

# РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук  
Кафедра прикладной информатики и теории вероятностей

## ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

дисциплина:    Архитектура компьютера

Студент: Ким Ангелина

Группа: НММбд-01-22

МОСКВА

2022 г.

## Содержание

1. Цель работы .....	4
2. Задание .....	5
3. Теоретическое введение .....	6
3.1 Введение в GNU Linux.....	6
3.2 Введение в командную строку GNU Linux.....	6
4. Выполнение лабораторной работы .....	9
4.1 Настройка VirtualBox.....	9
4.2 Запуск виртуальной машины и установка системы .....	12
4.3 Выполнение заданий для самостоятельной работы .....	13
4.3.1 1 задание .....	13
4.3.2 2 задание .....	13
4.3.3 3 задание .....	15
4.3.4 4 задание .....	15
5. Выводы .....	18

## Список иллюстраций

Рис 4.1.1 Установка виртуальной машины VirtualBox .....	9
Рис 4.1.2 Окно «Имя машины и тип ОС» .....	10
Рис 4.1.3 Окно «Носители» виртуальной машины: выбор образа оптического диска .....	11
Рис. 4.1.4 Виртуальная машина главный экран .....	11
Рис 4.3.1.1 Рабочий стол .....	13
Рис. 4.3.2.1 Открытие браузера .....	14
Рис. 4.3.2.2 Открытие текстового процессора .....	14
Рис. 4.3.2.3 Открытие текстового редактора .....	15
Рис. 4.3.3 Запускаем терминал .....	15
Рис. 4.3.4.1 Установка Midnight Commander .....	16
Рис. 4.3.4.2 Установка Midnight Commander .....	16
Рис. 4.3.4.2 Установка Git .....	17
Рис. 4.3.4.3 Установка Nasm (Netwide Assembler) .....	17

## **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

1. Запустить установленную в VirtualBox ОС
2. Найдите в меню приложений и запустите браузер (например Firefox), текстовый процессор (например, LibreOffice Writer) и любой текстовый редактор.
3. Запустите терминал (консоль).
4. Установите основное программное обеспечение необходимое для дальнейшей работы.

4.1. Midnight Commander (mc) – это файловый менеджер с терминальным интерфейсом.

Команда для установки mc через терминал `user@dk4n31:~$ sudo dnf install -y mc`

Команда для запуска mc `user@dk4n31:~$ mc`

4.2. Git – система управления версиями.

Команда для установки git через терминал `user@dk4n31:~$ sudo dnf install -y git`

4.3. Nasm (Netwide Assembler) – свободный ассемблер для архитектуры Intel x86.

Команда для установки nasm через терминал `user@dk4n31:~$ sudo dnf install -y nasm`

### **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 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:~$
```

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:

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



## 4. Выполнение лабораторной работы

### 4.1 Настройка VirtualBox

Для выполнения лабораторной работы мне необходимо установить на свое устройство виртуальную машину VirtualBox (<https://www.virtualbox.org/>) операционной системы Linux (дистрибутив Fedora) (рис. 4.1.1).

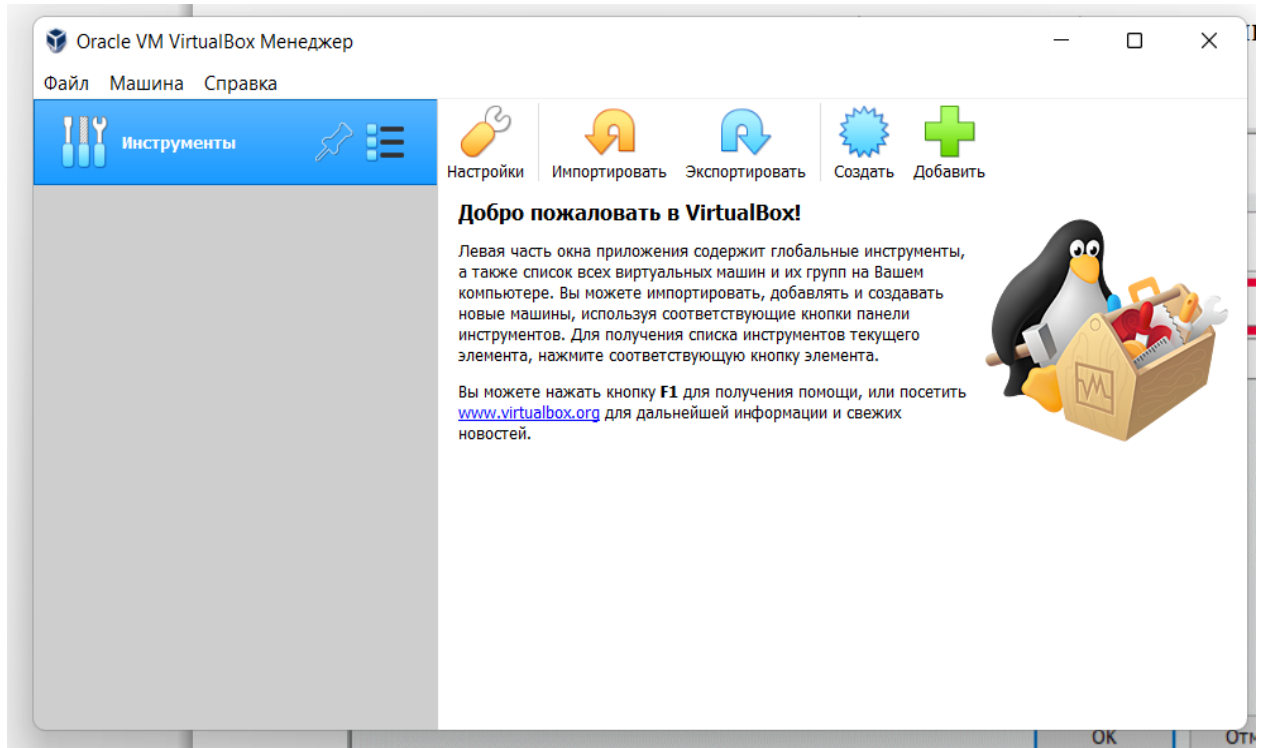


Рис 4.1.1 Установка виртуальной машины VirtualBox

Далее нужно настроить виртуальную машину. Для этого, в первую очередь нам нужно поменять имя пользователя. Для этого в настройках VirtualBox нужно перейти во вкладку «Общие» и вместо «user\_name» написать «apkim».

Следующим шагом нужно создать новую виртуальную машину. Для этого в VirtualBox нужно выбрать «Машина» - «Создать».

#### Добро пожаловать в VirtualBox!

Левая часть окна приложения содержит глобальные инструменты, а также список всех виртуальных машин и их групп на Вашем компьютере. Вы можете импортировать, добавлять и создавать новые машины, используя соответствующие кнопки панели инструментов. Для получения списка инструментов текущего элемента, нажмите соответствующую кнопку элемента.

Вы можете нажать кнопку **F1** для получения помощи, или посетить [www.virtualbox.org](http://www.virtualbox.org) для дальнейшей информации и свежих новостей.

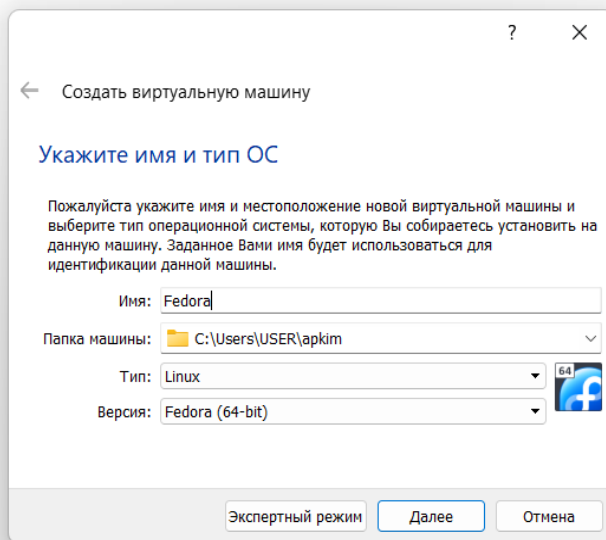


Рис 4.1.2 Окно «Имя машины и тип ОС»

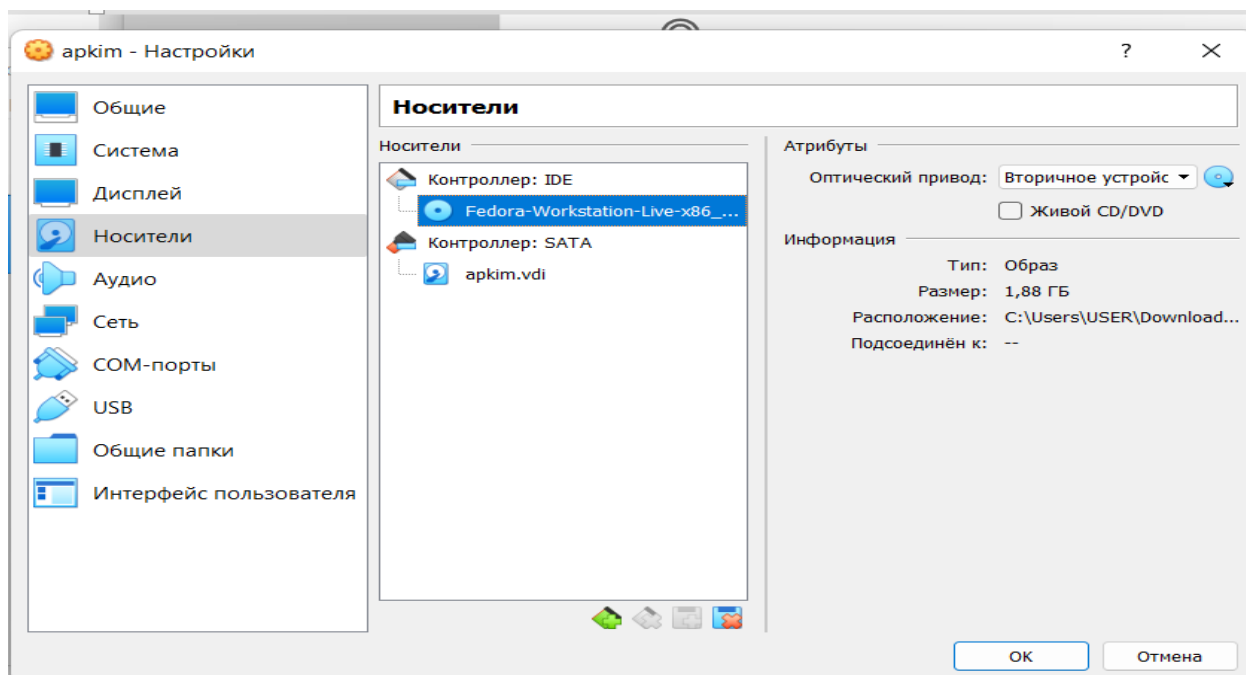


Рис 4.1.3 Окно «Носители» виртуальной машины: выбор образа оптического диска

Необходимо указать имя виртуальной машины, тип операционной системы – Linux, Fedora (рис. 4.1.2). Размер основной памяти виртуальной машины – от 2048 МБ. Задаем конфигурацию жёсткого диска – загрузочный, VDI (VirtualBox Disk Image), динамический виртуальный диск. Размер диска задаем 80 ГБ. В настройках виртуальной машины во вкладке «Дисплей» - «Экран» следует увеличить доступный объем видеопамати до 128 МБ. В настройках виртуальной машины во вкладке «Носители» добавим новый привод оптических дисков и выберем образ (рис. 4.1.3). Виртуальная машина готова к работе (рис. 4.1.4)

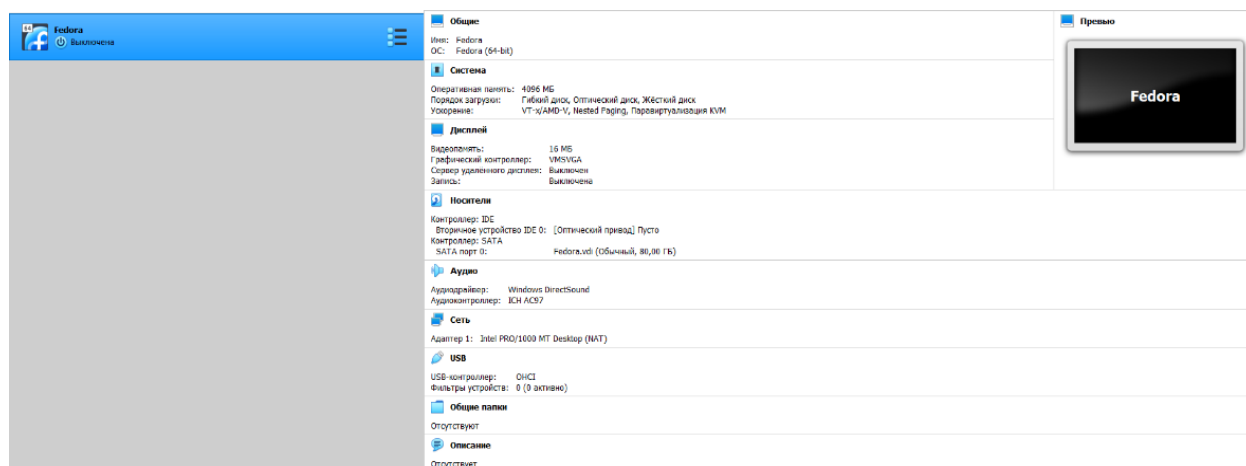


Рис. 4.1.4 Виртуальная машина главный экран

## 4.2 Запуск виртуальной машины и установка системы

Запускаем виртуальную машину «Машина» - «Запустить».

После загрузки с виртуального оптического диска мы видим окно с двумя вариантами. Выбираем вариант «Install to Hard Drive».

При необходимости корректируем часовой пояс, раскладку клавиатуры. Место установки ОС оставляем без изменения. Следует последовательно проверить настройки даты и времени, клавиатуры, настройки сети и места установки.

После установки всех настроек можно продолжить установку. В настройках диска выбираем «Обычный раздел». После этого шага нажимаем на кнопку «Начать установку». Установка начнется, а нам будет предложено задать пароль для пользователя с нашим логином. Имя пользователя создаем в соответствии с соглашением об именовании (имя пользователя совпадает с логином студента в дисплейном классе).

После окончания установки, следует закрыть окно установщика и выключить систему.

После того, как виртуальная машина отключится, следует изъять образ диска из дисковод. Не удаляем при этом сам дисковод.

Нажимаем на значок диска и выбираем пункт изъять. После извлечения в дисководе должно быть пусто.

Виртуальная машина готова к работе и ее можно запустить «Машина» - «Запустить».

## 4.3 Выполнение заданий для самостоятельной работы

### 4.3.1 1 задание

Запустить установленную в VirtualBox ОС (рис. 4.3.1.1)

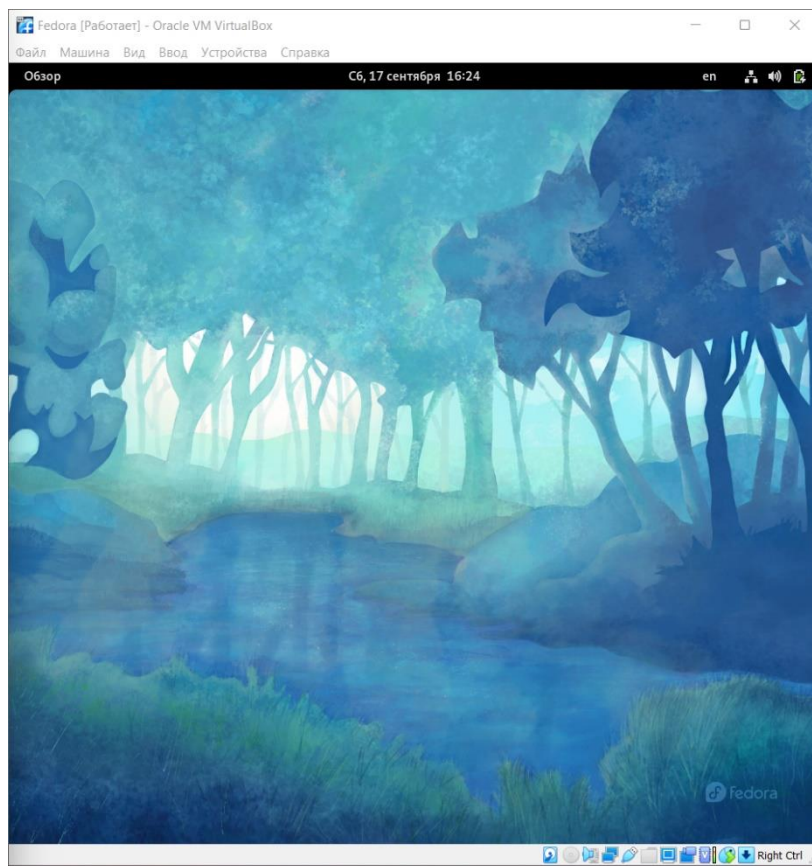


Рис 4.3.1.1 Рабочий стол

### 4.3.2 2 задание

Находим в меню приложения и запускаем браузер (например, Firefox)  
(рис.4.3.2.1)

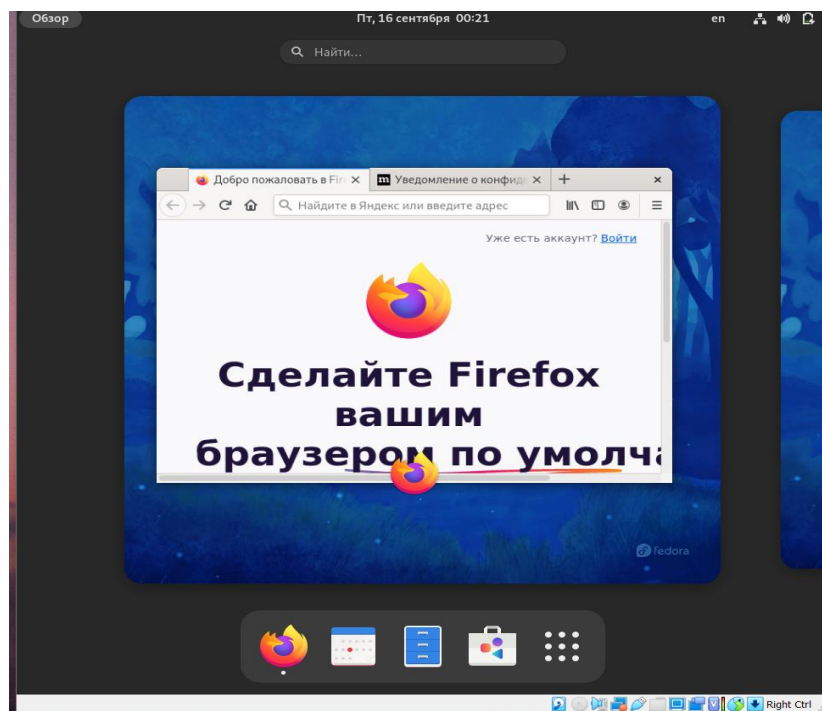


Рис. 4.3.2.1 Открытие браузера

Запускаем текстовый процессор (например, LibreOffice Writer) (рис. 4.3.2.2)

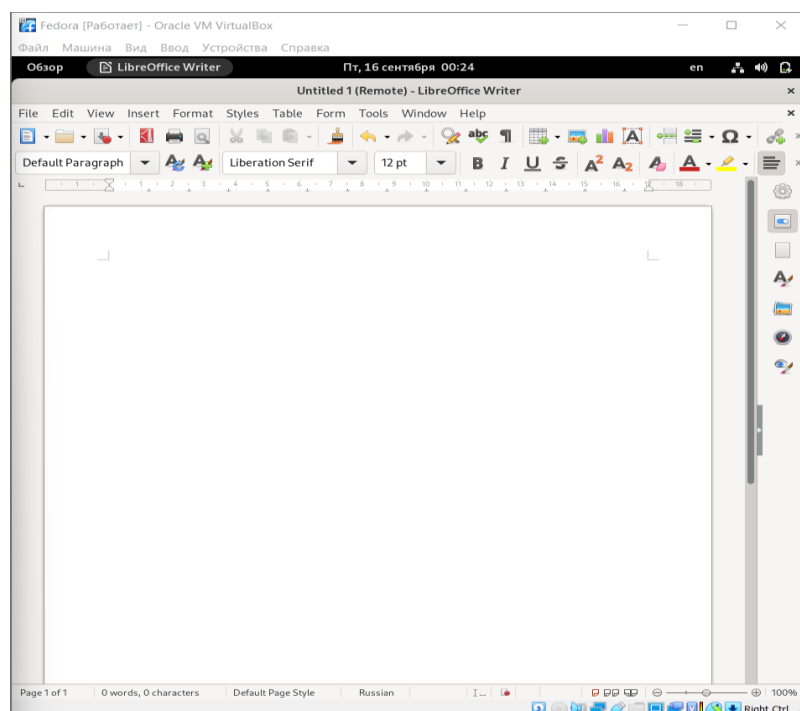


Рис. 4.3.2.2 Открытие текстового процессора

Запускаем текстовый редактор (рис. 4.3.2.3).

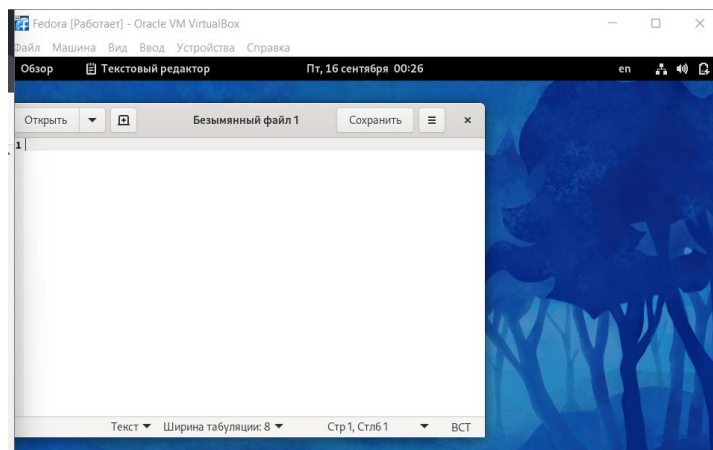


Рис. 4.3.2.3 Открытие текстового редактора

### 4.3.3 3 задание

Запускаем терминал (консоль) (рис.4.3.3).

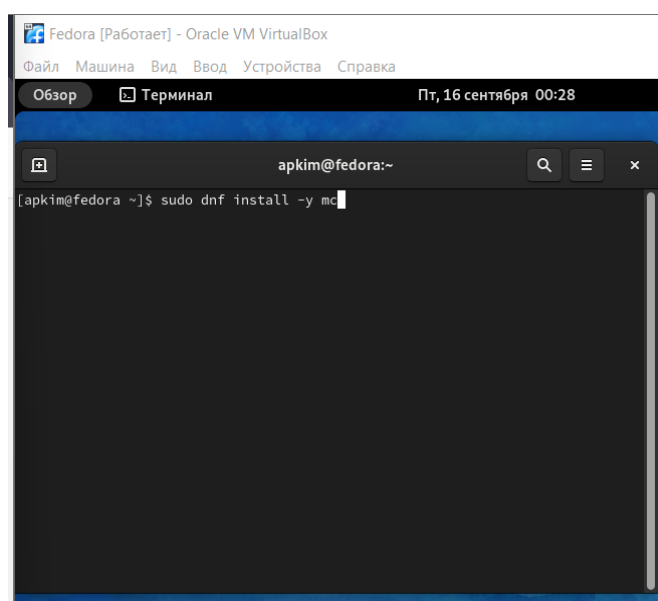


Рис. 4.3.3 Запускаем терминал

### 4.3.4 4 задание

Устанавливаем основное программное обеспечение необходимое для дальнейшей работы.

#### 4.3.4.1 Midnight Commander

Midnight Commander (mc) — это файловый менеджер с терминальным интерфейсом. Команда для установки mc через терминал (рис. 4.3.4.1, рис. 4.3.4.2)

```
user@dk4n31:~$ sudo dnf install -y mc
```

Команда для запуска mc

user@dk4n31:~\$ mc

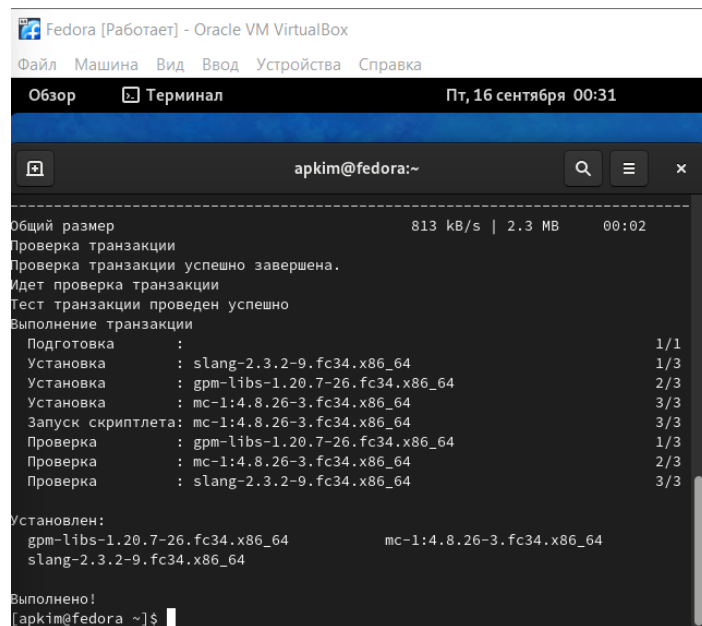


Рис. 4.3.4.1 Установка Midnight Commander

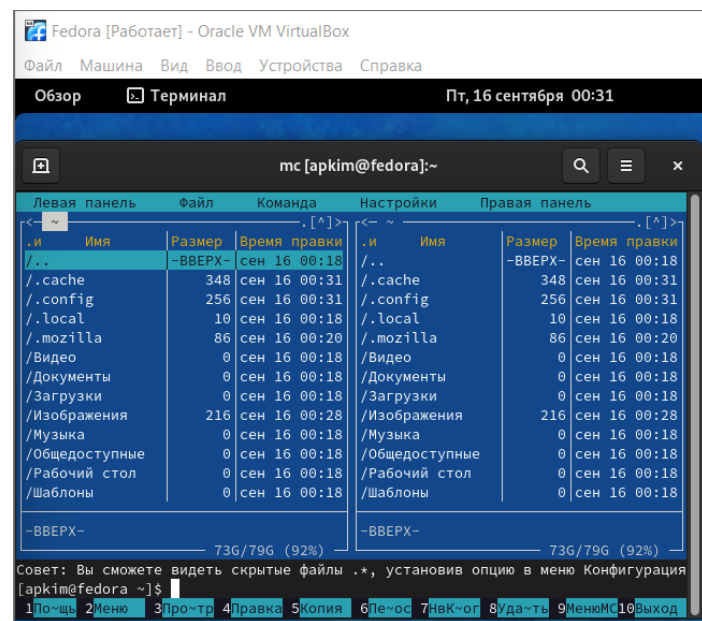


Рис. 4.3.4.2 Установка Midnight Commander

#### 4.3.4.2 Git

Git – система управления версиями.

Команда для установки git через терминал (рис. 4.3.1.4.2)



user@dk4n31:~\$ sudo dnf install -y git

```
[apkim@fedora ~]$ sudo dnf install -y git
Последняя проверка окончания срока действия метаданных: 0:03:38 назад, Пт 16 с
ен 2022 00:30:30.
Пакет git-2.31.1-1.fc34.x86_64 уже установлен.
Зависимости разрешены.
Отсутствуют действия для выполнения
Выполнено!
[apkim@fedora ~]$
```

Рис. 4.3.4.2 Установка Git

#### 4.3.4.3 Nasm (Netwide Assembler)

Nasm (Netwide Assembler) – свободный ассемблер для архитектуры Intel x86.

Команда для установки nasm через терминал (рис. 4.3.1.4.3)

user@dk4n31:~\$ sudo dnf install -y nasm

```
=====
Пакет      Архитектура  Версия      Репозиторий  Размер
=====
Установка:
nasm       x86_64       2.15.05-1.fc34  updates      432 k
=====
Результат транзакции
=====
Установка 1 Пакет

Объем загрузки: 432 k
Объем изменений: 2.9 М
Загрузка пакетов:
nasm-2.15.05-1.fc34.x86_64.rpm          902 kB/s | 432 kB    00:00
-----
Общий размер                          473 kB/s | 432 kB    00:00
Проверка транзакции
Проверка транзакции успешно завершена.
Идет проверка транзакции
Тест транзакции проведен успешно
Выполнение транзакции
Подготовка      :                               1/1
Установка       : nasm-2.15.05-1.fc34.x86_64  1/1
Запуск скрипта  : nasm-2.15.05-1.fc34.x86_64  1/1
Проверка        : nasm-2.15.05-1.fc34.x86_64  1/1

Установлен:
nasm-2.15.05-1.fc34.x86_64

Выполнено!
[apkim@fedora ~]$
```

Рис. 4.3.4.3 Установка Nasm (Netwide Assembler)

## **5. Выводы**

В ходе выполнения лабораторной работы я приобрела практические навыки установки операционной системы Linux на виртуальную машину VirtualBox, а также настройки минимально необходимых для дальнейшей работы сервисов (Midnight Commander, Git, Nasm).