. Задание**
- 3. Теоретическое введение**
- 4. Выполнение лабораторной работы**
- 5. Самостоятельная работа**
- 6. Вывод**

## **1. Цель работы**

Целью данной работы является приобретение практических навыков установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов.

## **2. Задание**

Установить и минимально настроить операционную систему на виртуальную машину.

### **3. Теоретическое введение**

#### **3.1. Введение в GNU Linux**

Операционная система (ОС) — это комплекс взаимосвязанных программ, предназначенных для управления ресурсами компьютера и организации взаимодействия с пользователем. Сегодня наиболее известными операционными системами являются ОС семейства Microsoft Windows и UNIX-подобные системы.

GNU Linux — семейство переносимых, многозадачных и многопользовательских операционных систем, на базе ядра Linux, включающих тот или иной набор утилит и программ проекта GNU, и, возможно, другие компоненты. Как и ядро Linux, системы на его основе, как правило, создаются и распространяются в соответствии с моделью разработки свободного и открытого программного обеспечения (Open-Source Software). Linux-системы распространяются в основном бесплатно в виде различных дистрибутивов.

Дистрибутив GNU Linux — общее определение ОС, использующих ядро Linux и набор библиотек и утилит, выпускаемых в рамках проекта GNU, а также графическую оконную подсистему X Window System. Дистрибутив готов для конечной установки на пользовательское оборудование. Кроме ядра и, собственно, операционной системы дистрибутивы обычно содержат широкий набор приложений, таких как редакторы документов и таблиц, мультимедийные проигрыватели, системы для работы с базами данных и т.д. Существуют дистрибутивы, разрабатываемые как при коммерческой поддержке (Red Hat / Fedora, SLED / OpenSUSE, Ubuntu), так и исключительно усилиями добровольцев (Debian, Slackware, Gentoo, ArchLinux).

#### **3.2 Введение в командную строку в командную строку GNU Linux**

Работу ОС GNU Linux можно представить в виде функционирования множества взаимосвязанных процессов. При загрузке системы сначала запускается ядро, которое, в свою очередь, запускает оболочку ОС (от англ. shell «оболочка»). Взаимодействие пользователя с системой Linux (работа с данными и управление работающими в системе процессами) происходит в

интерактивном режиме посредством командного языка. оболочка операционной системы (или командная оболочка, интерпретатор команд) — интерпретирует (т.е. переводит на машинный язык) вводимые пользователем команды, запускает соответствующие программы (процессы), формирует и выводит ответные сообщения. Кроме того, на языке командной оболочки можно писать небольшие программы для выполнения ряда последовательных операций с файлами и содержащимися в них данными — сценарии (скрипты).

Из командных оболочек GNU Linux наиболее популярны `bash`, `csh`, `ksh`, `zsh`. Команда `echo $SHELL` позволяет проверить, какая оболочка используется. В качестве предустановленной командной оболочки GNU Linux используется одна из наиболее распространённых разновидностей командной оболочки — `bash` (Bourne again shell).

В GNU Linux доступ пользователя к командной оболочке обеспечивается через терминал (или консоль). Запуск терминала можно осуществить через главное меню «Приложения» «Стандартные» «Терминал (или Консоль)» или нажав `Ctrl + Alt + t`. Интерфейс командной оболочки очень прост. Обычно он состоит из приглашения командной строки (строки, оканчивающейся символом `$`), по которому пользователь вводит команды:

```
iivanova@dk4n31:~$ 7
```

Это приглашение командной оболочки, которое несёт в себе информацию об имени пользователя `iivanova`, имени компьютера `dk4n31` и текущем каталоге, в котором находится пользователь, в данном случае это домашний каталог пользователя, обозначенный как `~`).

Команды могут быть использованы с ключами (или опциями) — указаниями, модифицирующими поведение команды. Ключи обычно начинаются с символа `(-)` или `(--)` и часто состоят из одной буквы. Кроме ключей после команды могут быть использованы аргументы (параметры) — названия объектов, для которых нужно выполнить команду (например, имена файлов и каталогов). Например, для подробного просмотра содержимого каталога `documents` может быть использована команда `ls` с ключом `-l`:

```
iivanova@dk4n31:~$ ls -l documents
```

В данном случае `ls` – это имя команды, `l` – ключ, `documents` – аргумент. Команды, ключи и аргументы должны быть отделены друг от друга пробелом. Ввод команды завершается нажатием клавиши «Enter» , после чего команда передаётся оболочке на исполнение. Результатом выполнения команды могут являться сообщения о ходе выполнения команды или об ошибках. Появление приглашения командной строки говорит о том, что выполнение команды завершено.

Иногда в GNU Linux имена программ и команд слишком длинные, однако `bash` может завершать имена при их вводе в терминале. Нажав клавишу «Tab» , можно завершить имя команды, программы или каталога. Например, предположим, что нужно использовать программу `mcedit`. Для этого наберите в командной строке `mc`, затем нажмите один раз клавишу «Tab» . Если ничего не происходит, то это означает, что существует несколько возможных вариантов завершения команды. Нажав клавишу «Tab» ещё раз, можно получить список имён, начинающихся с `mc`:

```
iivanova@dk4n31:~$ mc
mc mcd mcedit mclasserase mcookie mcview
mcat mcdiff mcheck mcomp mcopy
iivanova@dk4n31:~$ mc
```

Более подробно о работе в операционной системе Linux см., например, в [13; 16].

## 4. Выполнение лабораторной работы

1. Я создала виртуальную машину с именем kinikulina

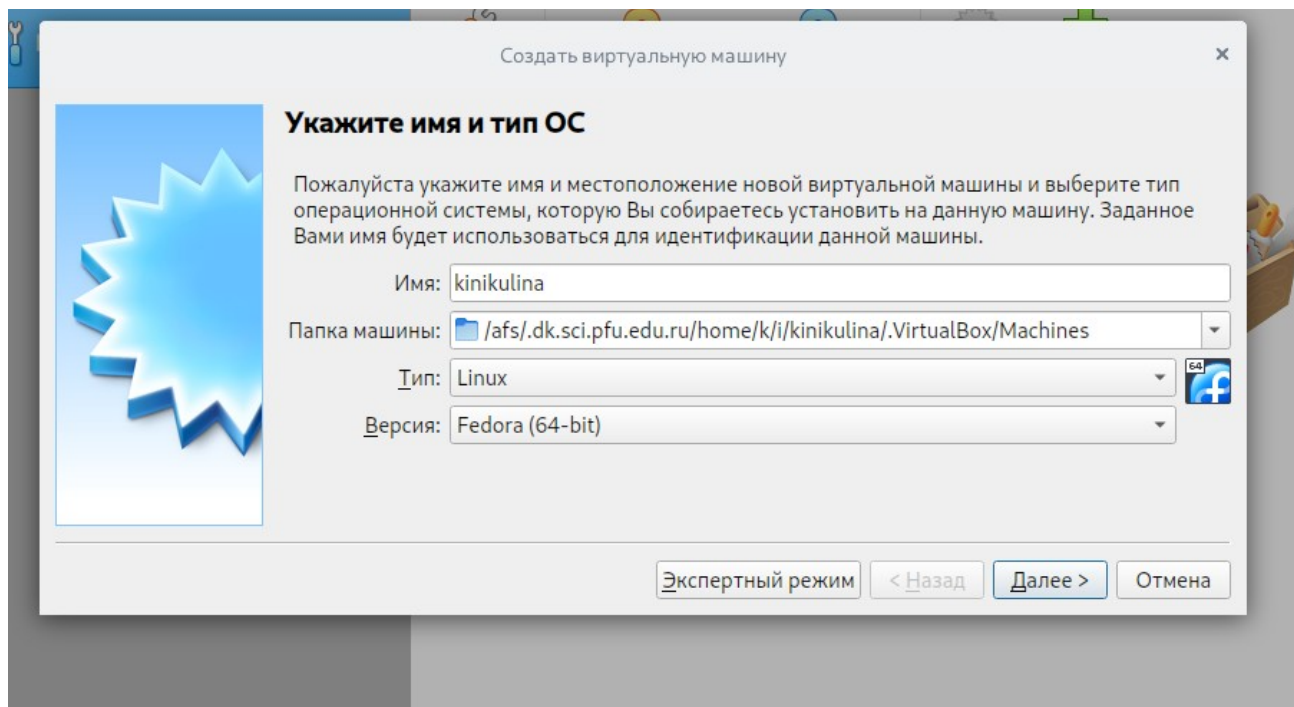


Рис. 1.1. Окно «Имя машины и тип ОС»

2. Я указала объем памяти равный 2048 мб.

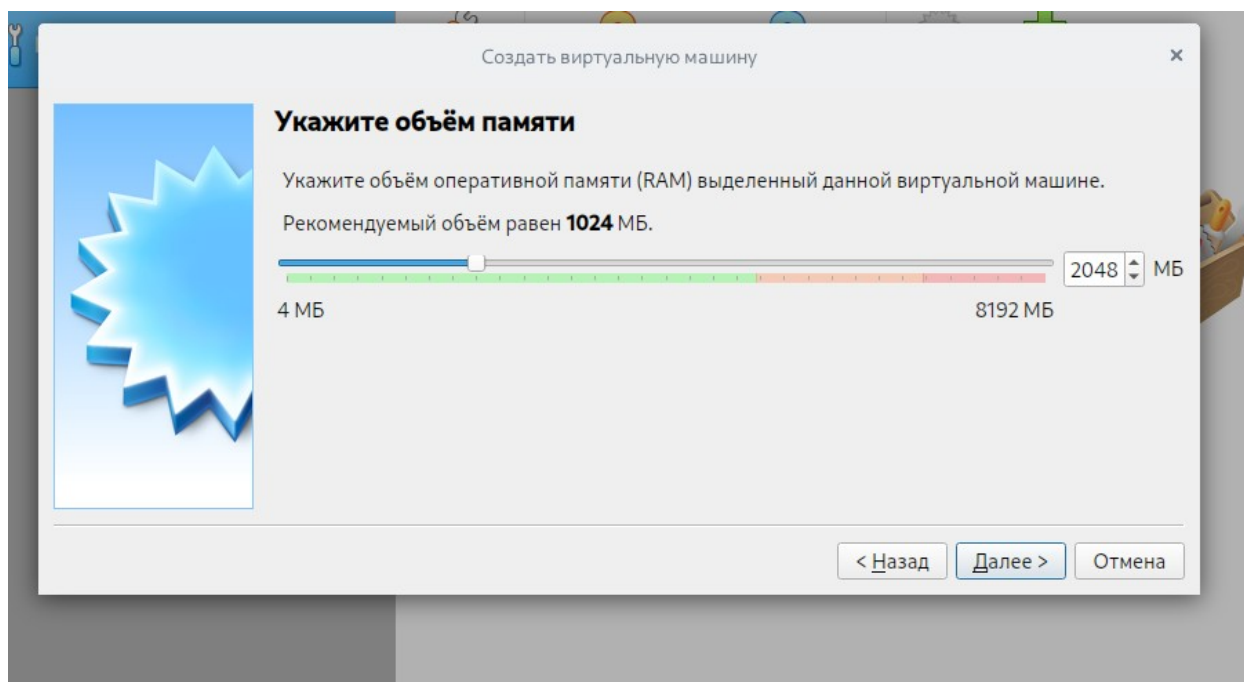


Рис. 1.2. «Размер основной памяти»



### 3. Я создала новый виртуальный жесткий диск.

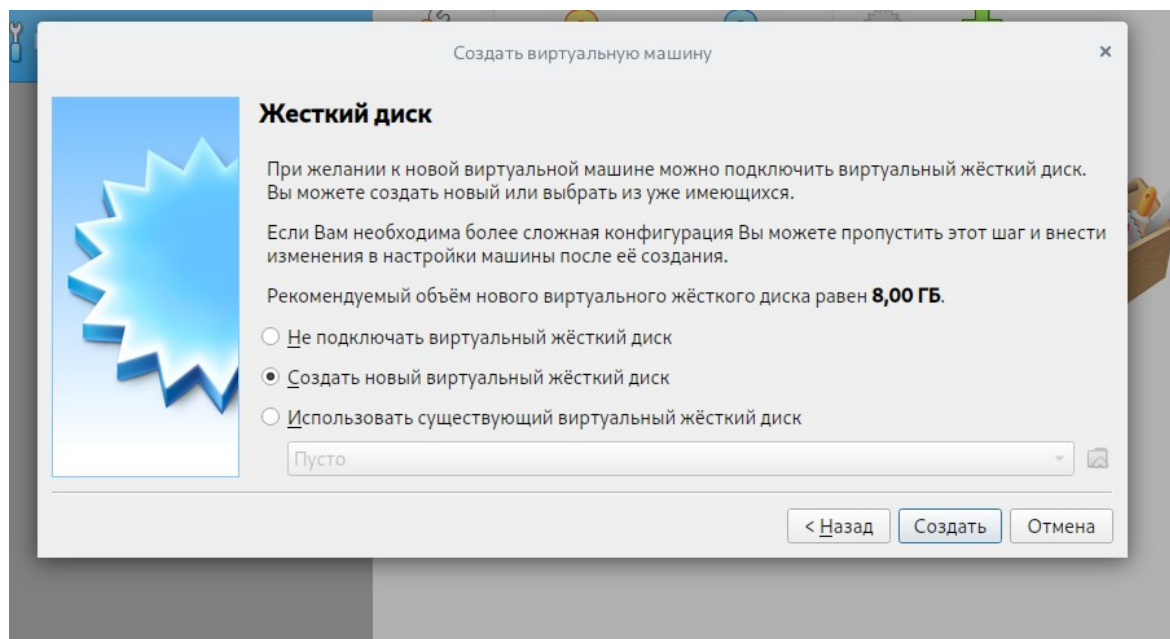


Рис. 1.3. «Окно подключения или создания жесткого диска на виртуальной машине»

### 4. Я указала тип VDI (Virtualbox Disk Image)

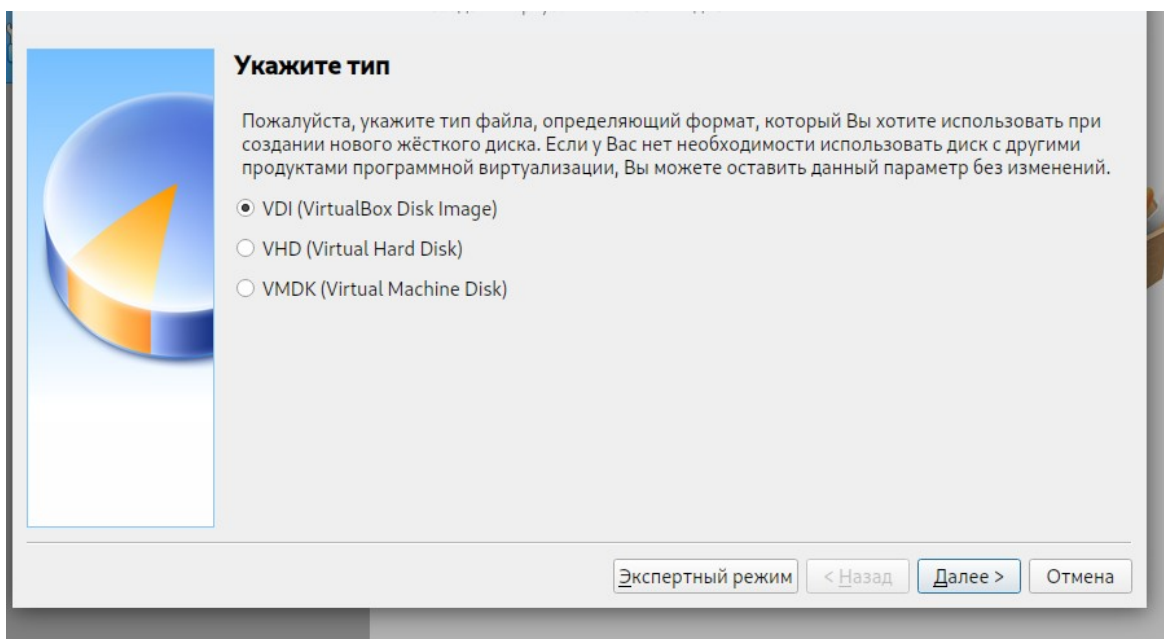


Рис. 1.4. «Окно определения типа подключения виртуального жесткого диска»

## 5. Я выбрала динамический виртуальный жесткий диск

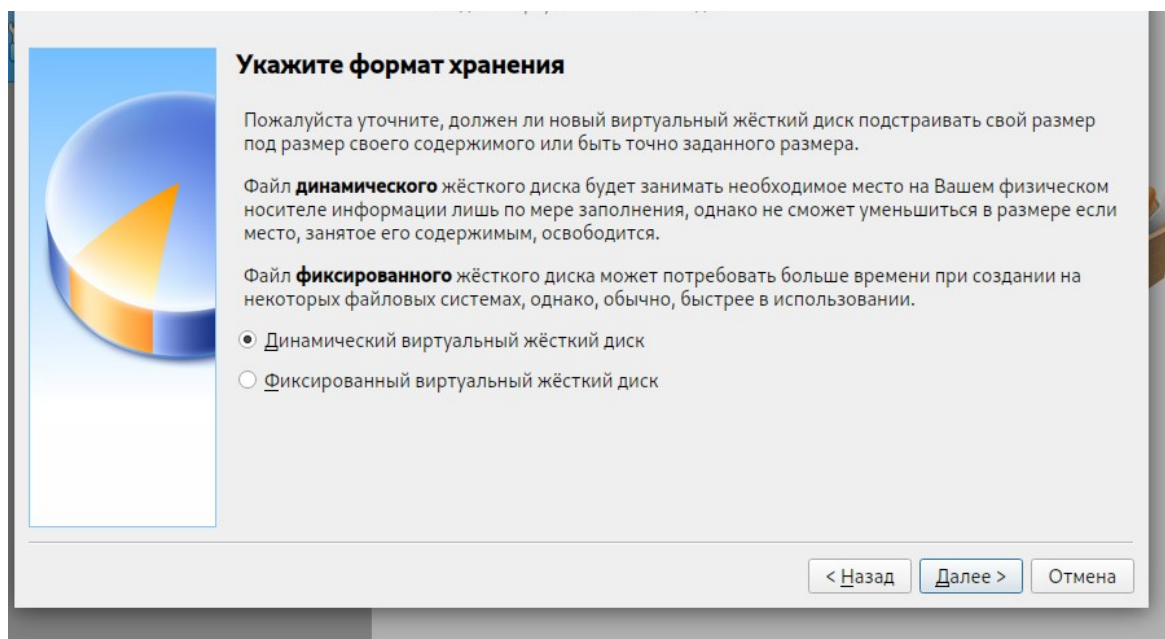


Рис. 1.5. «Окно определения формата виртуального жесткого диска»

## 6. Я выделила размер файла 80,4 Гб и указала имя виртуального жесткого диска

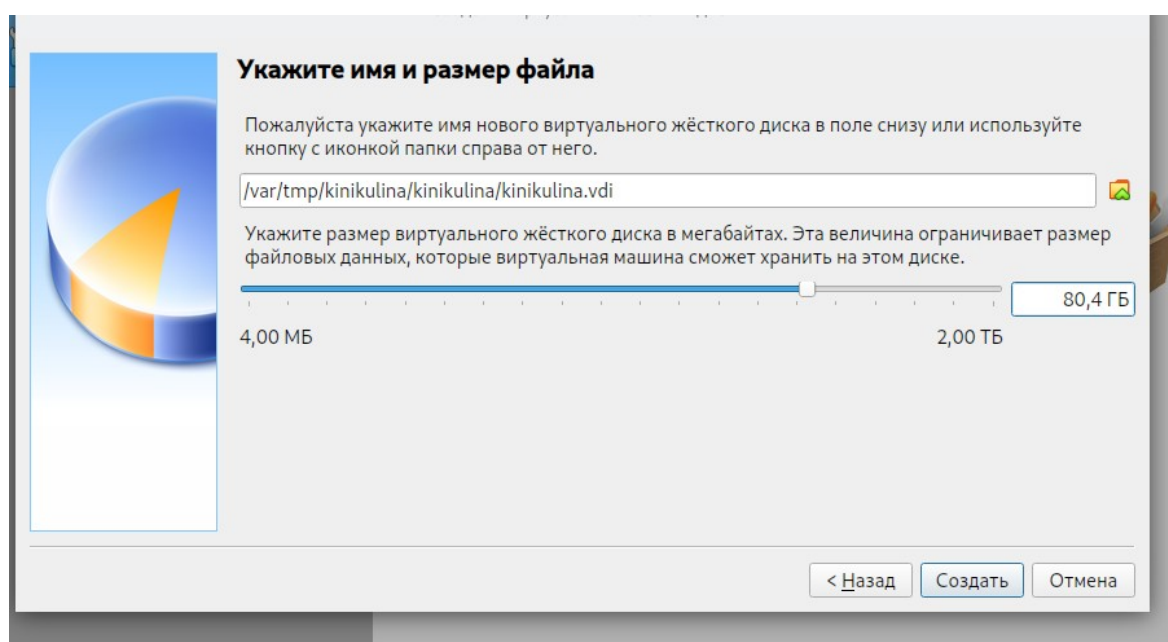


Рис. 1.6. «Окно определения размера виртуального динамического жесткого диска и его расположения»

## 7. Я настроила 128 МБ видеопамати

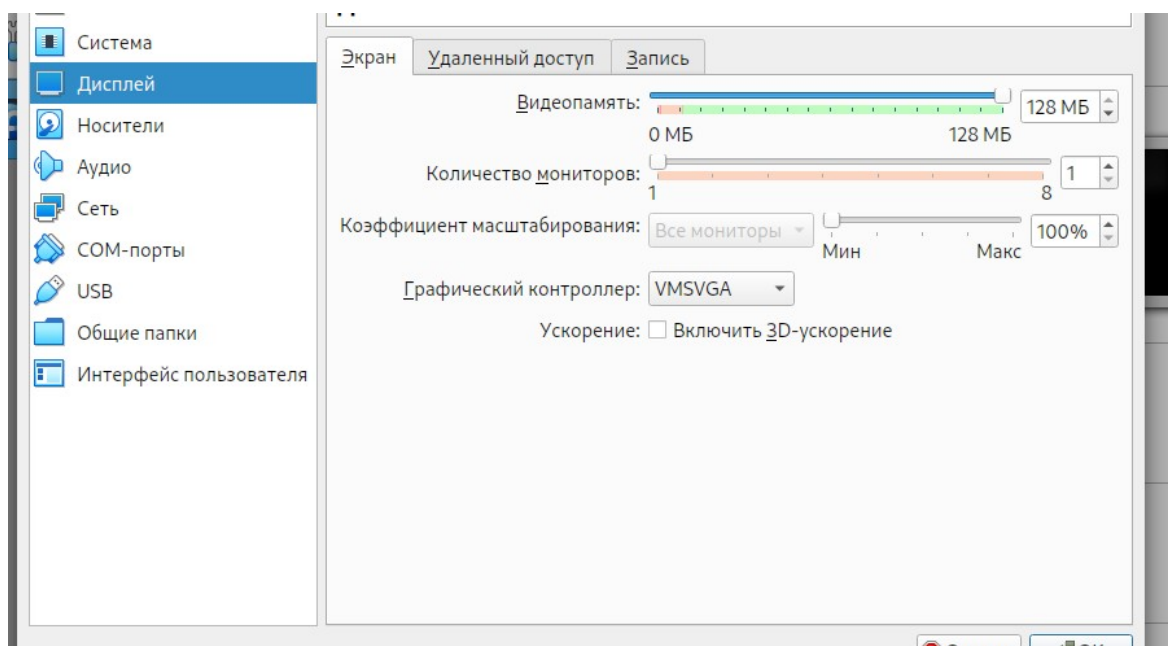


Рис. 1.7. «Настройки виртуальной машины»

## 8. К контроллеру Ide я подключила Fedora – Workstation – live

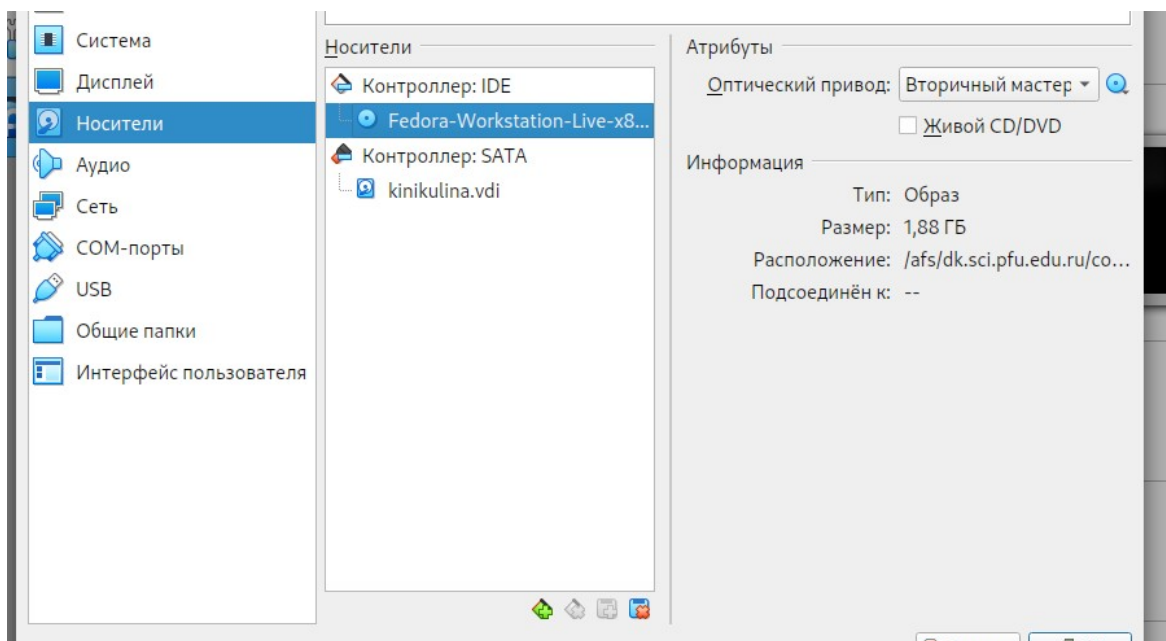


Рис. 1.8. «Окно «Носители» виртуальной машины: выбор образа оптического диска»

9. Я начала установку с системы выбора языка, выбрала время и регион (Европа, Москва). Также я установила место установки. В качестве локального диска я выбрала диск, который сделала при создании виртуальной машины.

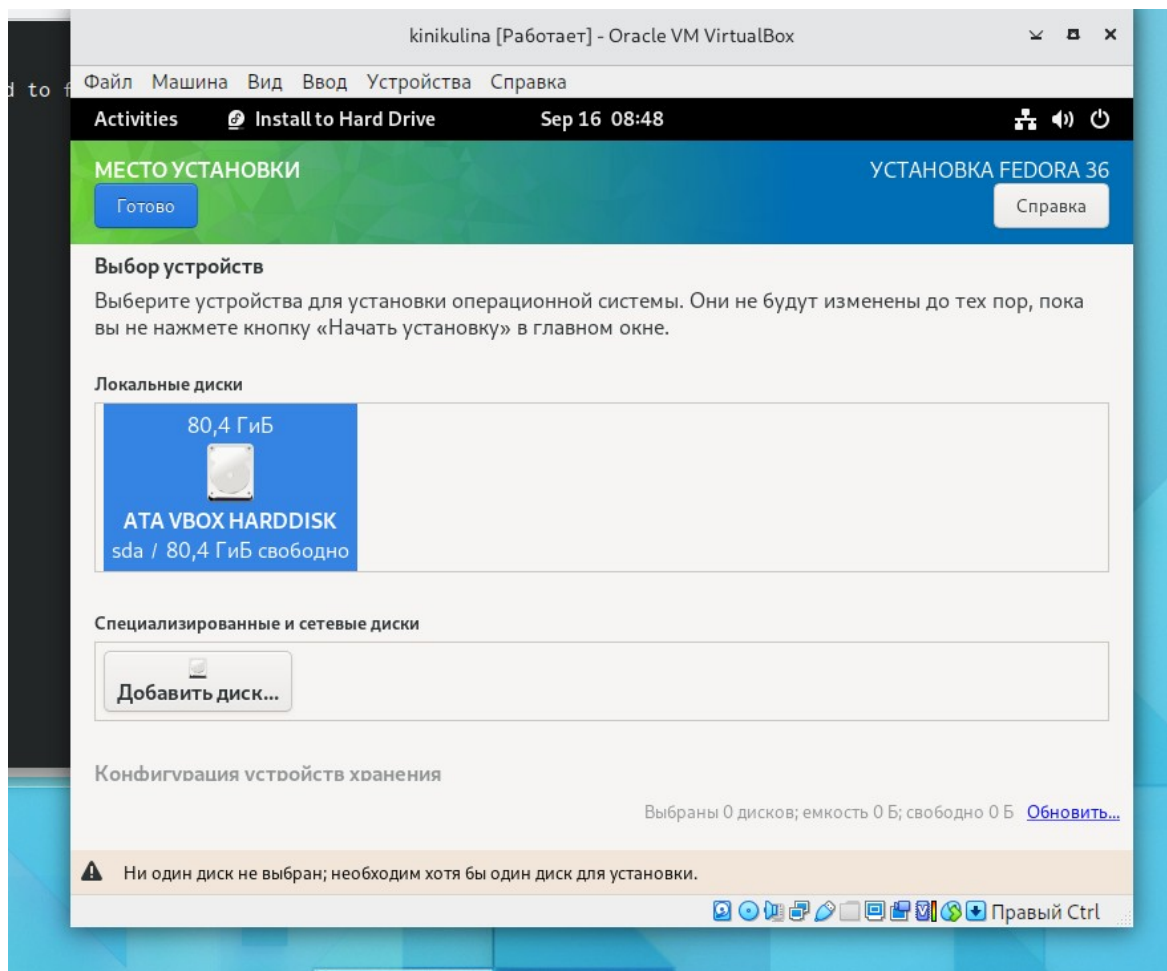


Рис. 2.1. «Окно выбора места установки»

10. Я начала загрузку операционной системы

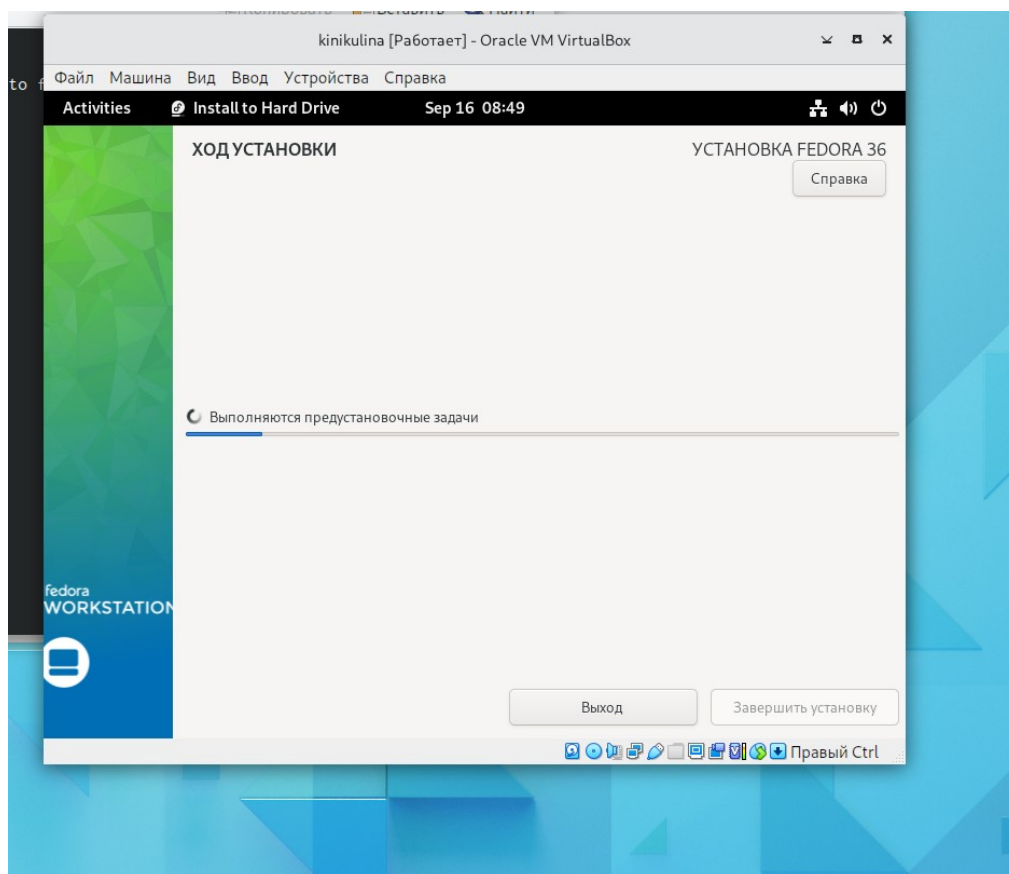


Рис. 2.2. «Окно установки»

11. Я установила имя пользователя и пароль пользователя для безопасности

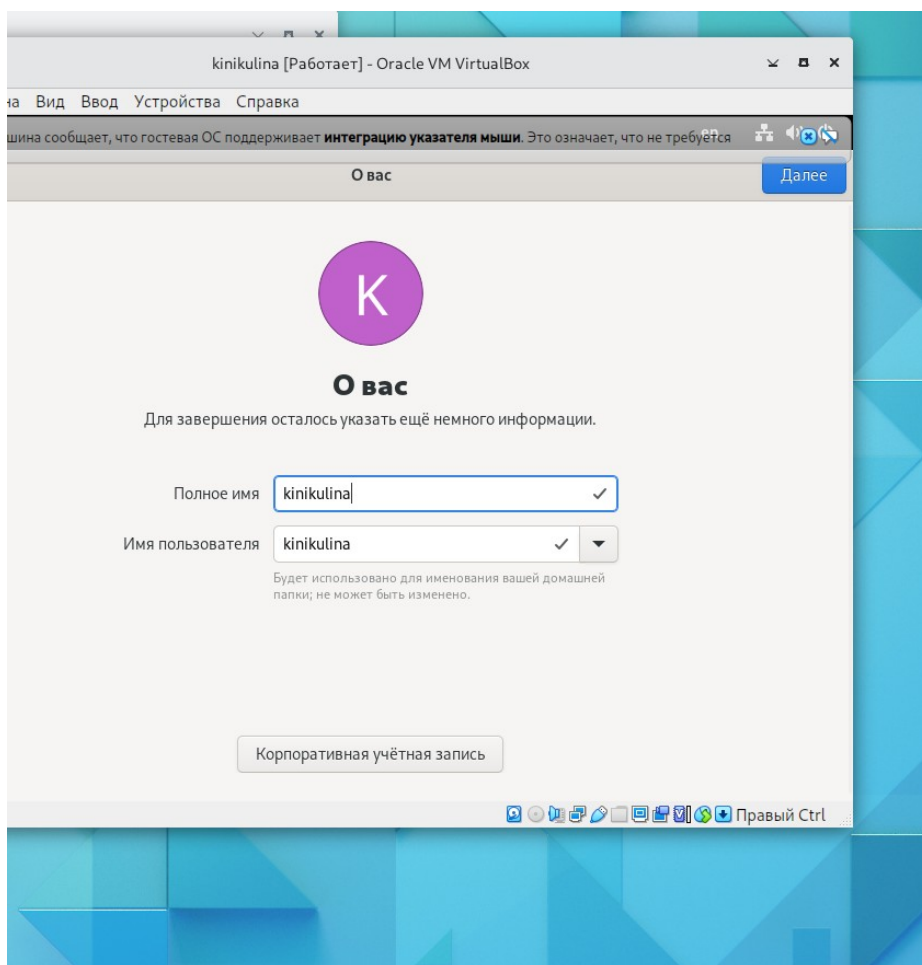
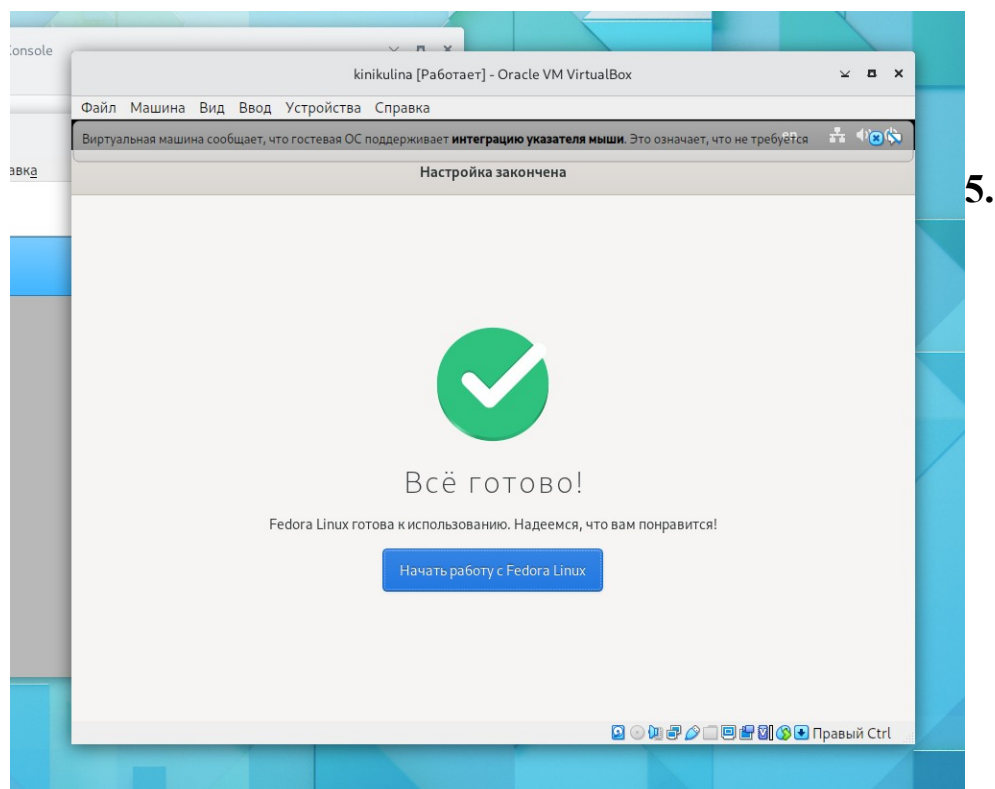


Рис. 2.3. «Окно установки имени пользователя»



5.



## Самостоятельная работа

1. С помощью терминала я установила Midnight Commander (mc), проверила наличие системы Git , а также я проверил наличие системы Nasm (Netwide Assembler)

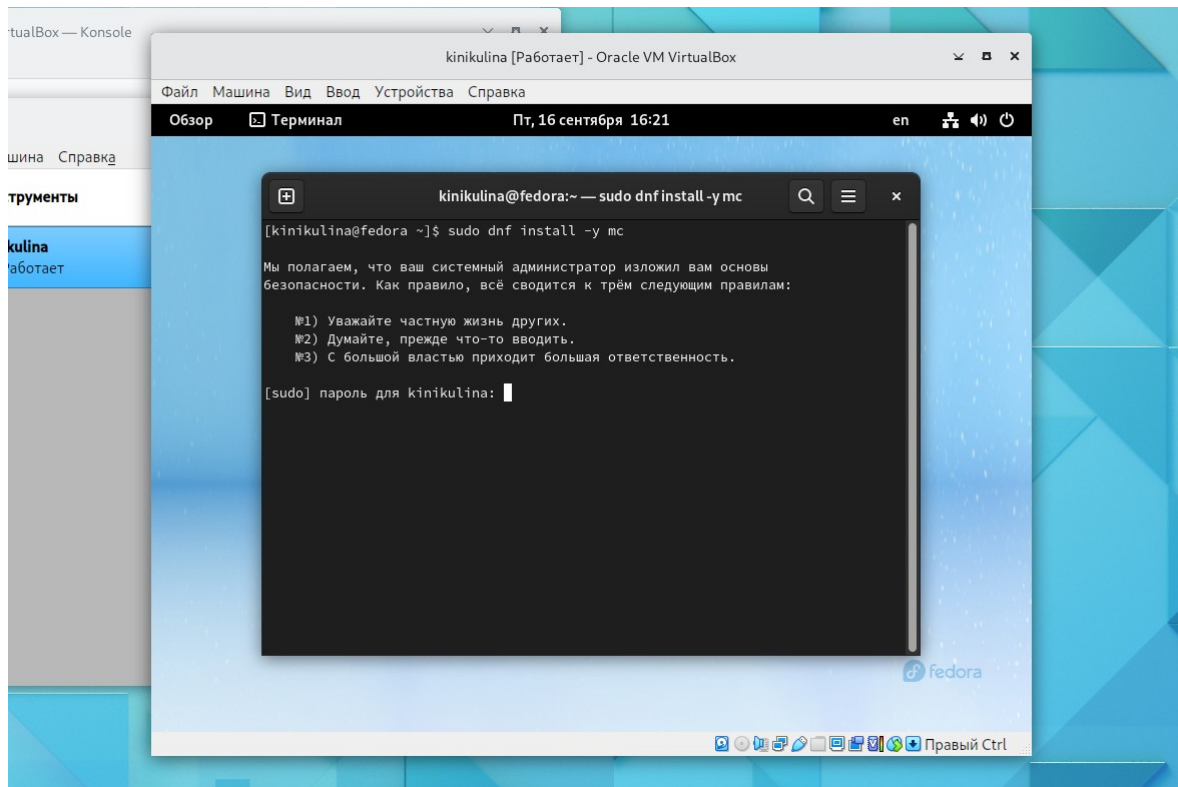
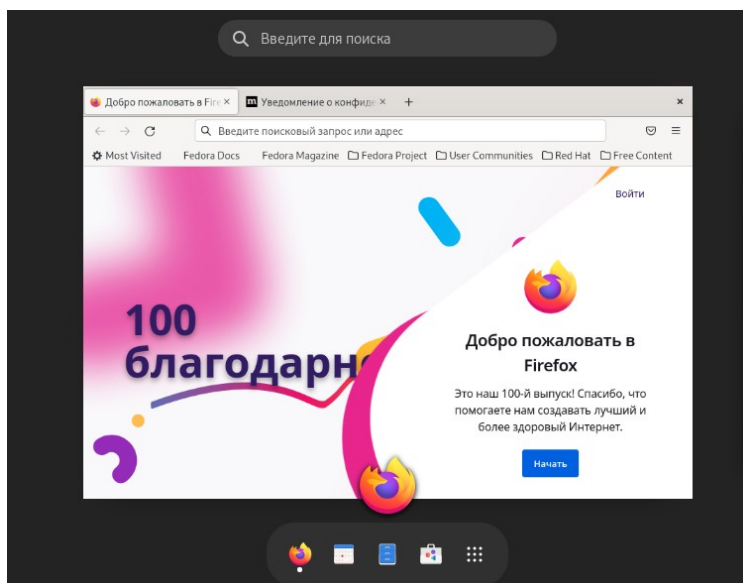


Рис. 3.1. «Терминал. Установка mc

2. Нашла в меню приложений и запустила браузер, текстовый процессор и текстовый редактор.



## **6. Вывод**

Я приобрела практические навыки установки операционной системы на виртуальную машину и настроила минимально необходимые для дальнейшей работы сервисы.