

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

дисциплина: Операционные системы

Студент: Лебедева Алёна

Группа: НБИбд-02-21

Ст. билет №: 1032212267

Москва

2022 г.

Цель работы:

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

Ход работы:

- 1) Скачиваем и устанавливаем VirtualBox, которая необходима для запуска виртуальных машин (рис. 1, рис. 2). Устанавливаем скаченный файл.

Welcome to VirtualBox.org!

VirtualBox is a powerful x86 and AMD64/Intel64 [virtualization](#) product for enterprise as well as home use. Not only is VirtualBox an extremely feature rich, high performance product for enterprise customers, it is also the only professional solution that is freely available as Open Source Software under the terms of the GNU General Public License (GPL) version 2. See "[About VirtualBox](#)" for an introduction.

Presently, VirtualBox runs on Windows, Linux, Macintosh, and Solaris hosts and supports a large number of [guest operating systems](#) including but not limited to Windows (NT 4.0, 2000, XP, Server 2003, Vista, Windows 7, Windows 8, Windows 10), DOS/Windows 3.x, Linux (2.4, 2.6, 3.x and 4.x), Solaris and OpenSolaris, OS/2, and OpenBSD.

VirtualBox is being actively developed with frequent releases and has an ever growing list of features, supported guest operating systems and platforms it runs on. VirtualBox is a community effort backed by a dedicated company: everyone is encouraged to contribute while Oracle ensures the product always meets professional quality criteria.



Hot picks:

- Pre-built virtual machines for developers at [Oracle Tech Network](#)
- **Hyperbox** Open-source Virtual Infrastructure Manager [project site](#)
- **phpVirtualBox** AJAX web interface [project site](#)

Рис. 1

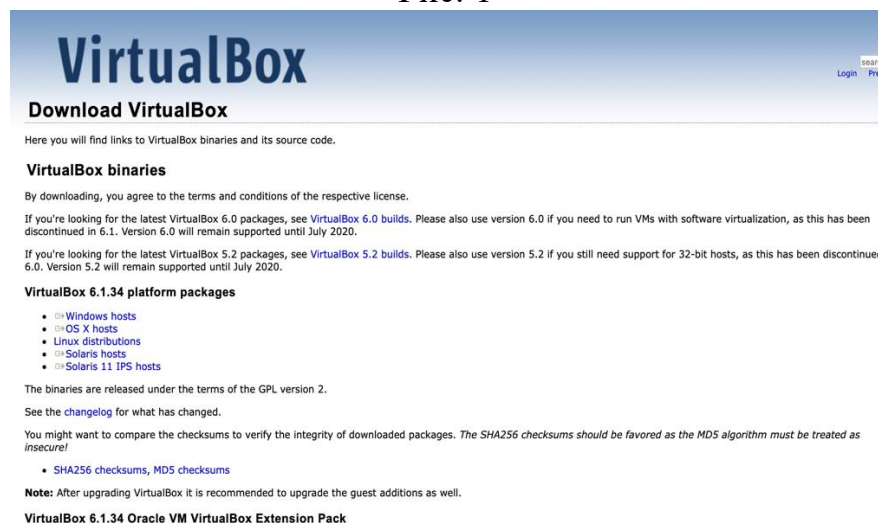


Рис. 2

2) Скачиваем дистрибутив Linux Fedora-35 (рис. 3)

Загрузить Fedora 35 Workstation.

Мы рады, что вы решили попробовать Fedora Workstation. Мы знаем — вам понравится.

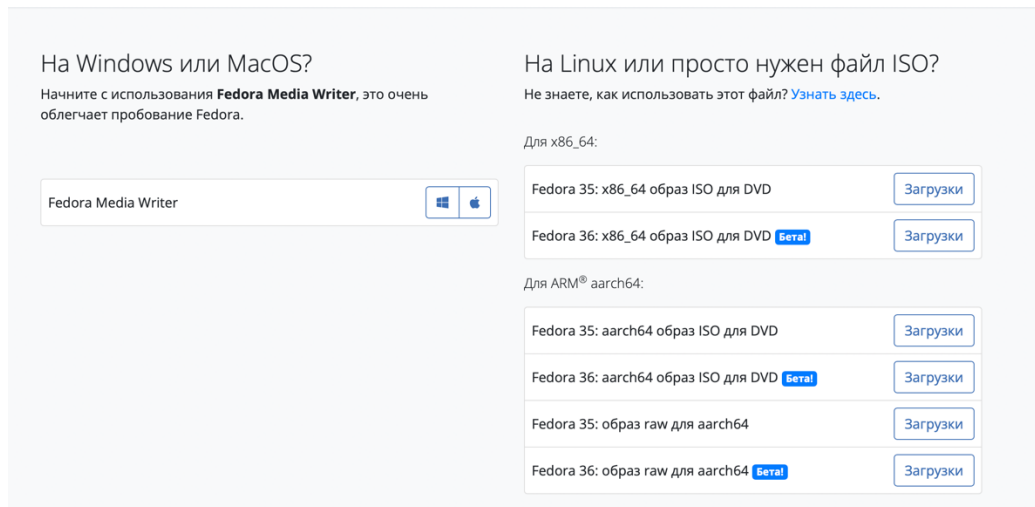


Рис. 3

3) Запускаем виртуальную машину и проверяем месторасположения каталога для виртуальных машин. Затем переходим к созданию новой виртуальной машины. Для этого в VirtualBox мы выбираем Машина – Создать (рис. 4)

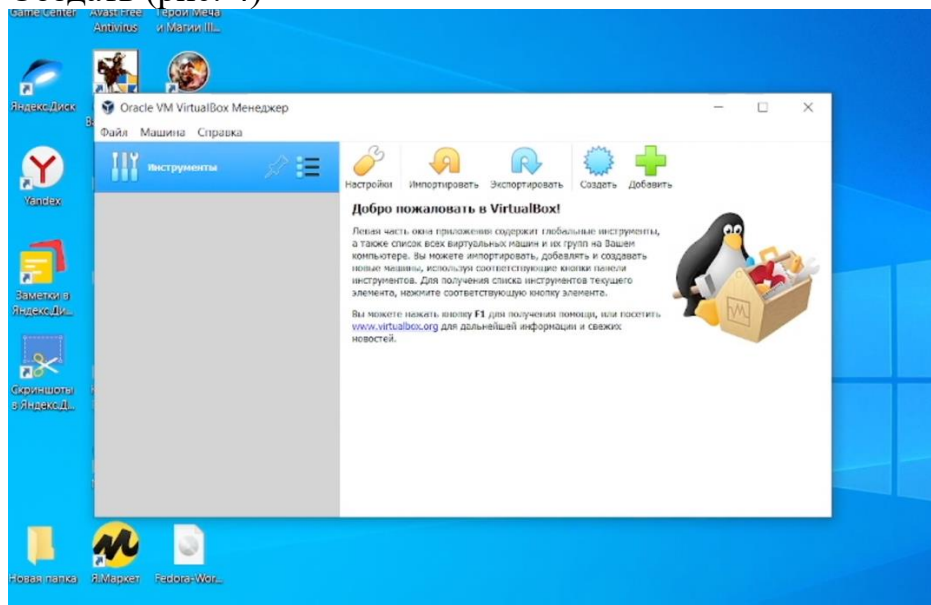


Рис. 4

4) Указываем имя виртуальной машины и тип ОС (рис. 5)

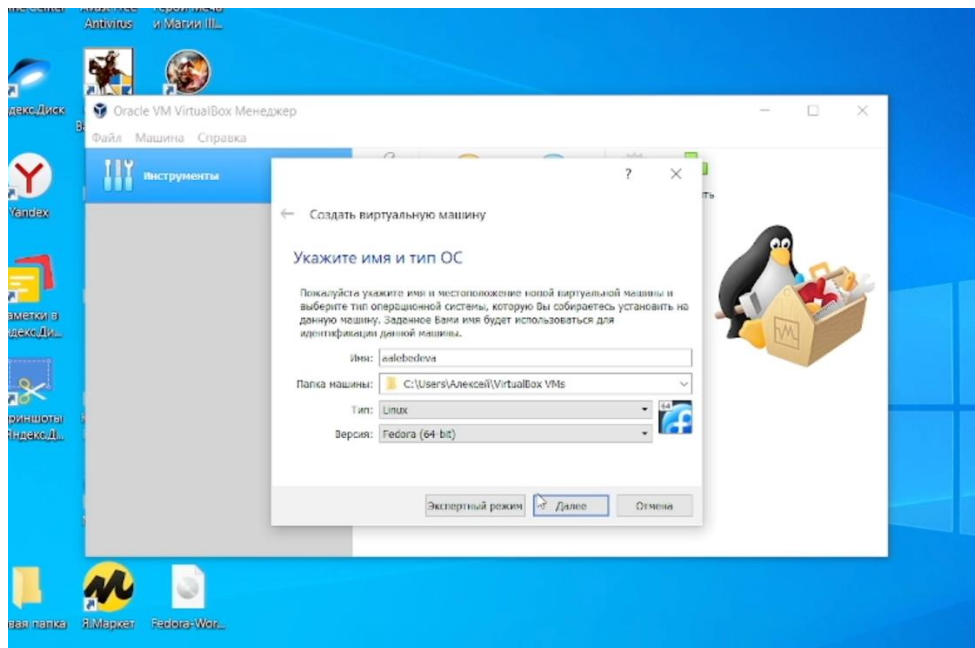


Рис. 5

- 5) Указываем размер основной памяти виртуальной машины – 2048 МБ (рис. 6)

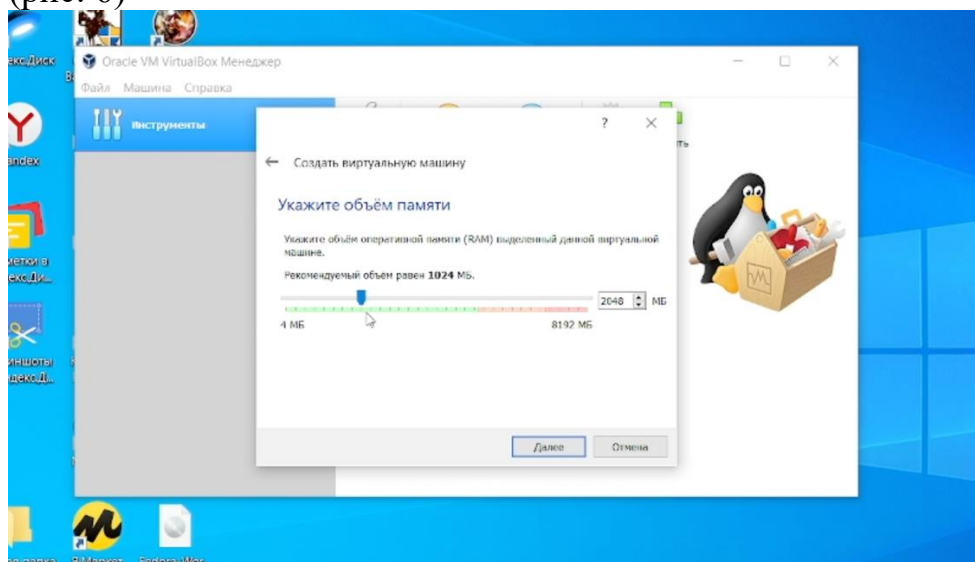


Рис. 6

- 6) Задаем конфигурацию жесткого диска VDI, динамический виртуальный жесткий диск (рис. 7-8)

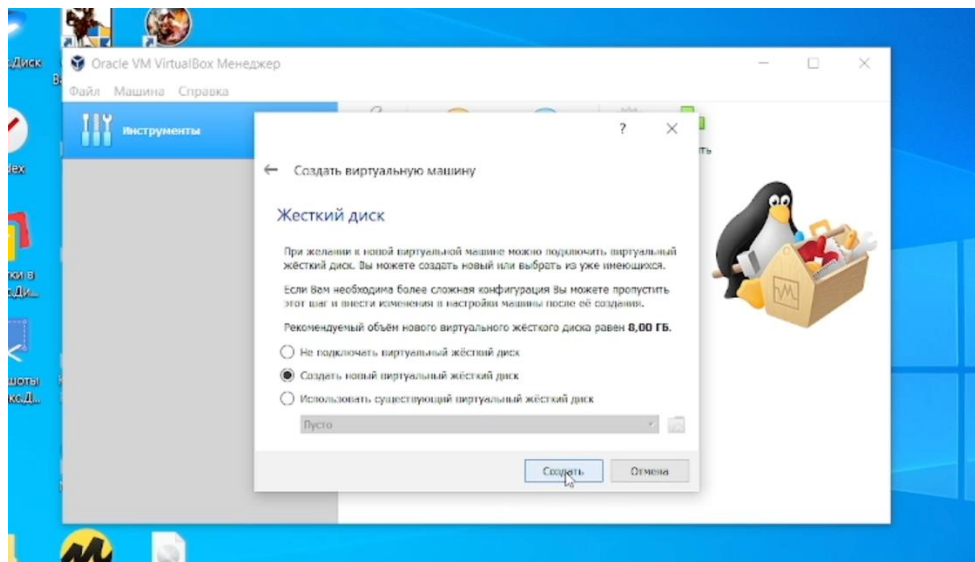


Рис. 7

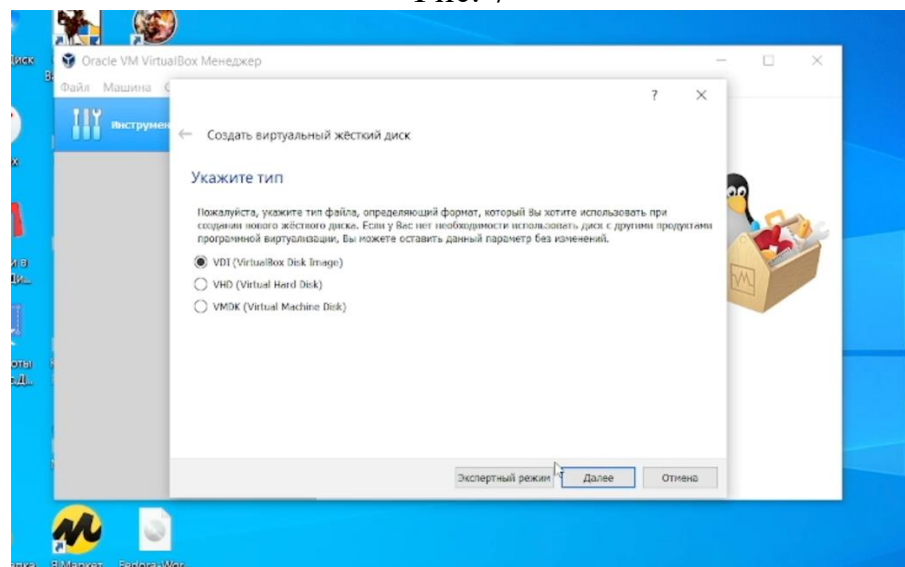


Рис. 8

7) Задаем размер диска 80 ГБ и его расположение (рис. 9-10)

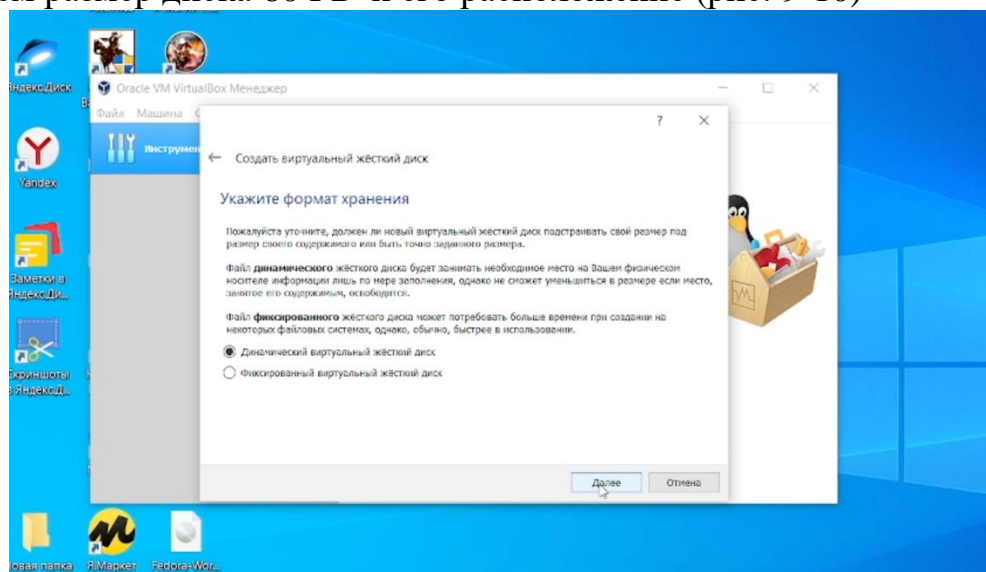


Рис. 9

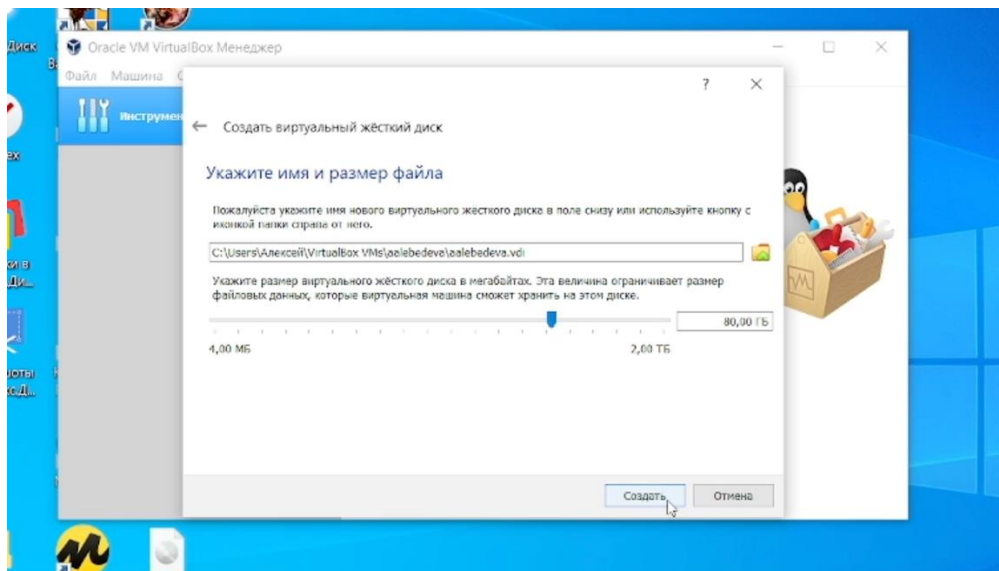


Рис. 10

- 8) Запускаем виртуальную машину
- 9) Заходим в Свойства - Носители в виртуальной машине и добавляем новый привод оптических дисков. Выбираем образ, который мы ранее скачали на наш компьютер (рис. 11-12)

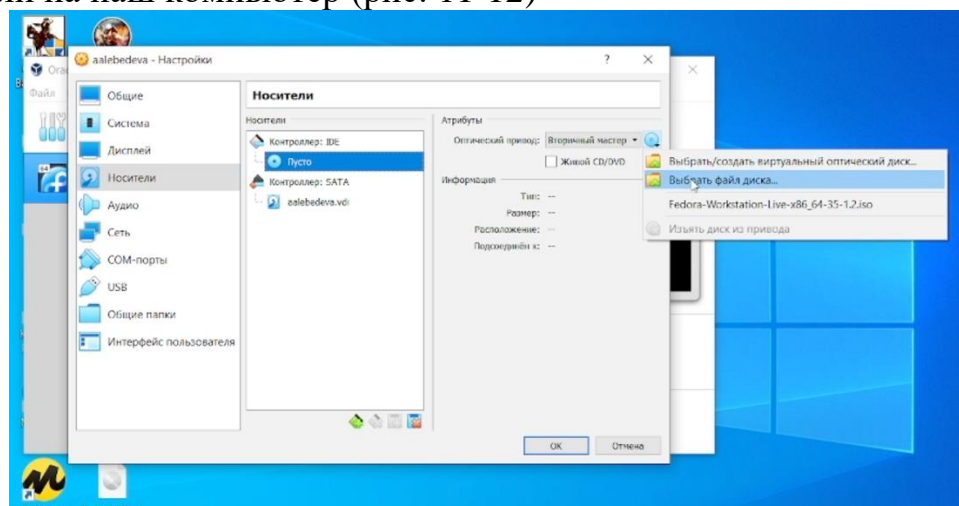


Рис. 11

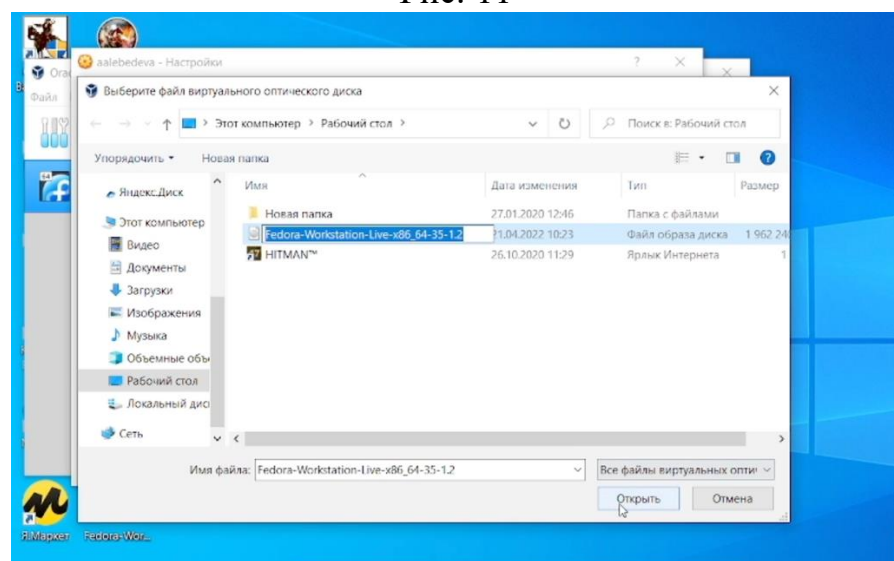


Рис. 12

- 10) Запуская виртуальную машину, выбираю язык интерфейса (рис. 13-15)

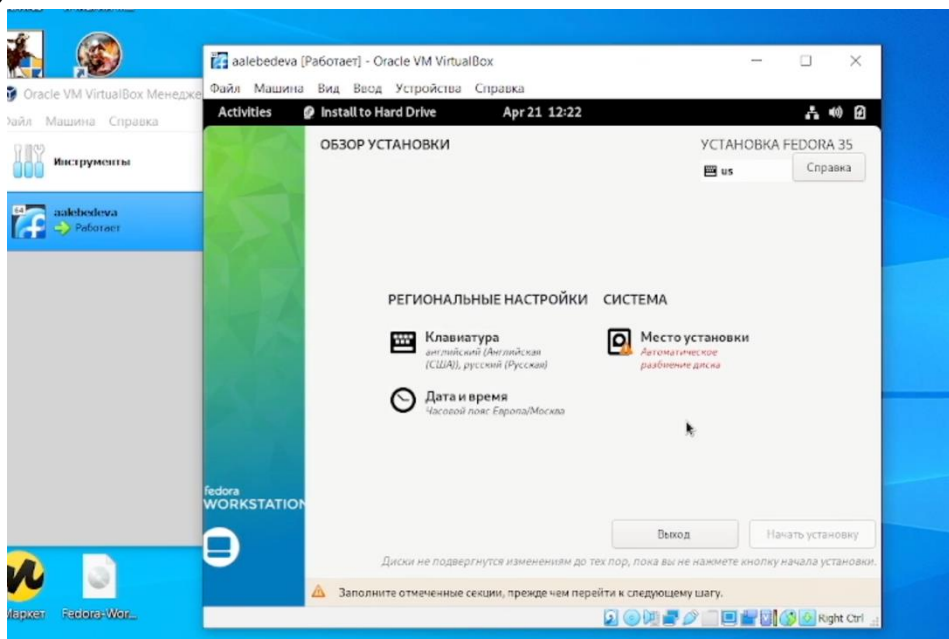


Рис. 13

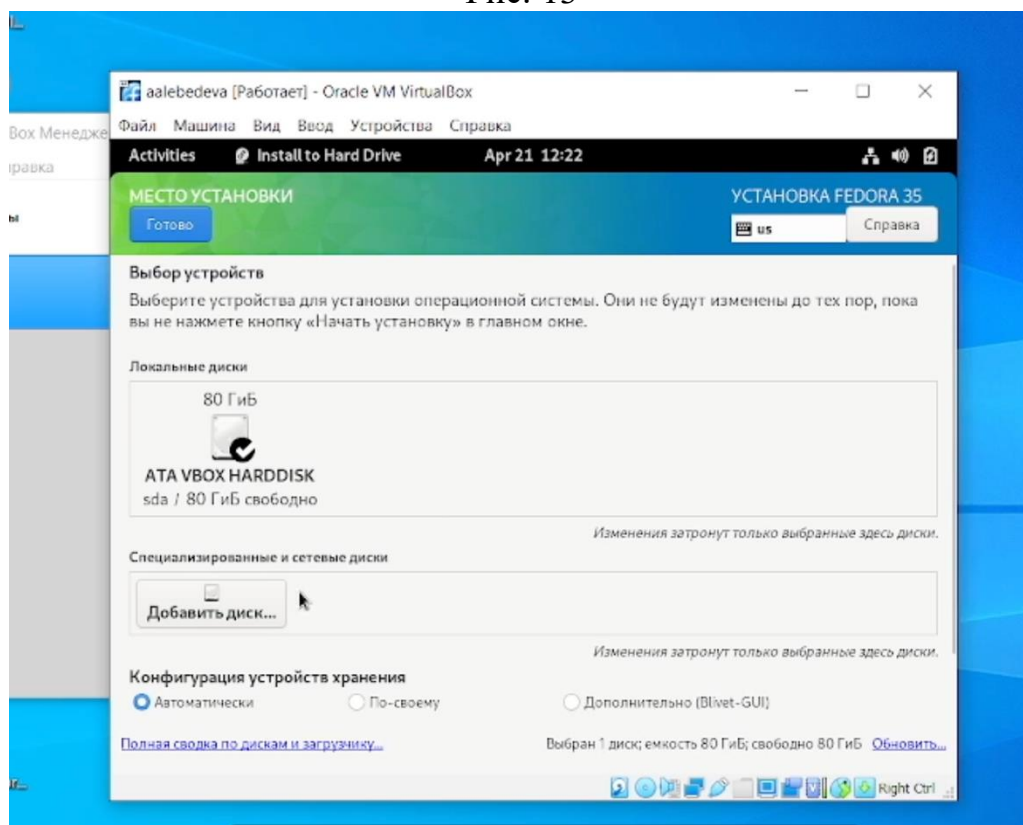


Рис. 14

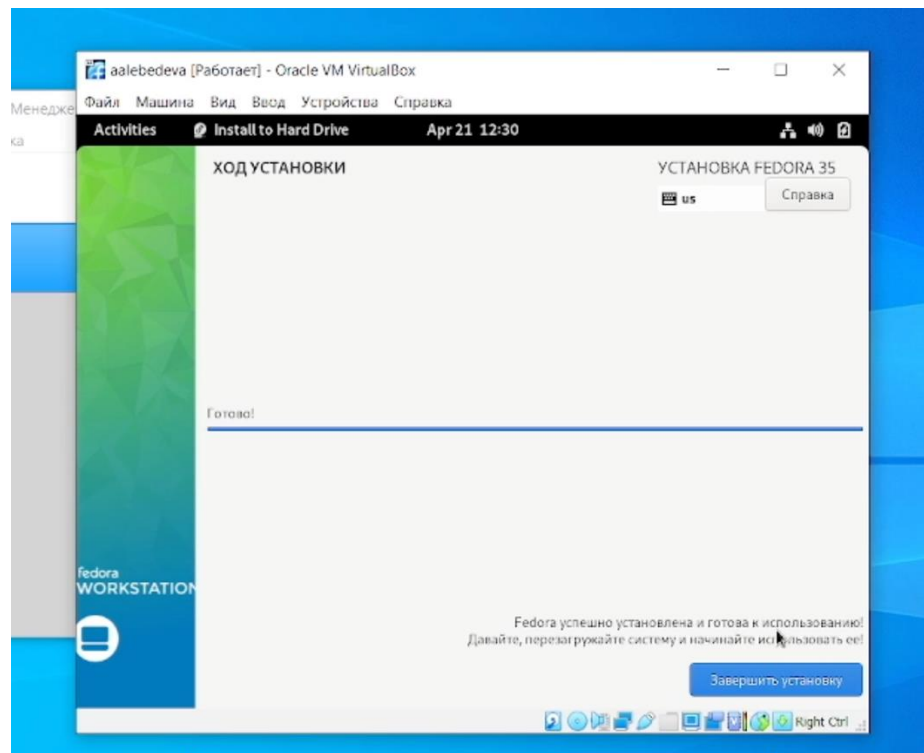


Рис. 15

- 11) После завершения установки виртуальной системы корректно перезапускаю виртуальную машину
- 12) Входим под заданной при установке учётной записью (Устанавливаю имя и пароль для пользователя) (рис.16-17)

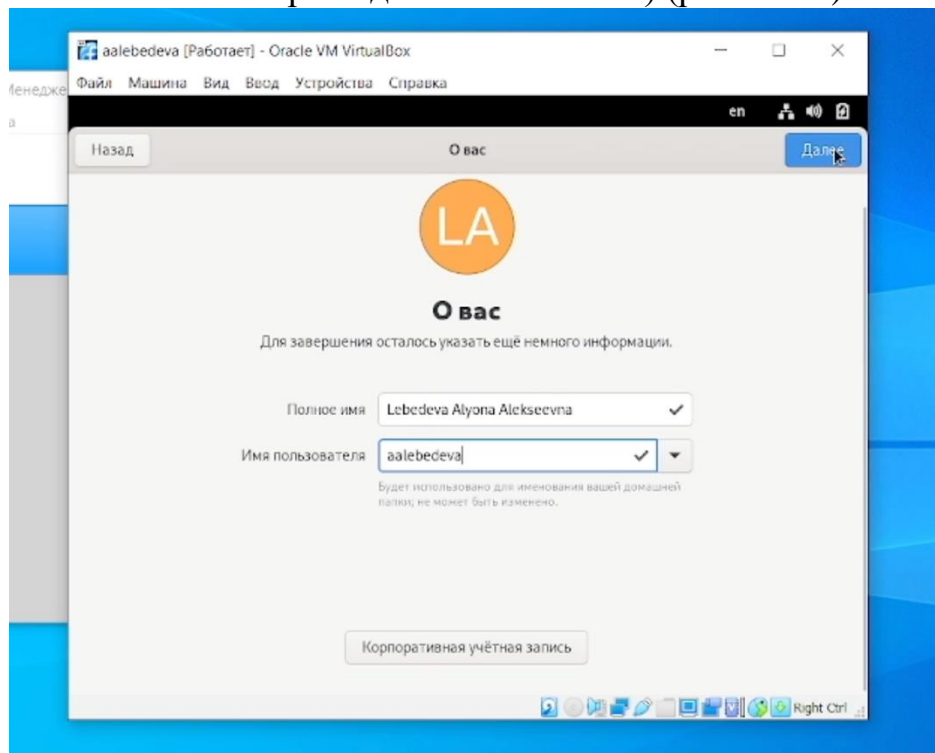


Рис. 16

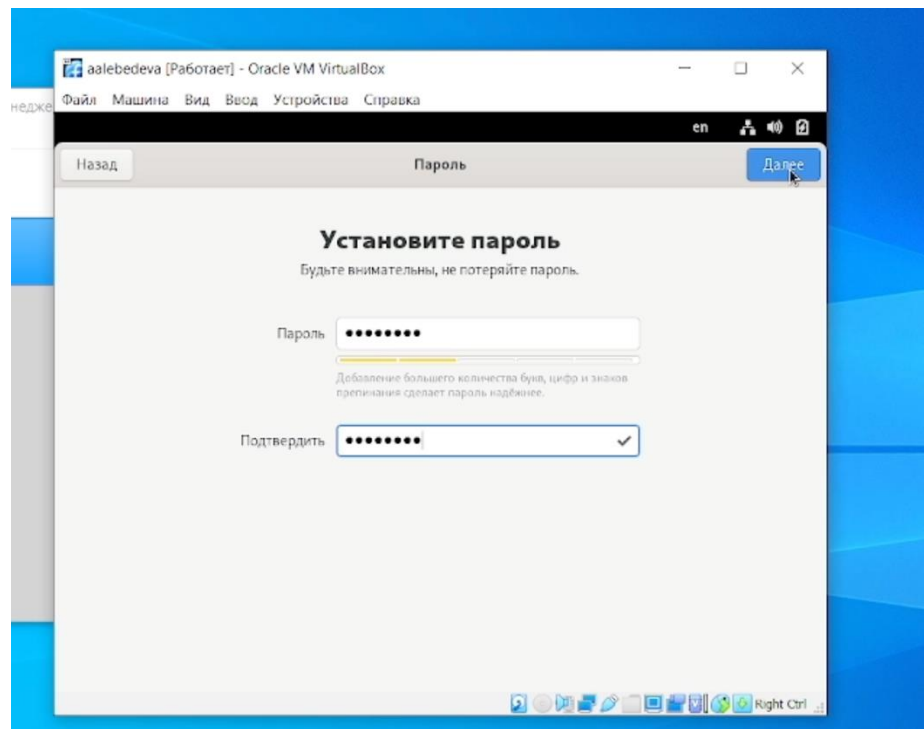


Рис. 17

- 13) В меню Устройства виртуальной машины подключаем образ диска дополнительной гостевой ОС (рис.18-20)

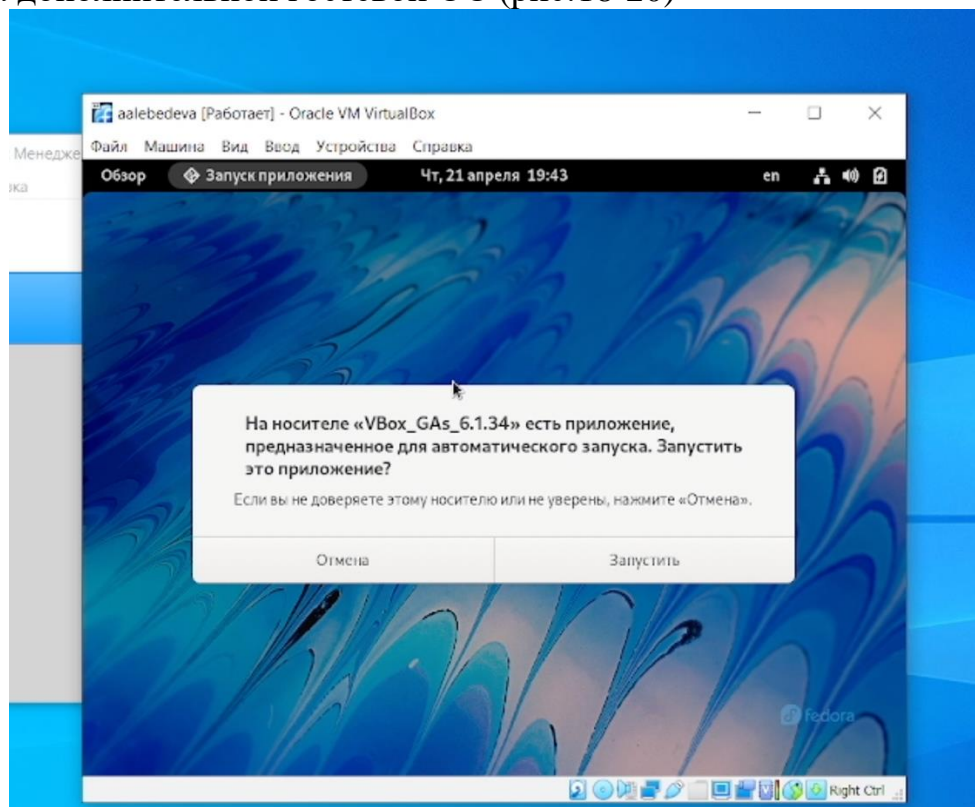


Рис. 18

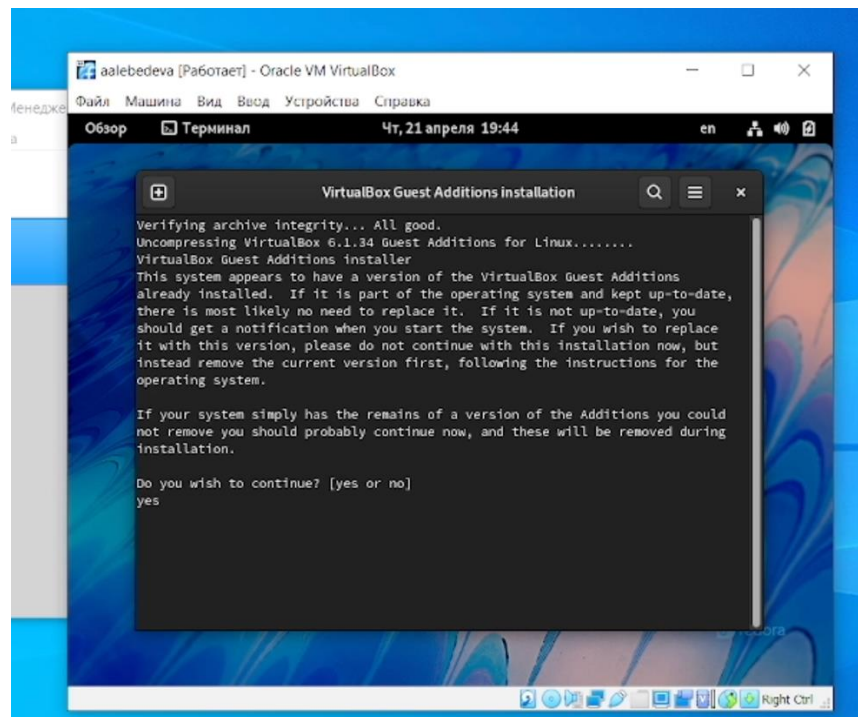


Рис. 19

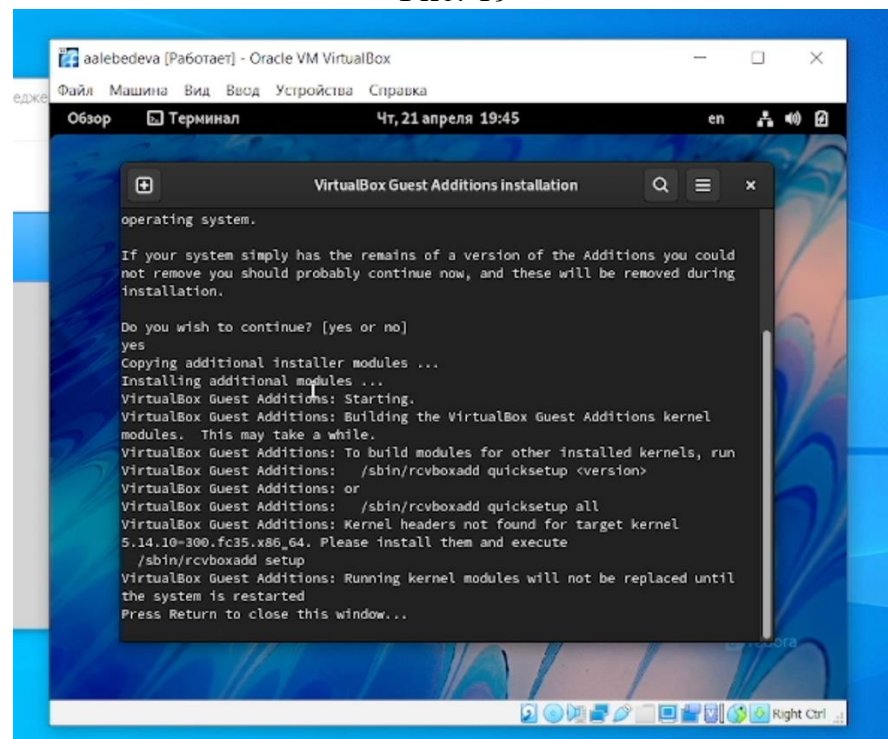


Рис. 20

- 14) После загрузки дополнительно нажимаем Enter и корректно перезагружаем виртуальную машину

Домашняя работа:

Загружаем графическое окружение и открываем терминал. В окне терминала анализирую последовательность загрузки системы, выполнив команду `dmesg` и введя пароль. (рис. 21-22)

```
aalebedeva@fedora:~$ dmesg | grep -i "Linux version"
[ 0.000000] Linux version 5.14.10-300.fc35.x86_64 (mockbuild@bkernel01.iad2.fedoraproject.org) (gcc (GCC) 11.2.1 20210728 (Red Hat 11.2.1-1), GNU ld version 2.37-10.fc35) #1 SMP Thu Oct 7 20:48:44 UTC 2021
[aalebedeva@fedora ~]$ dmesg | grep -i "Detected Mhz available"
[aalebedeva@fedora ~]$ dmesg | grep -i "MHz"
[ 0.000007] tsc: Detected 2400.000 MHz processor
[ 2.659505] e1000 0000:00:03:0 eth0: (PCI:33MHz:32-bit) 08:00:27:2a:d5:4e
[aalebedeva@fedora ~]$ dmesg | grep -i "CPU0"
[ 0.199693] smpboot: CPU0: Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz (family: 0x6, model: 0x9e, stepping: 0xa)
[aalebedeva@fedora ~]$ dmesg | grep -i "Memory available"
[aalebedeva@fedora ~]$ dmesg | grep -i "Memory"
[ 0.002114] ACPI: Reserving FACP table memory at [mem 0x7fff00f0-0x7fff01e3]
[ 0.002116] ACPI: Reserving DSDT table memory at [mem 0x7fff0470-0x7fff2794]
[ 0.002117] ACPI: Reserving FACS table memory at [mem 0x7fff0200-0x7fff023f]
[ 0.002118] ACPI: Reserving FACS table memory at [mem 0x7fff0200-0x7fff023f]
[ 0.002119] ACPI: Reserving APIC table memory at [mem 0x7fff0240-0x7fff0293]
[ 0.002120] ACPI: Reserving SSDT table memory at [mem 0x7fff02a0-0x7fff046b]
[ 0.005576] Early memory node ranges
[ 0.008848] PM: hibernation: Registered nosave memory: [mem 0x00000000-0x00000fff]
[ 0.008850] PM: hibernation: Registered nosave memory: [mem 0x0009f000-0x0009ffff]
```

Рис. 21

```
[ 2.285567] [TTM] Zone kernel: Available graphics memory: 1006644 KiB
[ 2.285743] [drm] Max dedicated hypervisor surface memory is 507904 kiB
[ 2.285744] [drm] Maximum display memory size is 16384 kiB
[aalebedeva@fedora ~]$ dmesg | grep -i "Hypervisor detected"
[ 0.000000] Hypervisor detected: KVM
[aalebedeva@fedora ~]$ dmesg | grep -i "Mount"
[ 0.085165] Mount-cache hash table entries: 4096 (order: 3, 32768 bytes, linear)
[ 0.085169] Mountpoint-cache hash table entries: 4096 (order: 3, 32768 bytes, linear)
[ 4.213049] systemd[1]: Set up automount Arbitrary Executable File Formats File System Automount Point.
[ 4.229888] systemd[1]: Mounting Huge Pages File System...
[ 4.231812] systemd[1]: Mounting POSIX Message Queue File System...
[ 4.235315] systemd[1]: Mounting Kernel Debug File System...
[ 4.243308] systemd[1]: Mounting Kernel Trace File System...
[ 4.335658] systemd[1]: Starting Remount Root and Kernel File Systems...
[ 4.372686] systemd[1]: Mounted Huge Pages File System.
[ 4.374215] systemd[1]: Mounted POSIX Message Queue File System.
[ 4.374410] systemd[1]: Mounted Kernel Debug File System.
[ 4.377371] systemd[1]: Mounted Kernel Trace File System.
[ 5.986229] EXT4-fs (sda1): mounted filesystem with ordered data mode. Opts: (null). Quota mode: none.
[aalebedeva@fedora ~]$
```

Рис. 22

Используем команду «dmesg | grep -I «то, что ищем»», чтобы найти необходимую информацию

1. Версия ядра Linux (Linux version).
2. Частота процессора (Detected Mhz processor).

3. Модель процессора (CPU0).
4. Объем доступной оперативной памяти (Memory available).
5. Тип обнаруженного гипервизора (Hypervisor detected).
6. Тип файловой системы корневого раздела.
7. Последовательность монтирования файловых систем.

Контрольные вопросы:

1) Учётная запись содержит:

- a. Имя пользователя (user name)
- b. Идентификационный номер пользователя (UID)
- c. Идентификационный номер группы (GID)
- d. Пароль (password)
- e. Полное имя (full name)
- f. Домашний каталог (home directory)
- g. Начальную оболочку (login shell)

2) Команды терминала:

- a. Для получения справок по команде: man. Например, команда «man ls» выведет справку о команде «ls»
- b. Для перемещения по файловой системе: cd. Например, команда «cd aalebedeva» осуществляет переход в каталог aalebedeva
- c. Для просмотра содержимого каталога: ls. Например, команда «ls desktop» отобразит содержимое рабочего стола
- d. Для определения объёма каталога: du. Например, команда «du -k ~/aalebedeva» выведет размер каталога aalebedeva в килобайтах
- e. Для создания / удаления каталогов / файлов: mkdir / rmdir / rm.
- f. Для создания определённых прав на файл / каталог: chmod [опции] [путь]
- g. Для просмотра истории команд: history. Например «history 3» покажет список последних трёх команд

3) Файловая система

- a. Архитектура хранения битов на жёстком диске
- b. Это организация каталогов в соответствии с идеологией Unix

Файловая система – 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, 8120 байтов.

Типы файловых систем:

Ext2, Ext3, Ext4 или **Extended Filesystem**— стандартная файловая система, первоначально разработанная еще для Minix. Содержит максимальное количество функций и является наиболее стабильной в связи с редкими изменениями кодовой базы. Начиная с ext3 в системе используется функция журналирования. Сегодня версия ext4 присутствует во всех дистрибутивах Linux.

JFS или **Journaled File System** разработана в IBM в качестве альтернативы для файловых систем ext. Сейчас она используется там, где необходима высокая стабильность и минимальное потребление ресурсов (в первую очередь в многопроцессорных компьютерах). В журнале хранятся только метаданные, что позволяет восстанавливать старые версии файлов после сбоев.

ReiserFS также разработана в качестве альтернативы ext3, поддерживает только Linux. Динамический размер блока позволяет упаковывать несколько небольших файлов в один блок, что предотвращает фрагментацию и улучшает работу с небольшими файлами. Недостатком является риск потери данных при отключении энергии.

XFS рассчитана на файлы большого размера, поддерживает диски до 2 терабайт. Преимуществом системы является высокая скорость работы с большими файлами, отложенное выделение места, увеличение разделов на лету, незначительный размер служебной информации. К недостаткам относится невозможность уменьшения размера, сложность восстановления данных и риск потери файлов при аварийном отключении питания.

Btrfs или **B-Tree File System** легко администрируется, обладает высокой отказоустойчивостью и производительностью. Используется как файловая система по умолчанию в OpenSUSE и SUSE Linux.

Другие ФС, такие как NTFS, FAT, HFS, могут использоваться в Linux, но корневая файловая система на них не устанавливается, поскольку они для этого не предназначены.

- 4) Команды «findmnt» или «findmnt—al» будет отображать все подмонтированные файловые системы или искать файловую систему

5) Для завершения используются сигналы:

Сигнал SIGINT (номер 2) обычно посылается процессу, если пользователь терминала дал команду прервать процесс (обычно эта команда – сочетание клавиш Ctrl-C) .

Сигнал SIGABRT (номер 6) посылается программе в результате вызова функции abort(3). В результате программа завершается с сохранением на диске образа памяти.

Сигнал SIGKILL (номер 9) завершает работу программы. Программа не может ни обработать, ни игнорировать этот сигнал.

Сигнал SIGSEGV (номер 11) посылается процессу, который пытается обратиться к не принадлежащей ему области памяти. Если обработчик сигнала не установлен, программа завершается с сохранением на диске образа памяти.

Сигнал SIGTERM (номер 15) вызывает «вежливое» завершение программы. Получив этот сигнал, программа может выполнить необходимые перед завершением операции (например, высвободить занятые ресурсы). Получение SIGTERM свидетельствует не об ошибке в программе, а о желании ОС или пользователя завершить ее.

Сигнал SIGCHLD (номер 17) посылается процессу в том случае, если его дочерний процесс завершился или был приостановлен. Родительский процесс также получит этот сигнал, если он установил режим отслеживания сигналов дочернего процесса и дочерний процесс получил какой-либо сигнал. По умолчанию сигнал SIGCHLD игнорируется.

Сигнал SIGCONT (номер 18) возобновляет выполнение процесса, остановленного сигналом SIGSTOP.

Сигнал SIGSTOP (номер 19) приостанавливает выполнение процесса. Как и SIGKILL, этот сигнал не возможно перехватить или игнорировать.

Сигнал SIGTSTP (номер 20) приостанавливает процесс по команде пользователя (обычно эта команда – сочетание клавиш Ctrl-Z).

Сигнал SIGIO/SIGPOLL (в Linux обе константы обозначают один сигнал – номер 29) сообщает процессу, что на одном из дескрипторов, открытых асинхронно, появились данные. По умолчанию этот сигнал, как ни странно, завершает работу программы.

Вывод:

В ходе данной лабораторной работы я изучила, как установить операционную систему на виртуальную машину и настроить минимально необходимые для дальнейшей работы сервисы.