

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ
Факультет физико – математических и естественных наук
Кафедра прикладной математики и теории вероятностей

Отчет
по лабораторной работе №1
Дисциплина: Операционные системы

Студент: Ничипорова Елена Дмитриевна

Группа: НПМ-02-21

МОСКВА

2022

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

Ход работы

Загружаем в дисплейном классе операционную систему Linux и запускаем виртуальную машину, введя в командной строке команду: VirtualBox & (рис.1)

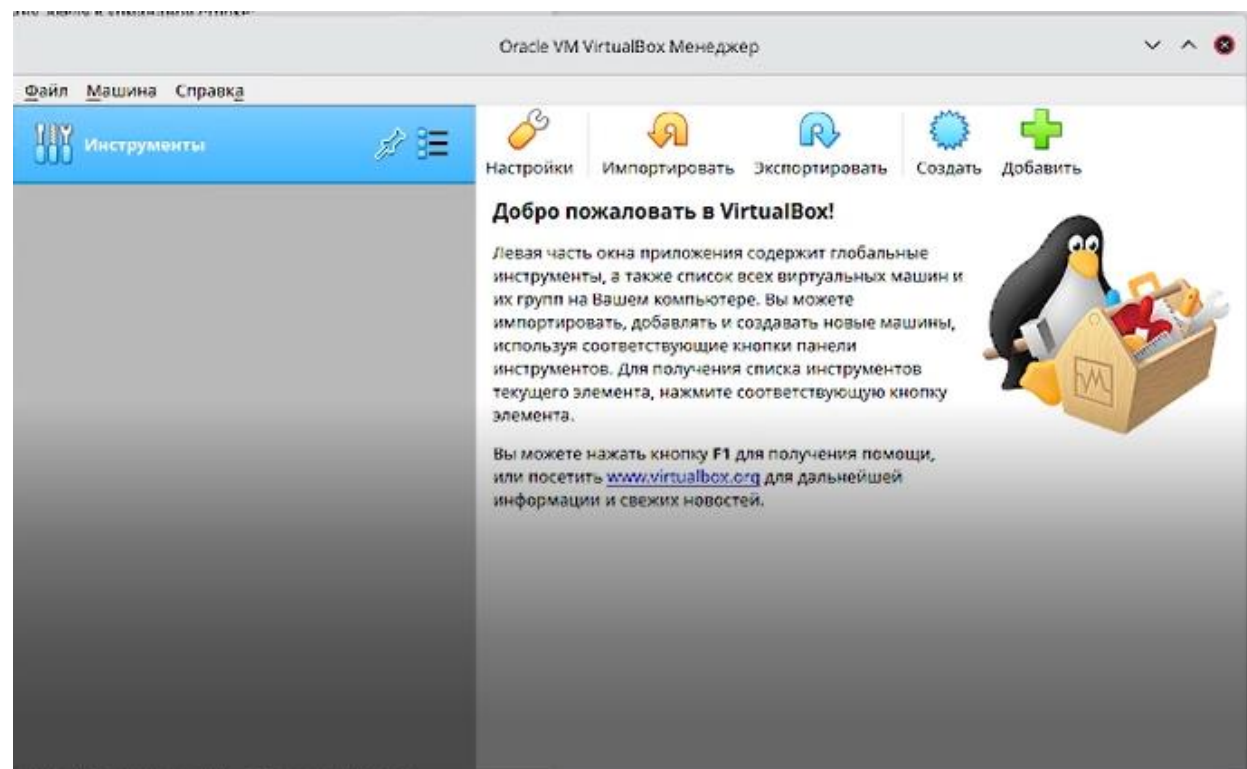


Рис.1

Далее проверяем в VirtualBox месторасположение каталога для виртуальной машины. Для этого в VirtualBox выбираем файл->свойства->общие->папка для машин. В настройках должно стоять /var/tmp/имя_пользователя (рис.2)

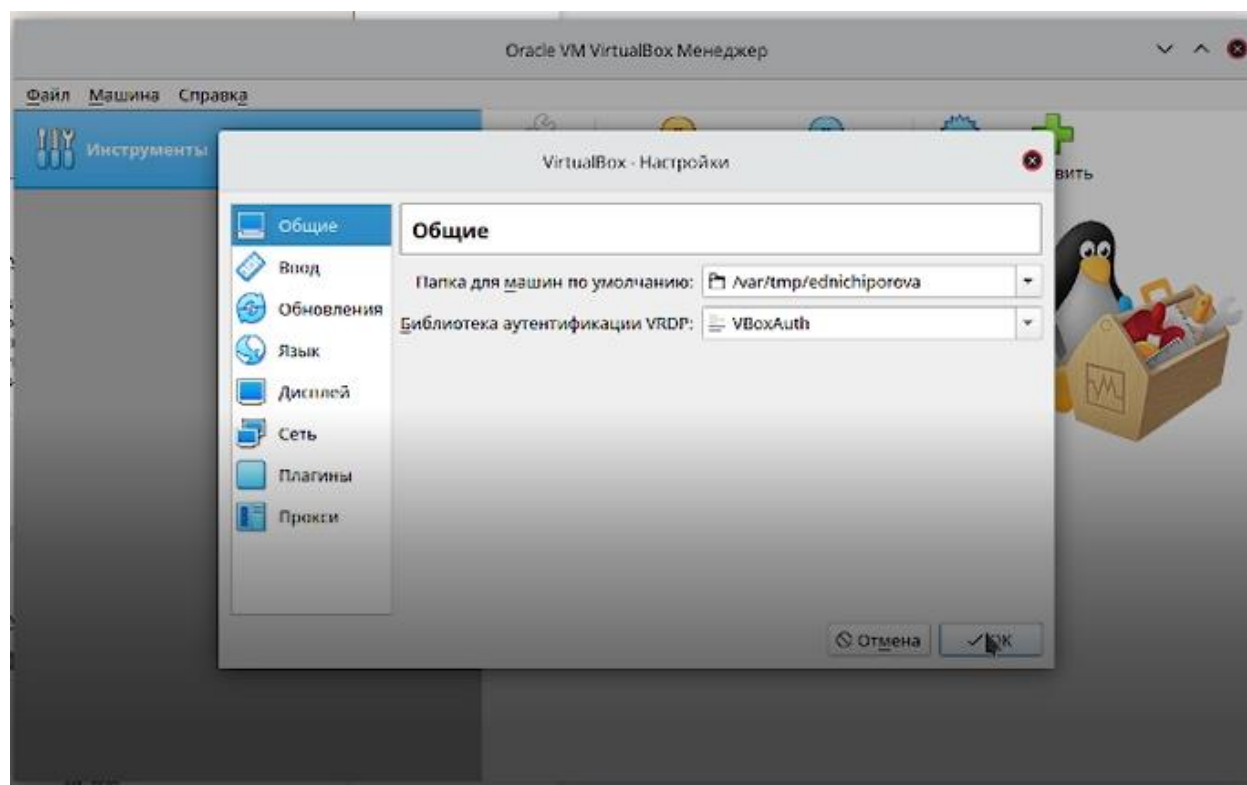


Рис.2

Создаем виртуальную машину. Для этого в VirtualBox выбираем машина->создать. Далее указываем имя виртуальной машины (мой логин в дисплейном классе) (рис.3)

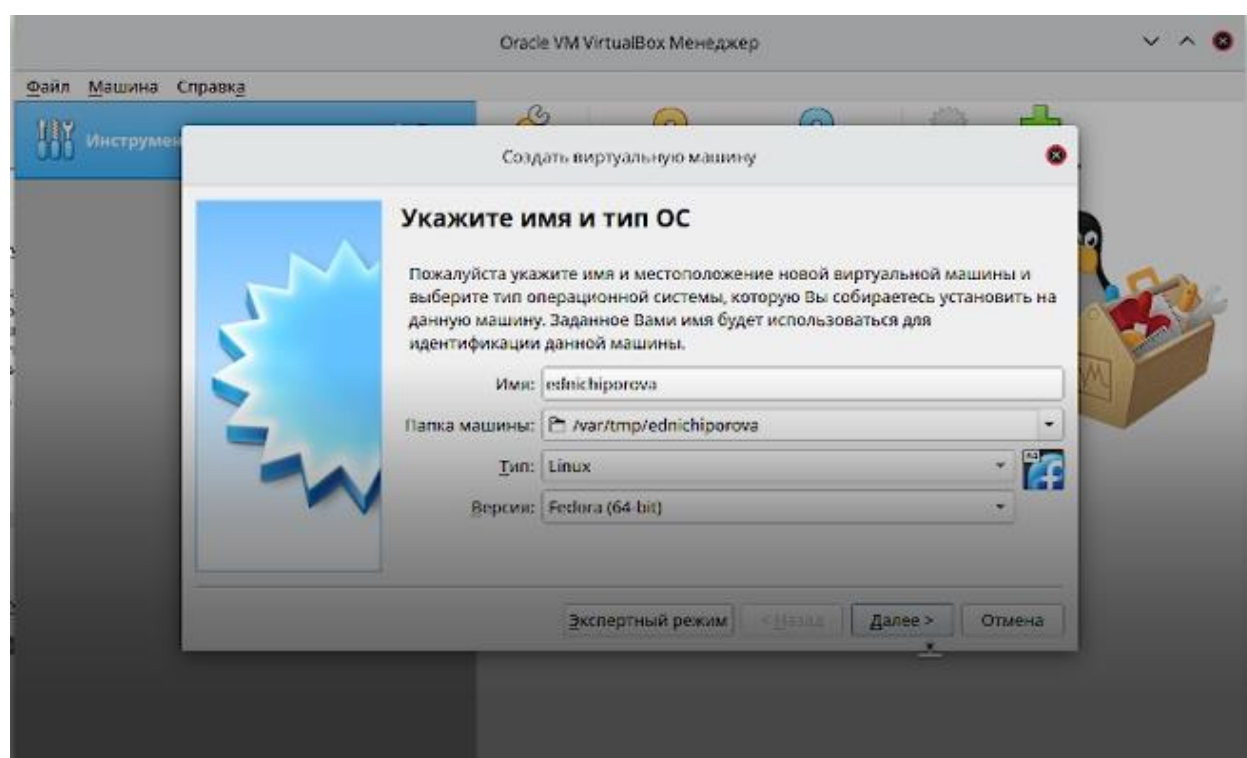


Рис.3

Указываем объем памяти – 4096 МБ (рис.4)

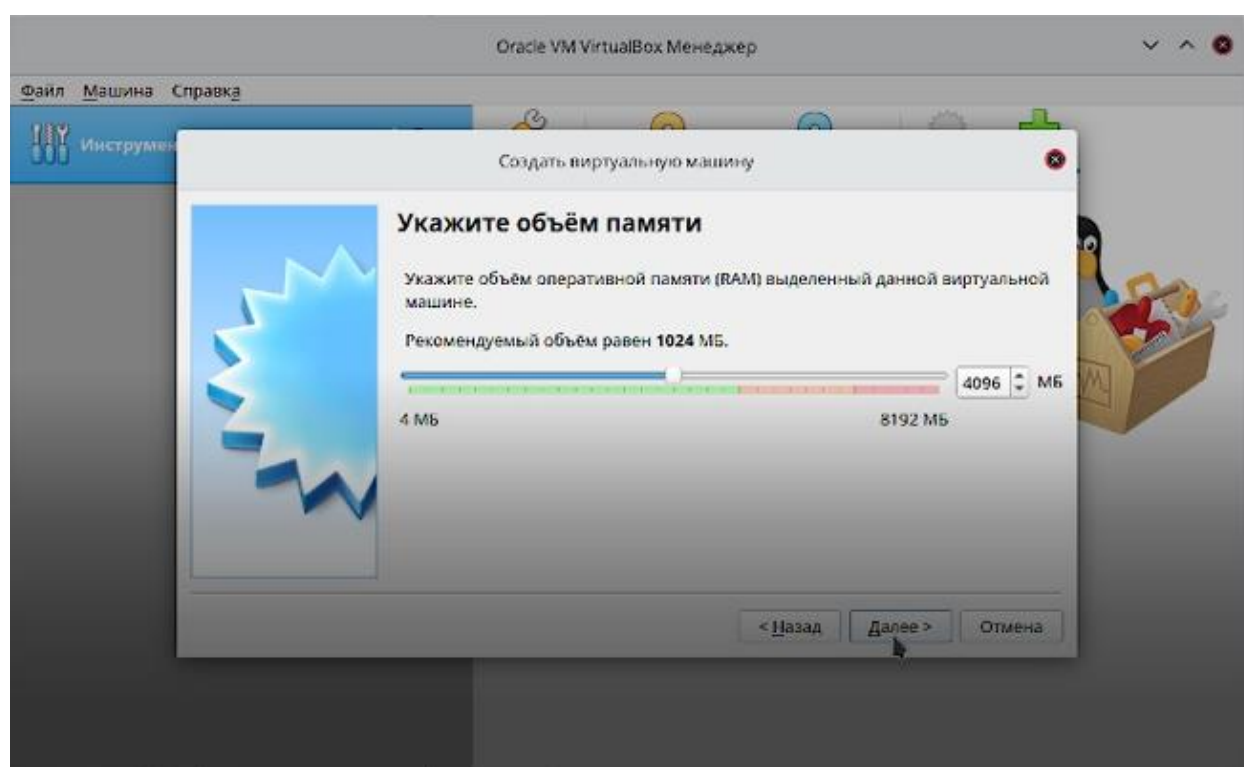


Рис.4

Задаем конфигурацию жесткого диска - загрузочный, VDI (VirtualBox Disk Image), динамический виртуальный диск (рис 5-7)

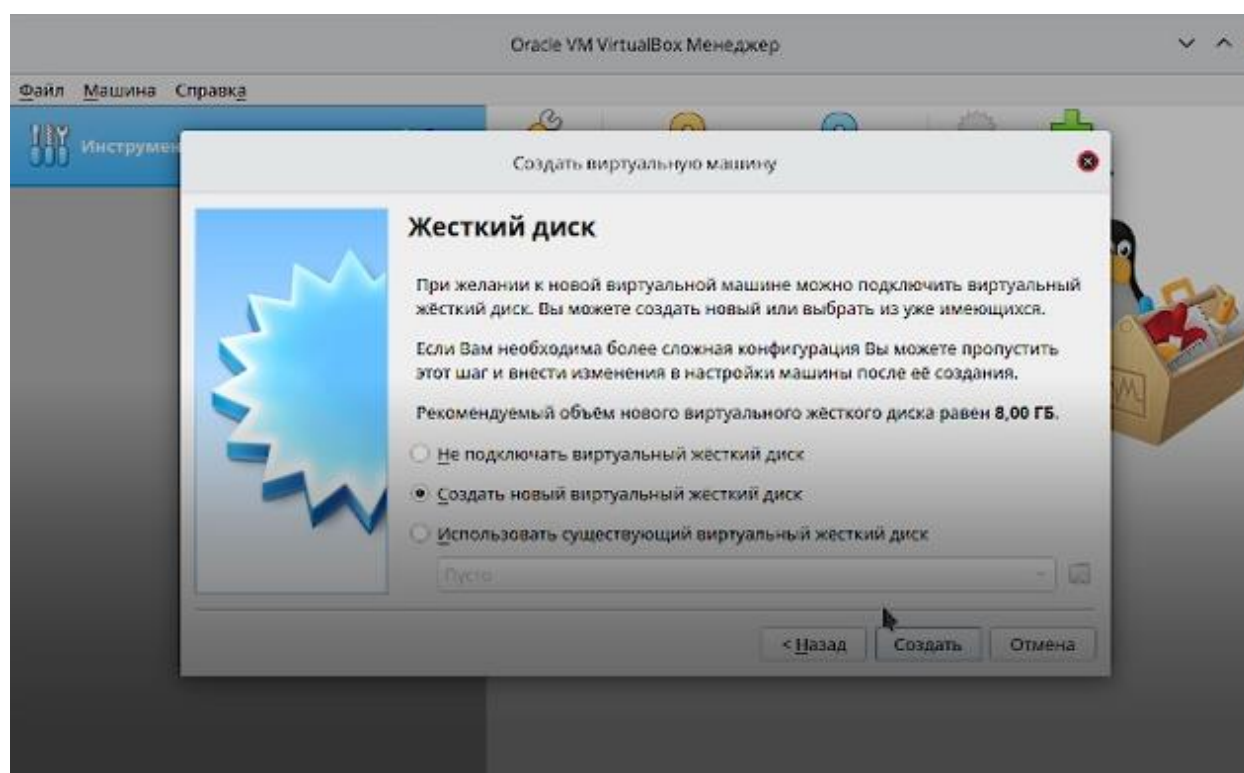


Рис.5

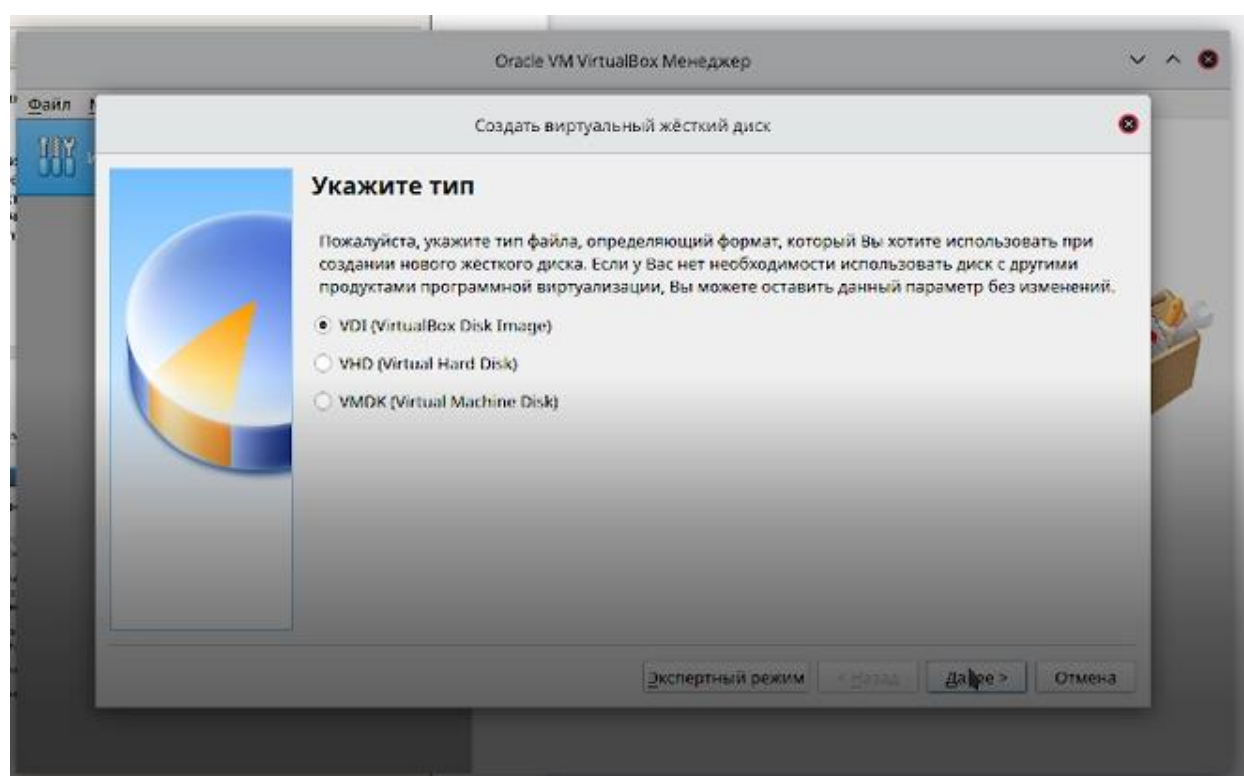


Рис.6

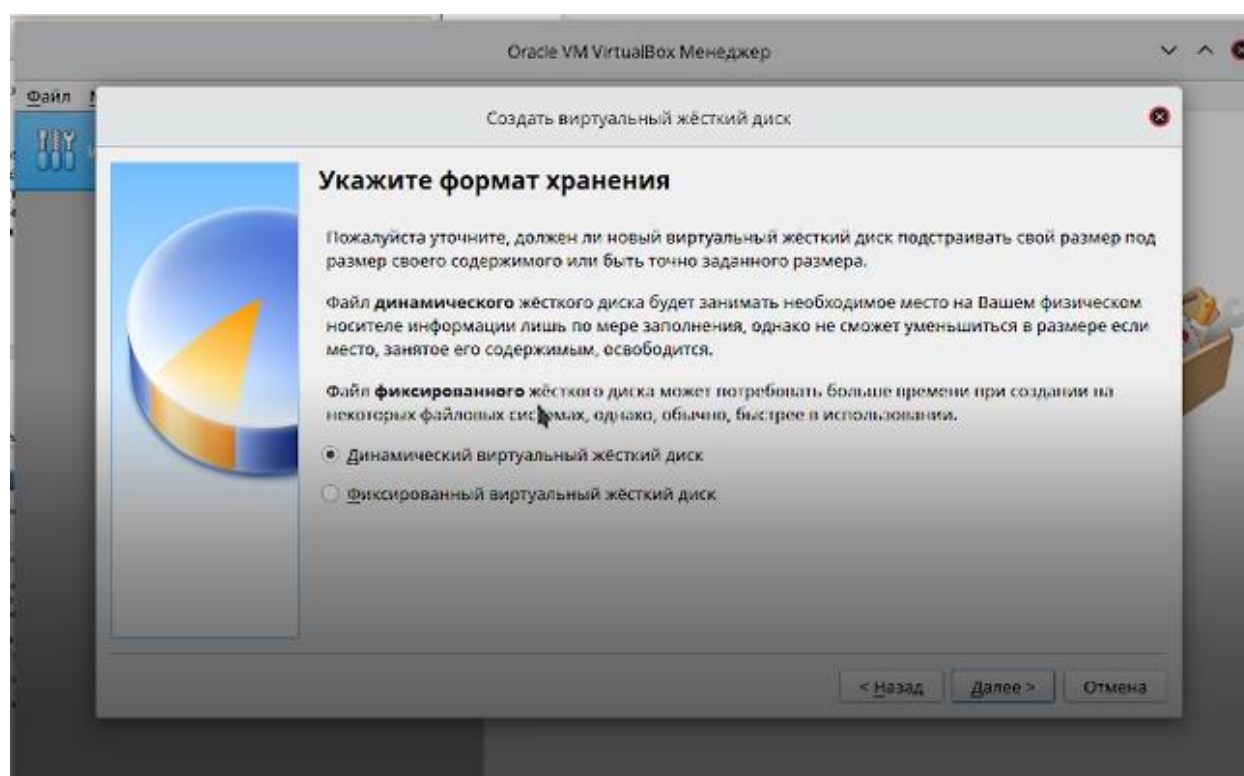


Рис.7

Указываем размер и формат хранения – 80 ГБ, его расположение — в данном случае /var/tmp/имя_пользователя/fedora.vdi (рис.8)

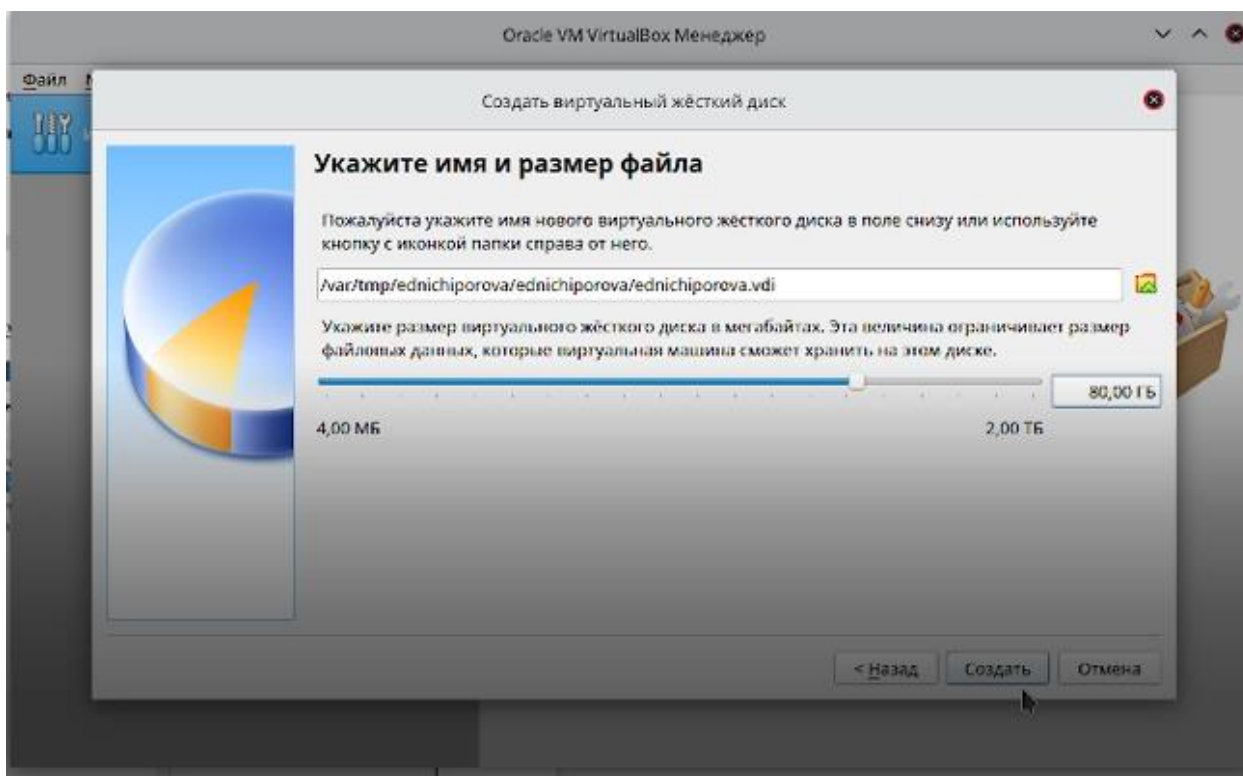


Рис.8

Дальше выбираем в VirtualBox носитель виртуальной машины. И запускаем виртуальную машину (рис.9)

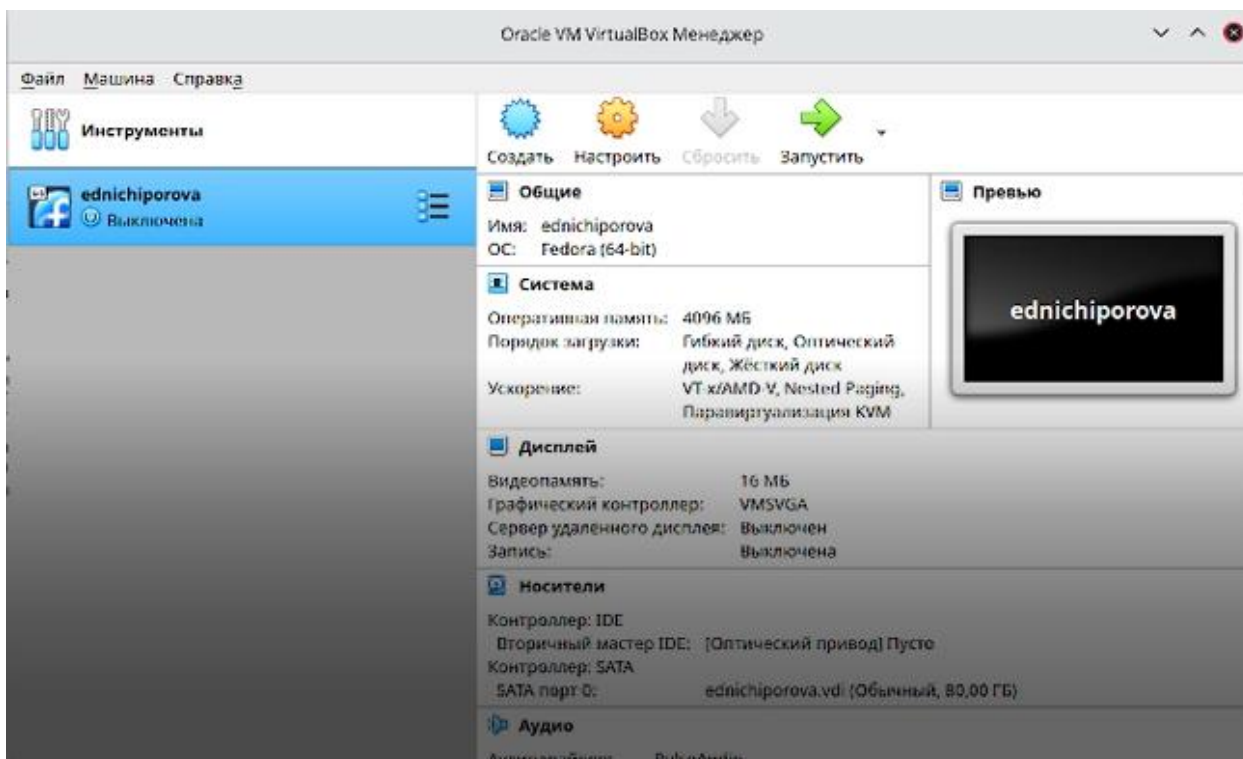


Рис.9

После установки указываем язык, время, имя пользователя. Я делаю отчеты с помощью скриншотов из моего видео, которое записывала в дисплейном классе. К сожалению, оно сохранилось не полностью, а времени, чтобы его переделать, у меня уже нет. Поэтому дальше я буду описывать свои действия без скриншотов.

После установки надо было перезагрузить виртуальную машину, а также отключить оптический диск (проверить его отключение, если такого не произошло, то самостоятельно удалить его)

Дальше надо было запустить виртуальную машину и залогиниться и получить права администратора. Все эти действия помогут нам при выполнении домашнего задания.

Домашнее задание:

Необходимо было получить следующую информацию:

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. Вот скрин с выполнения моего домашнего задания (рис. 10)

```
ednichiporova@dk4n71 ~$ dmesg | grep -i "Linux"
[ 0.148488] ACPI: Added _OSI(Linux-Dell-Video)
[ 0.148591] ACPI: Added _OSI(Linux-Lenovo-NV-HDMI-Audio)
[ 0.148697] ACPI: Added _OSI(Linux-HPI-Hybrid-Graphics)
[ 0.456788] mc: Linux media interface: v0.10
[ 0.457286] videodev: Linux video capture interface: v2.00
[ 0.457413] pps_core: LinuxPPS API ver. 1 registered
[ 0.457519] pps_core: Software ver. 5.3.6 - Copyright 2005-2007 Rodolfo Giometti <giometti@linux.it>
[ 2.659107] Linux agpgart interface v0.103
[ 6.101677] TERM=linux
[ 6.888118] usb usb1: Manufacturer: Linux 5.10.35-gentoo xhci-hcd
[ 6.889938] usb usb2: Manufacturer: Linux 5.10.35-gentoo xhci-hcd
[ 9.981652] systemd[1]: systemd 248 running in system mode. (+PAM +AUDIT -SELINUX -APPARMOR +IMA +SMACK +SECCOMP +LZ4 +XZ +ZLIB +ZSTD +XKBCOMMON +UTMP +SYSVINIT default-hierarchy=unified)
ednichiporova@dk4n71 ~$ dmesg | grep -i "MHz"
[ 0.506711] hpet0: 8 comparators, 64-bit 24.000000 MHz counter
[ 3.589623] tsc: Refined TSC clocksource calibration: 1703.995 MHz
ednichiporova@dk4n71 ~$ dmesg | grep -i "CPU0"
[ 0.260897] ACPI: SSDT 0xFFFF888100AB6000 000474 (v02 PmRef Cpu0Ist 00003000 INTL 20160527)
[ 0.266671] ACPI: SSDT 0xFFFF888100B0A800 000400 (v02 PmRef Cpu0Cst 00003001 INTL 20160527)
[ 0.269407] ACPI: SSDT 0xFFFF888100D4F200 00011B (v02 PmRef Cpu0Hwp 00003000 INTL 20160527)
ednichiporova@dk4n71 ~$ dmesg | grep -i "Memory"
[ 2.567520] Freeing initrd memory: 216704K
[ 2.584142] check: Scanning for low memory corruption every 60 seconds
[ 2.658931] Non-volatile memory driver v1.3
[ 6.085375] Freeing unused kernel image (initmem) memory: 1080K
[ 6.101147] Freeing unused kernel image (text/rodata gap) memory: 2044K
[ 6.101389] Freeing unused kernel image (rodata/data gap) memory: 216K
[ 15.139327] i2c i2c-0: 1/2 memory slots populated (from DMI)
[ 16.118944] amdgpu 0000:01:00.0: amdgpu: Trusted Memory Zone (TMZ) feature not supported
[ 16.332229] [TTM] Zone kernel: Available graphics memory: 4000178 KiB
[ 16.332251] [TTM] Zone dma32: Available graphics memory: 2097152 KiB
[ 16.332320] [drm] amdgpu: 2048M of VRAM memory ready
[ 16.332336] [drm] amdgpu: 3072M of GTT memory ready.
ednichiporova@dk4n71 ~$ dmesg | grep -i "Hypervisor detected"
ednichiporova@dk4n71 ~$ dmesg | grep -i "Mount"
[ 7.813399] EXT4-fs (sda8): mounted filesystem with ordered data mode. Opts: (null)
[ 11.778069] systemd[1]: Set up automount Arbitrary Executable File Formats File System Automount Point.
[ 11.785967] systemd[1]: Mounting Huge Pages File System...
[ 11.786697] systemd[1]: Mounting POSIX Message Queue File System...
[ 13.766054] EXT4-fs (sda8): re-mounted. Opts: user_xattr,acl
[ 19.041353] EXT4-fs (sda6): mounted filesystem with ordered data mode. Opts: (null)
ednichiporova@dk4n71 ~$
```

Рис.10

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 ($8 \cdot 2^{60}$ 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 ($16 \cdot 2^{40}$ 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 ($32 \cdot 2^{50}$ 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 ($4 \cdot 2^{40}$ байт) для 32-х разрядных систем. На данный 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 большой 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), специального 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 дамп памяти. Сочетание клавиш Ctrl+/,

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.

Вывод:

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 поиска информации об установленной операционной системе, используя консоль.