

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: Леон Атупанья Хосе Фернандо

Группа: НПМбд-02-20

МОСКВА

2021_

1. Цель работы

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

2. Ход работы

Для начала надо скачать VirtualBox, необходимую для запуска виртуальных машин. Скачать можно на сайте: <https://www.virtualbox.org/> (рисунок 1).

Необходимо выбрать версию своей операционной системы (рисунок 2).

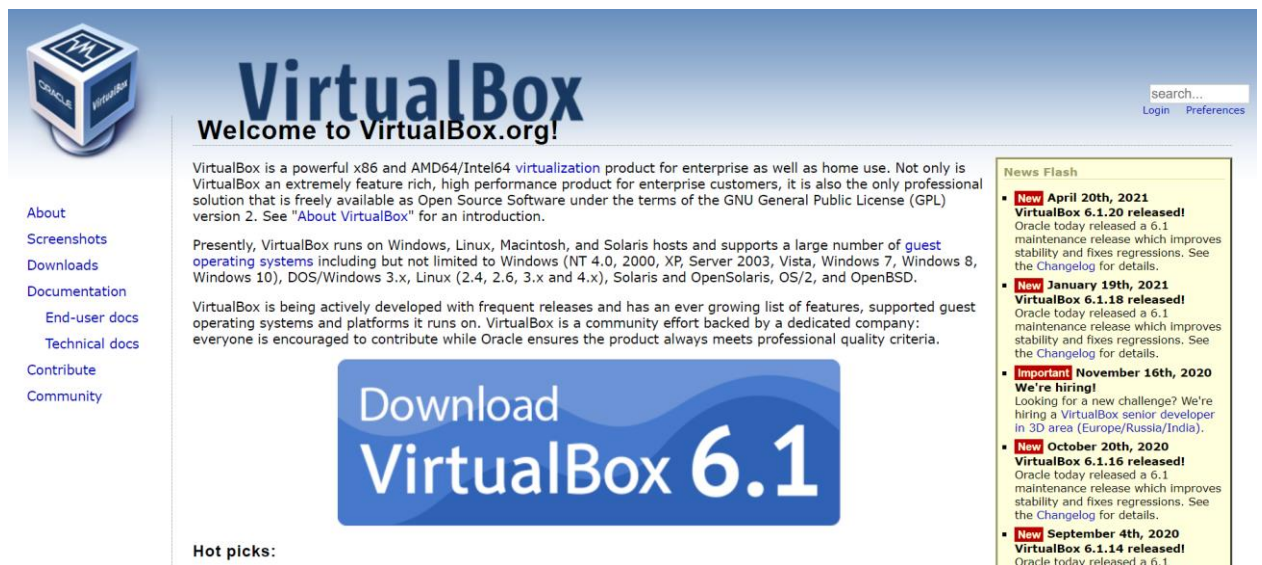


Рисунок 1



Рисунок 2

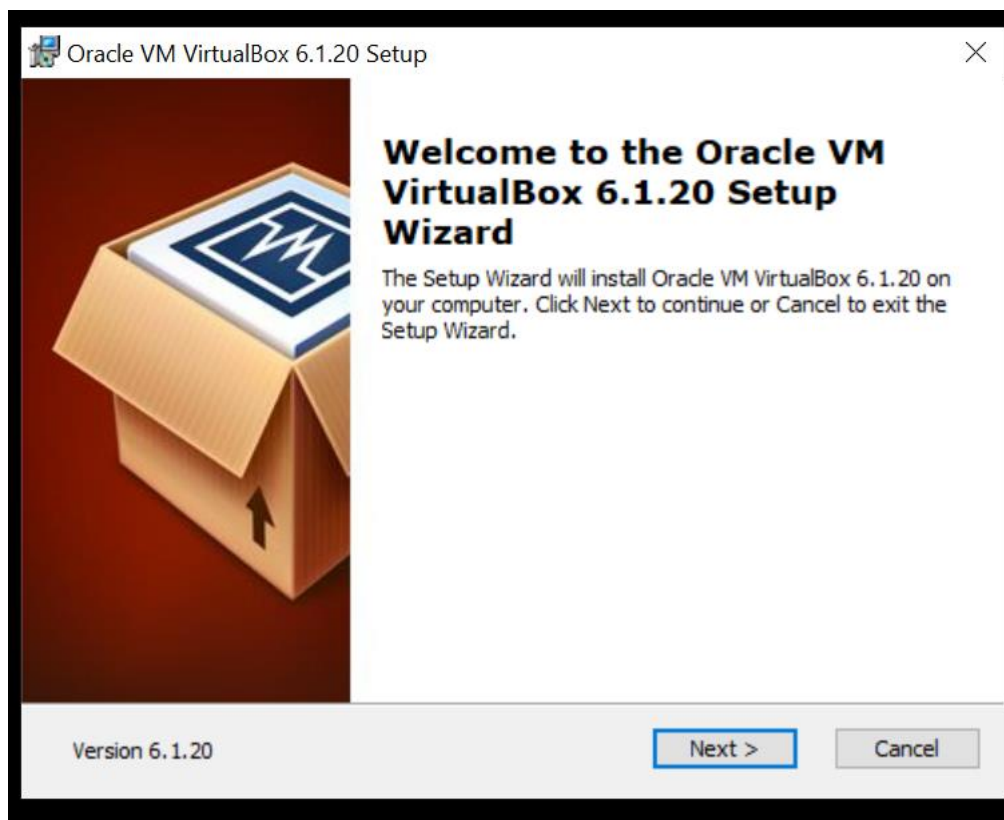


Рисунок 3

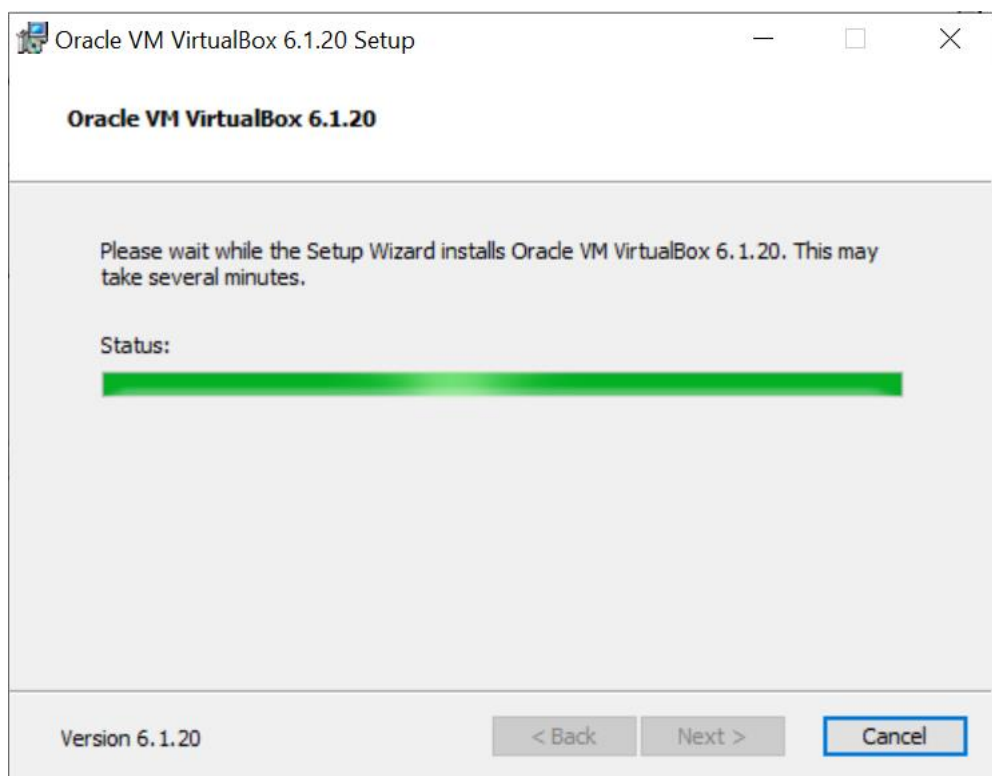
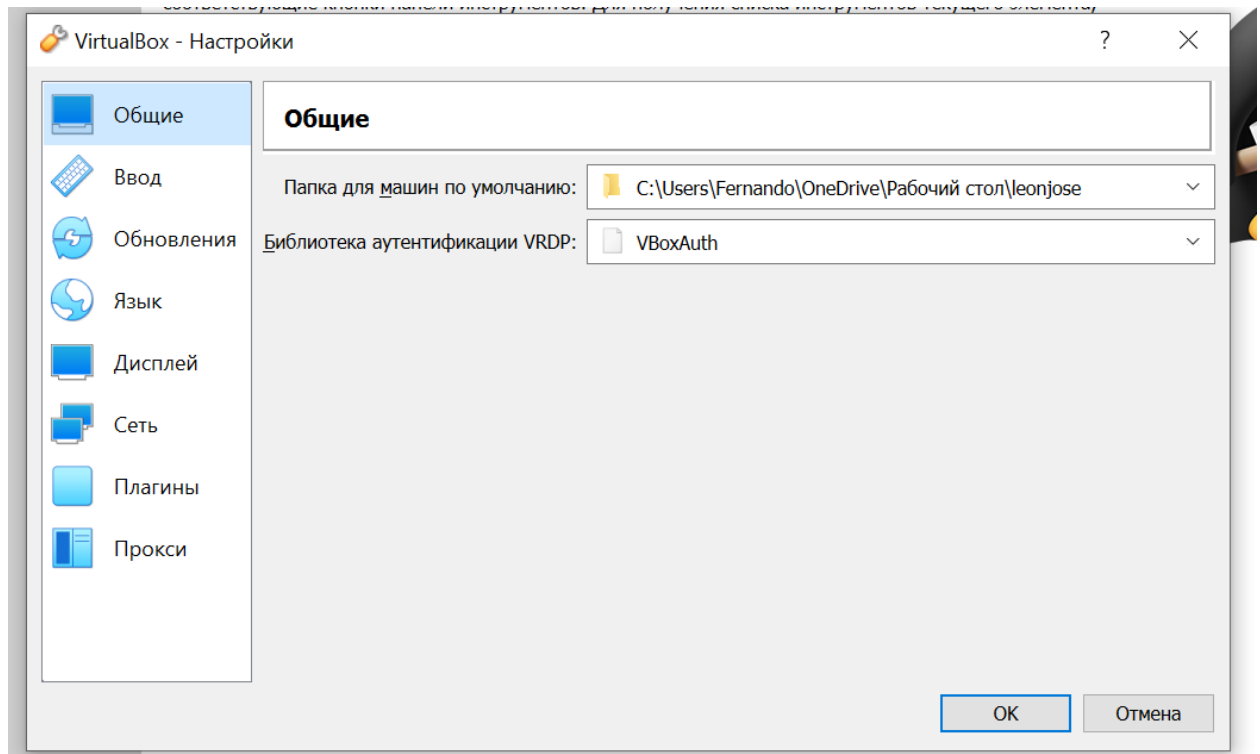


Рисунок 4

Далее создаём на рабочем столе папку, в которой будет храниться наша виртуальная машина. Имя папки — имя пользователя (логин студента в дисплейном классе). В данном случае «leonjose». Проверяем в свойствах

VirtualBox месторасположения папки для виртуальных машин. Для этого открываем VirtualBox, далее «файл» → «Свойства» → вкладка «Общие» и в поле «Папка для машин по умолчанию» указываем путь к папке, созданной ранее (рисунок 5).



Переходим к созданию виртуальную машину. Для этого нажимаем «Машина» → «Создать» и указываем имя виртуальной машины (логин в дисплейной классе, «leonjose») и тип операционной системы – Linux, RedHat 64 bits. (рисунок 6)

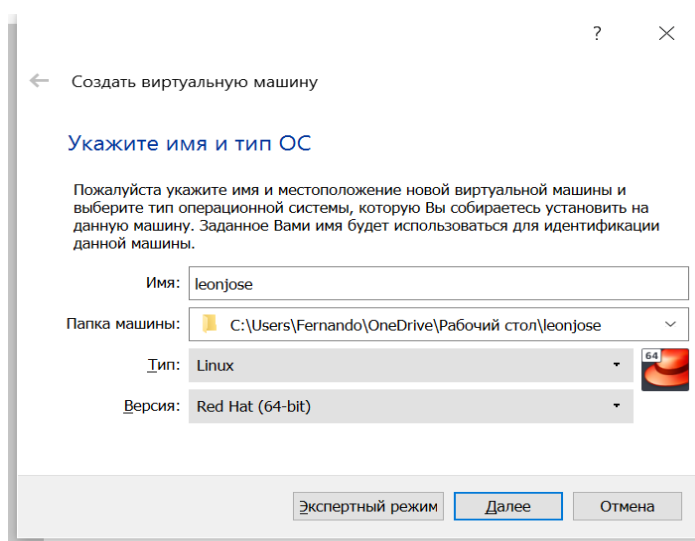


Рисунок 6

Указываем размер основной памяти виртуальной машины – 1024 МБ
(рисунок 7)

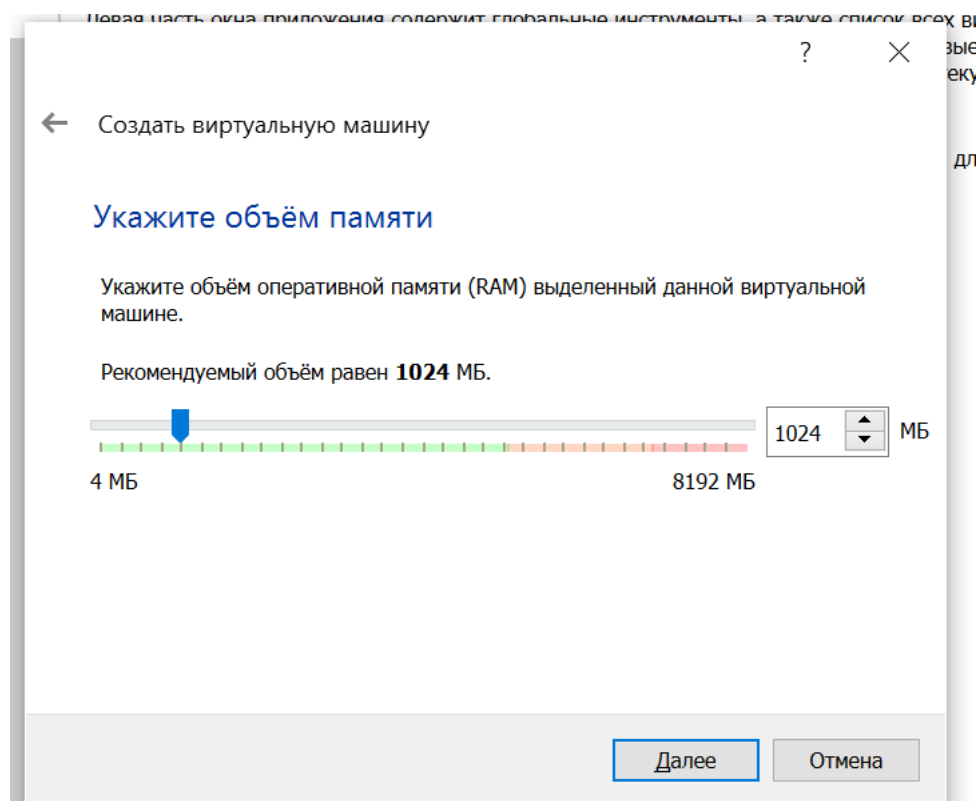


Рисунок 7

Создаем новый виртуальный жесткий диск (рисунок 8)

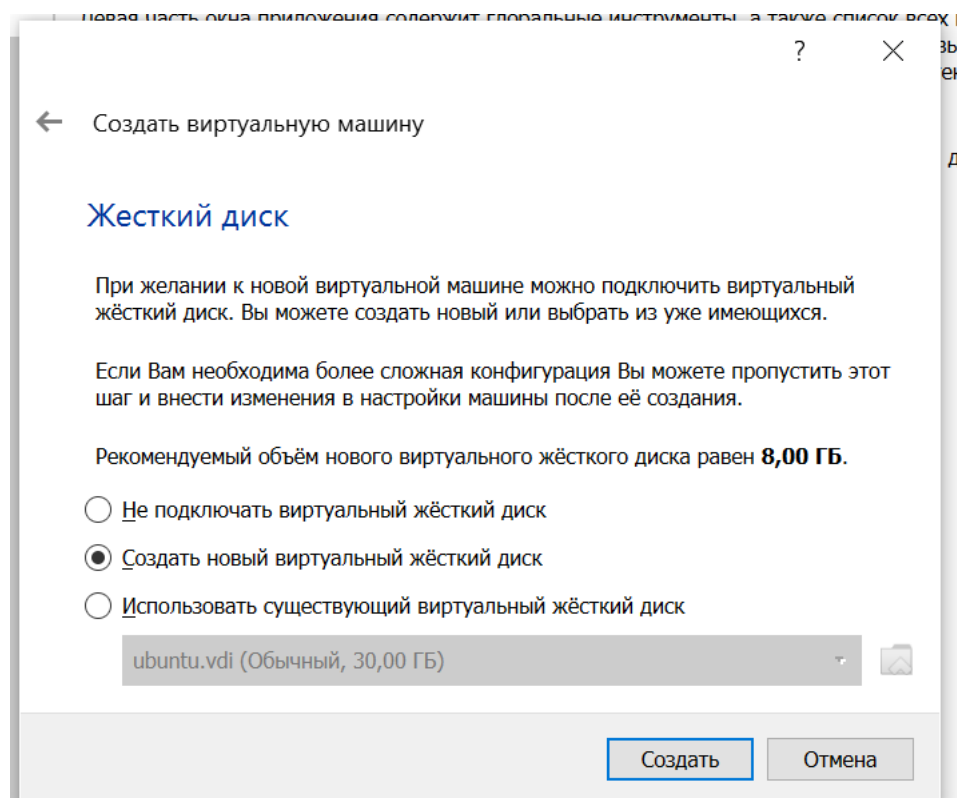


Рисунок 8

Задаем конфигурацию жесткого диска – VDI (VirtualBox disk Image), динамический виртуальный жесткий диск (рисунки 9 и 10).

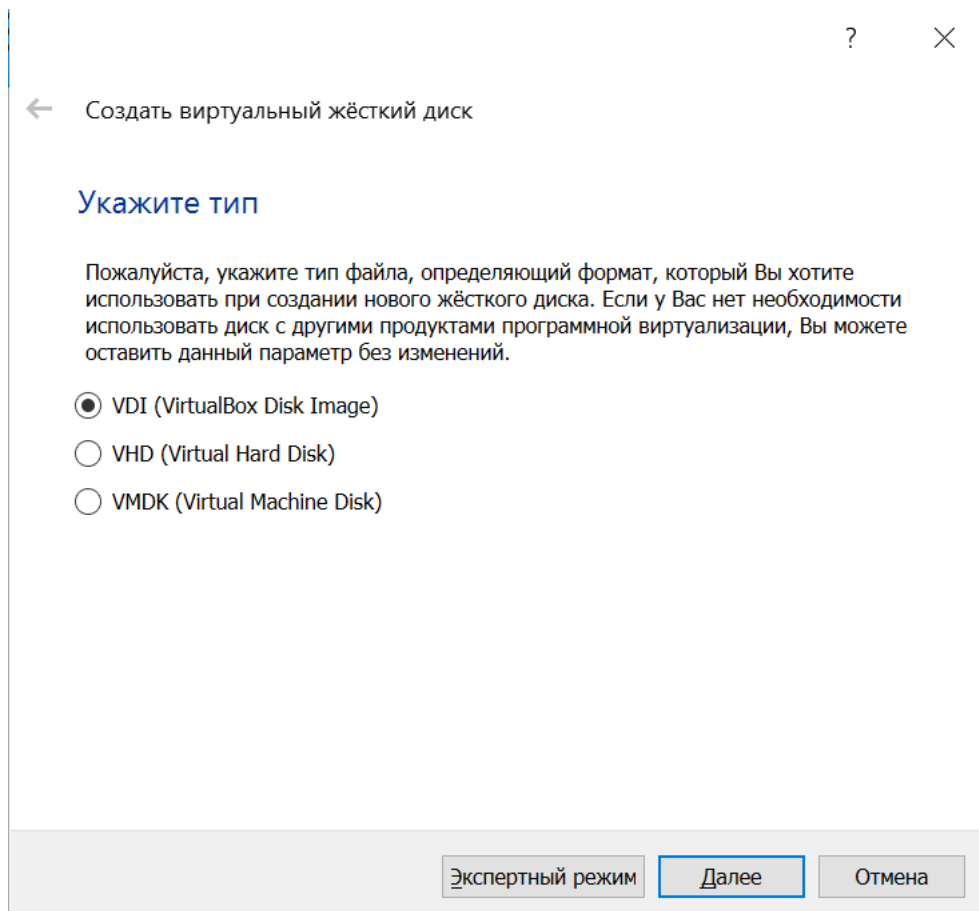


Рисунок 9

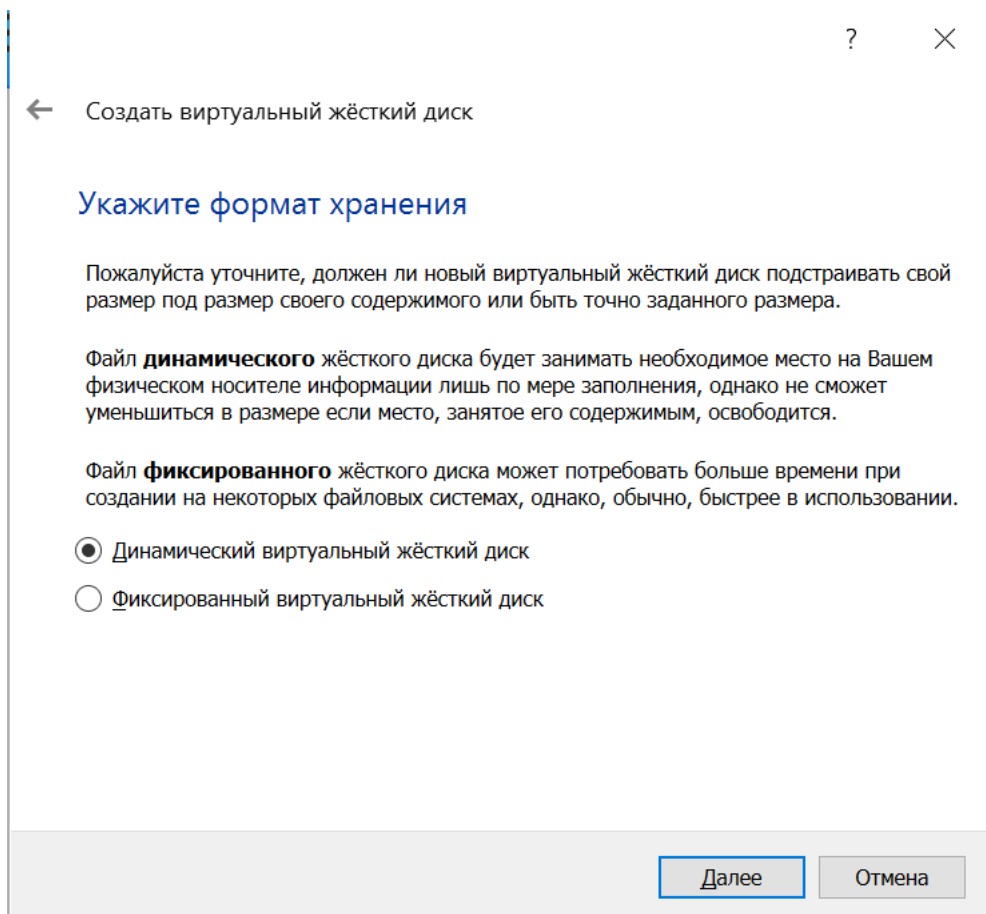


Рисунок 10

Задаем расположение и размер диска. В данном случае:

«C:\Users\Fernando\OneDrive\Рабочий стол\leonjose\leonjose\leonjose.vdi»

И 20 гв. (рисунок 11)

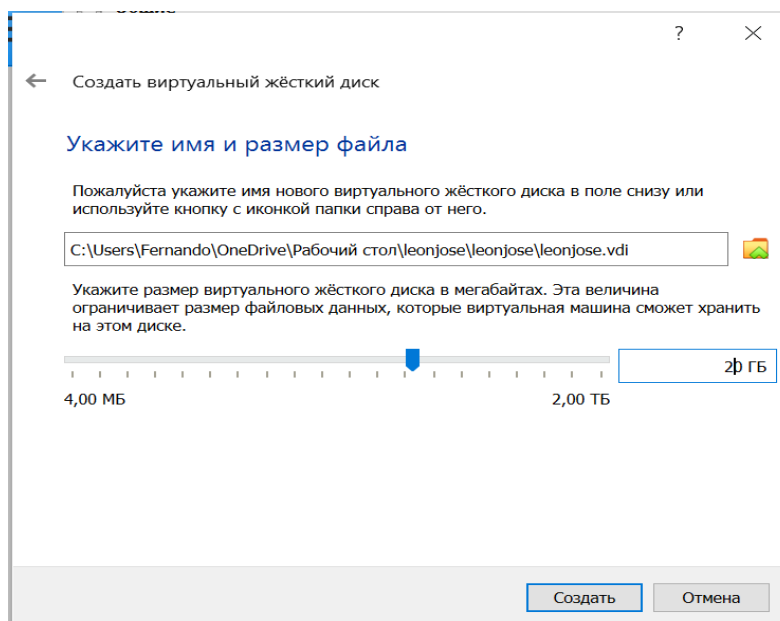


Рисунок 11

Далее нам необходимо скачать образ операционной системы. В данном

случае это « CentOS 7 x86_64 DVD 2009.iso ». Скачать можно на сайте:

https://mirror.yandex.ru/centos/7/isos/x86_64/ (рисунок 12)

Index of /centos/7/isos/x86_64/

./	06-Nov-2020 14:32	2495
0_README.txt	04-Nov-2020 11:37	4G
CentOS-7-x86_64-DVD-2009.iso	06-Nov-2020 14:44	176K
CentOS-7-x86_64-DVD-2009.torrent	02-Nov-2020 15:18	10G
CentOS-7-x86_64-Everything-2009.iso	06-Nov-2020 14:44	381K
CentOS-7-x86_64-Everything-2009.torrent	03-Nov-2020 14:55	973M
CentOS-7-x86_64-Minimal-2009.iso	06-Nov-2020 14:44	39K
CentOS-7-x86_64-Minimal-2009.torrent	26-Oct-2020 16:26	575M
CentOS-7-x86_64-NetInstall-2009.iso	06-Nov-2020 14:44	23K
CentOS-7-x86_64-NetInstall-2009.torrent	04-Nov-2020 11:38	398
sha256sum.txt	06-Nov-2020 14:37	1258
sha256sum.txt.asc		

Рисунок 12

Теперь в VirtualBox для нашей виртуальной машины выбираем «Свойства» → «Носитель». Добавляем новый природ оптических дисков и выбираем образ «CentOS 7 x86_64 DVD 2009.iso» (рисунок 13)

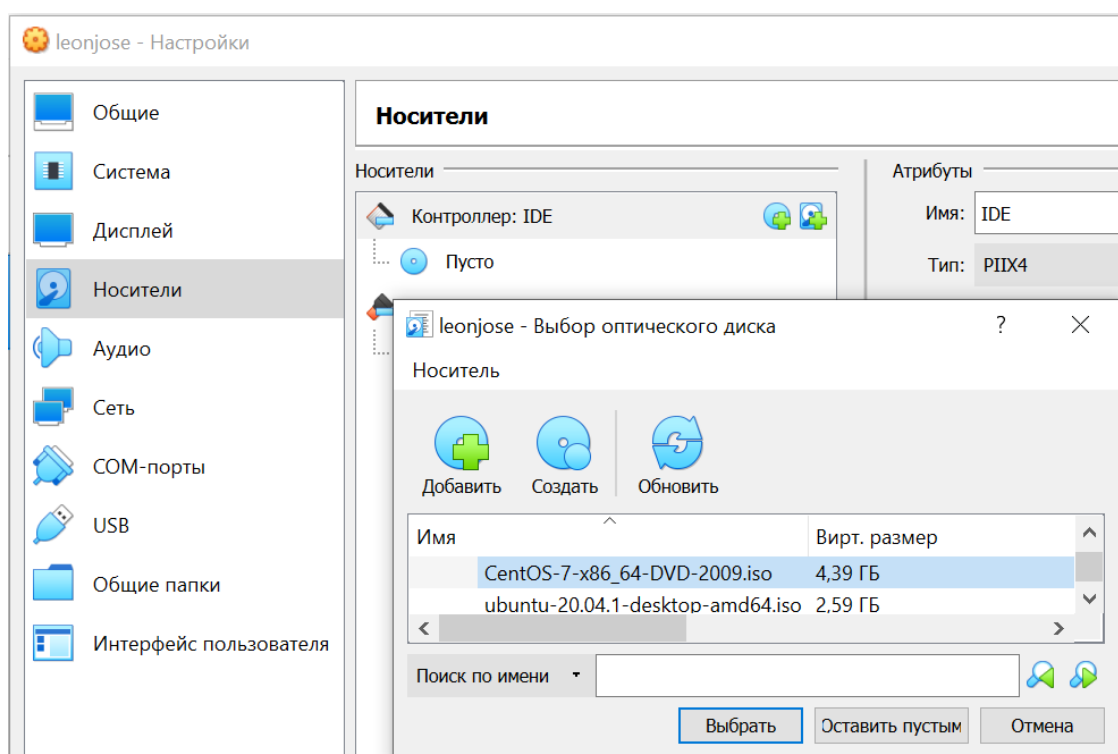


Рисунок 13

После этого необходимо запустить виртуальную машину и продолжить настройку (Рисунок 14)

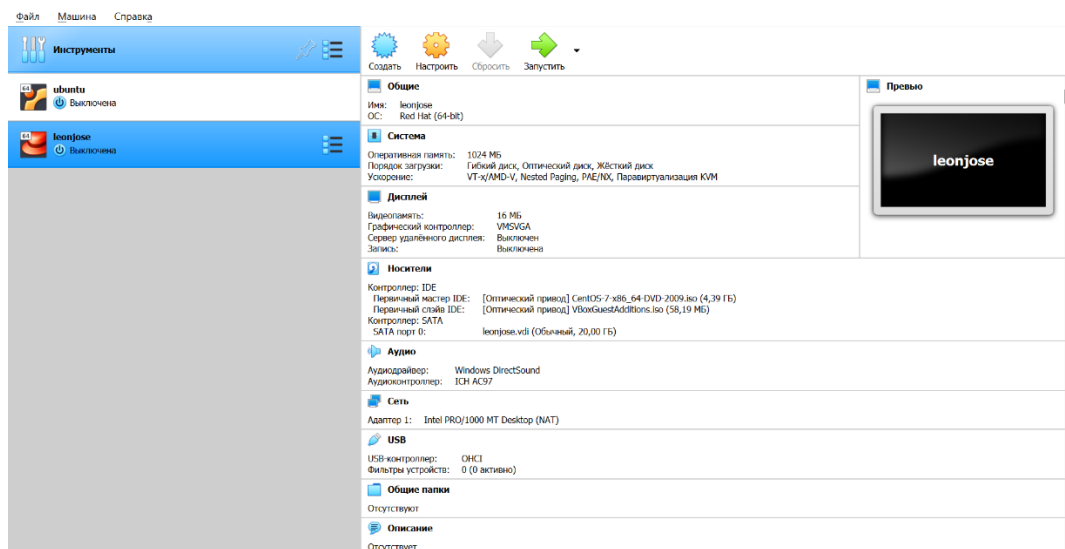


Рисунок 14

Следовательно, мы запустили и начали настраивать Centos7. Мы редактируем дату и время и добавляем имя пользователя и пароль для доступа к Centos7. (рисунок 15, 16,17,18)

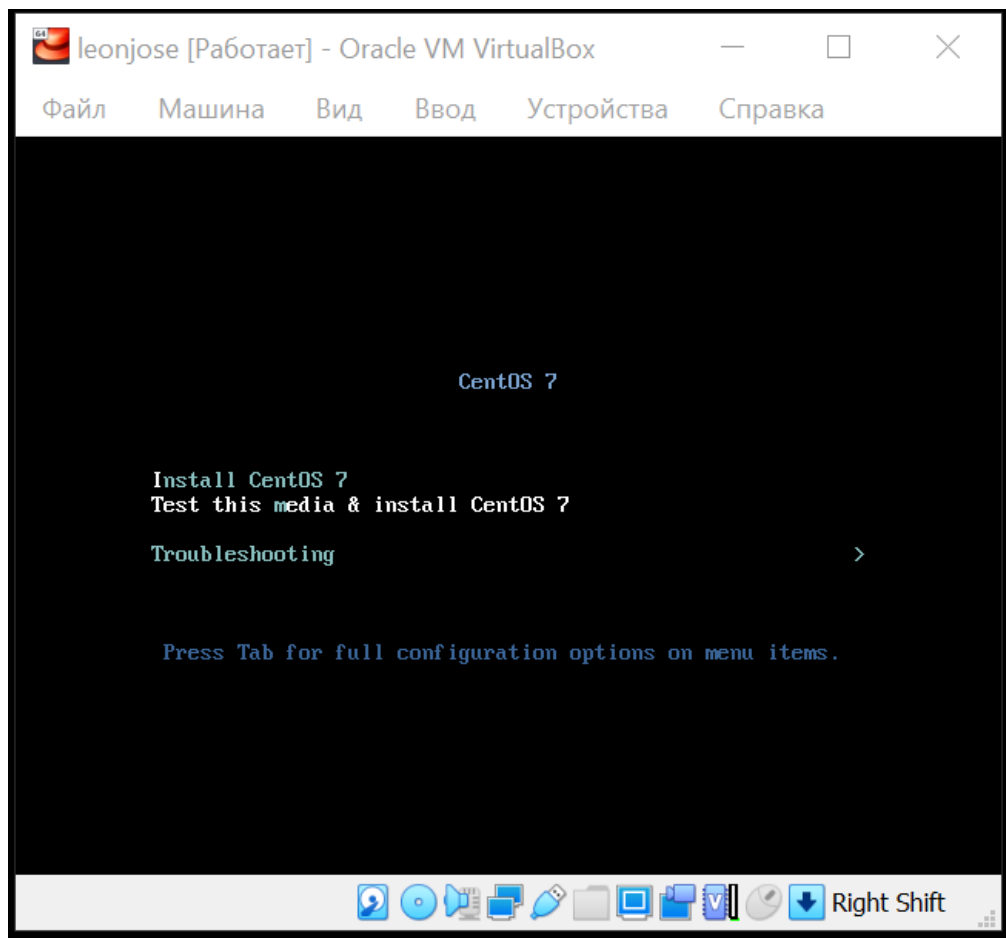


Рисунок 15

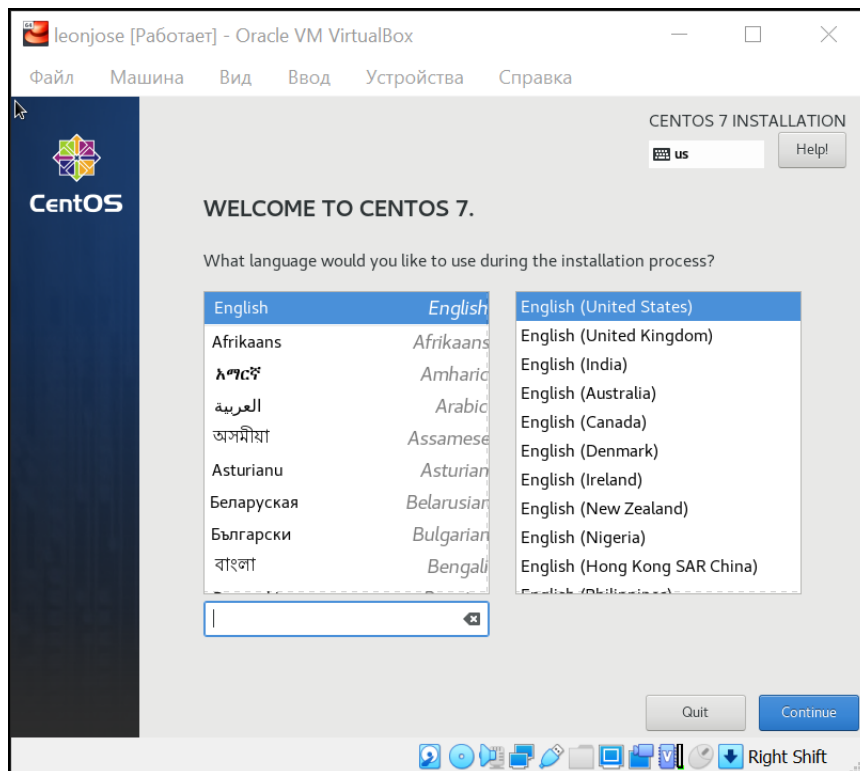


Рисунок 16

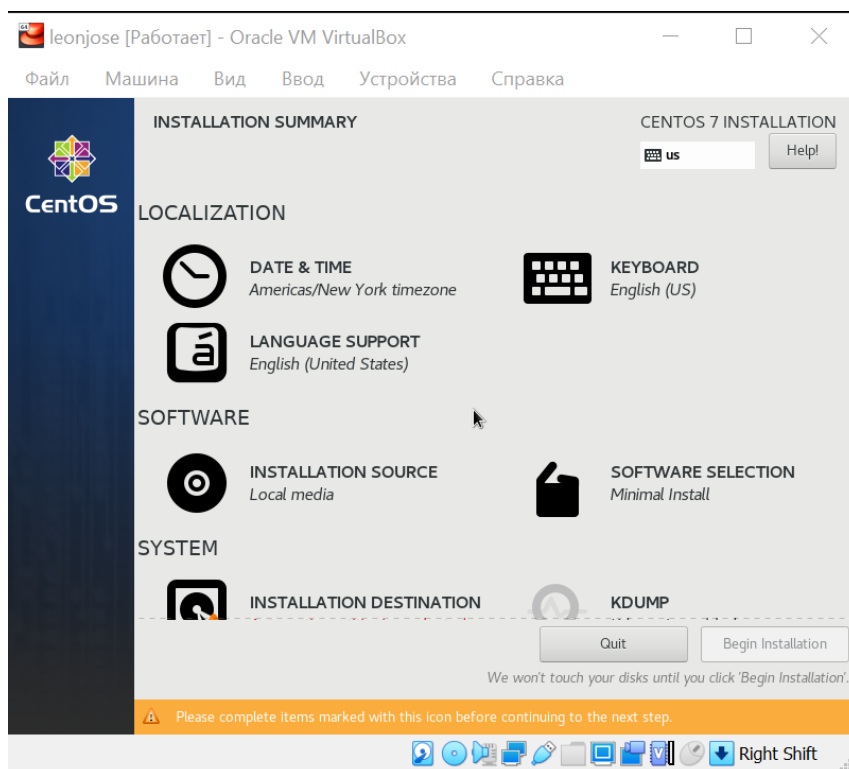


Рисунок 17

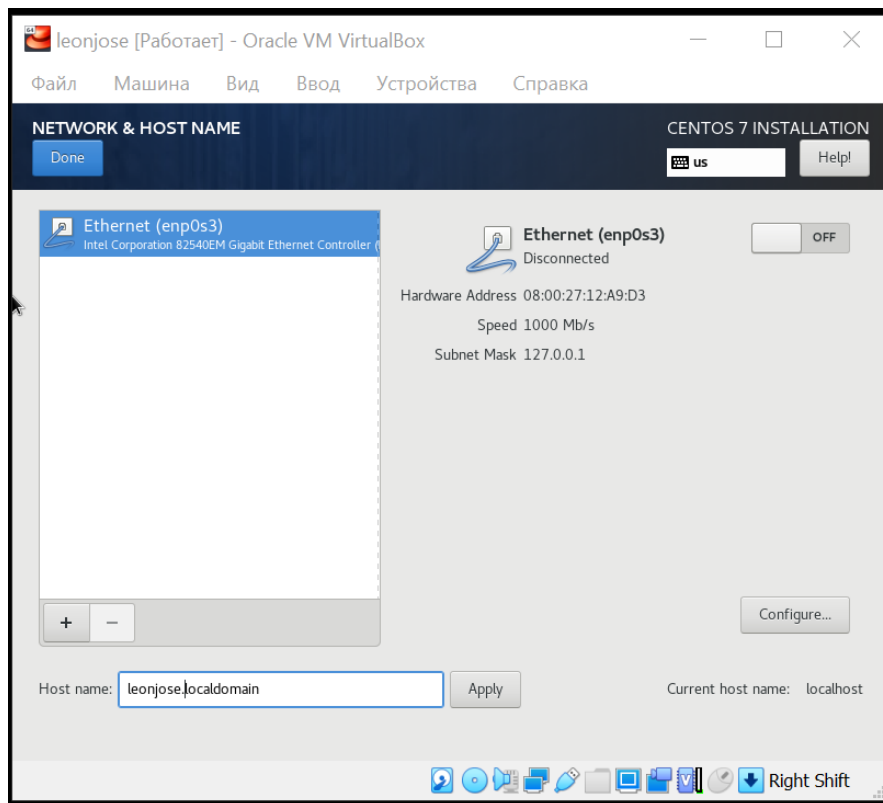


Рисунок 18

После того наш личный аккаунт начинает работать (рисунок 19 и 20).

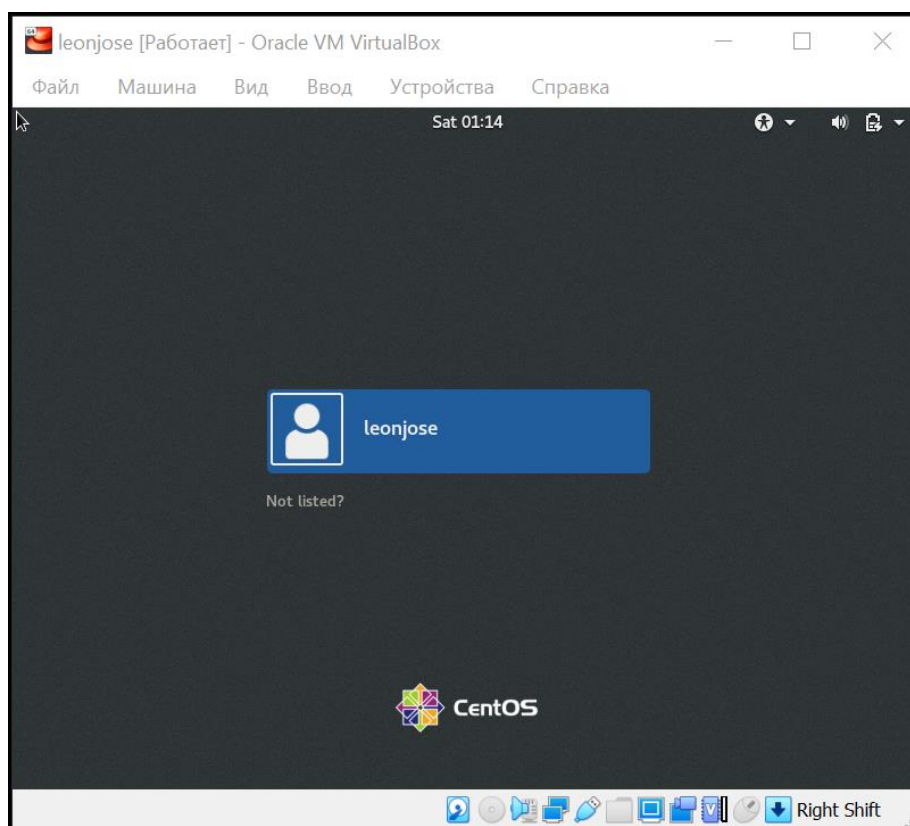


Рисунок 19

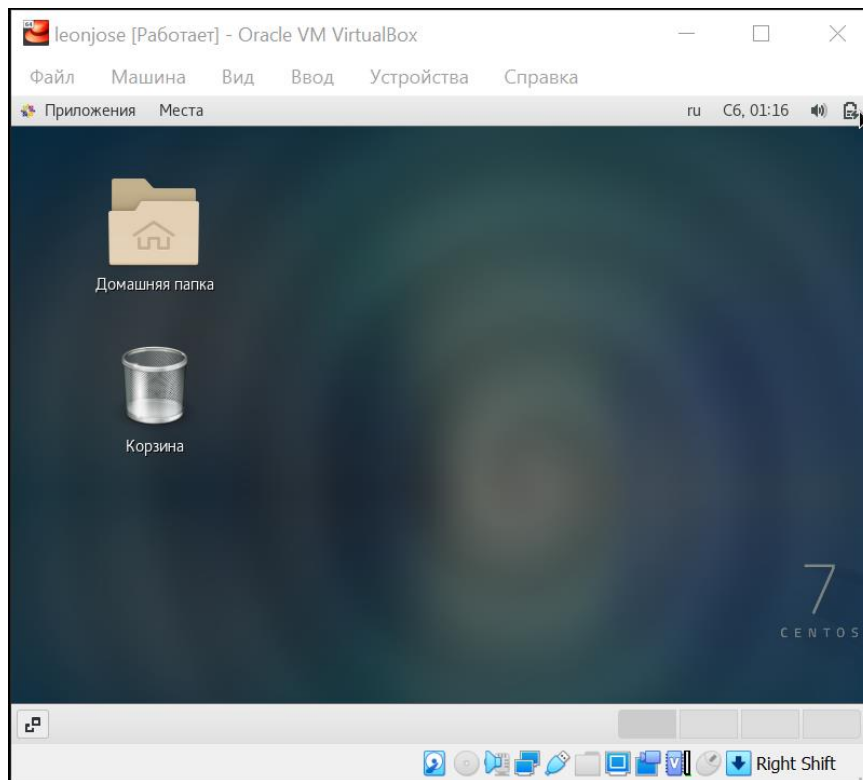


Рисунок 20

В конце мы подключаемся образ диска к ОС (рисунок 21 и 22).

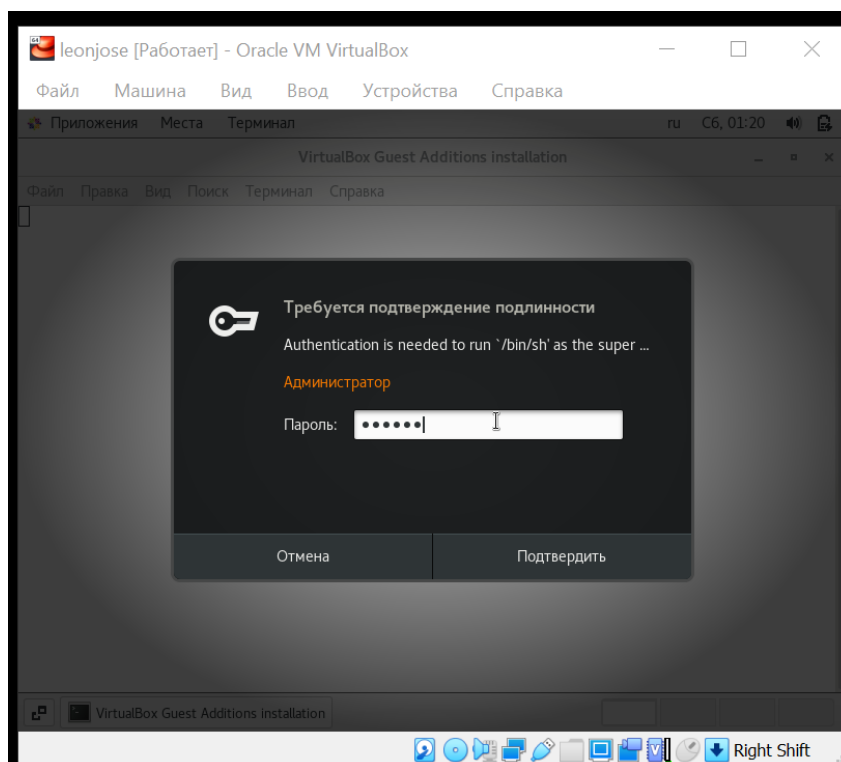


Рисунок 21

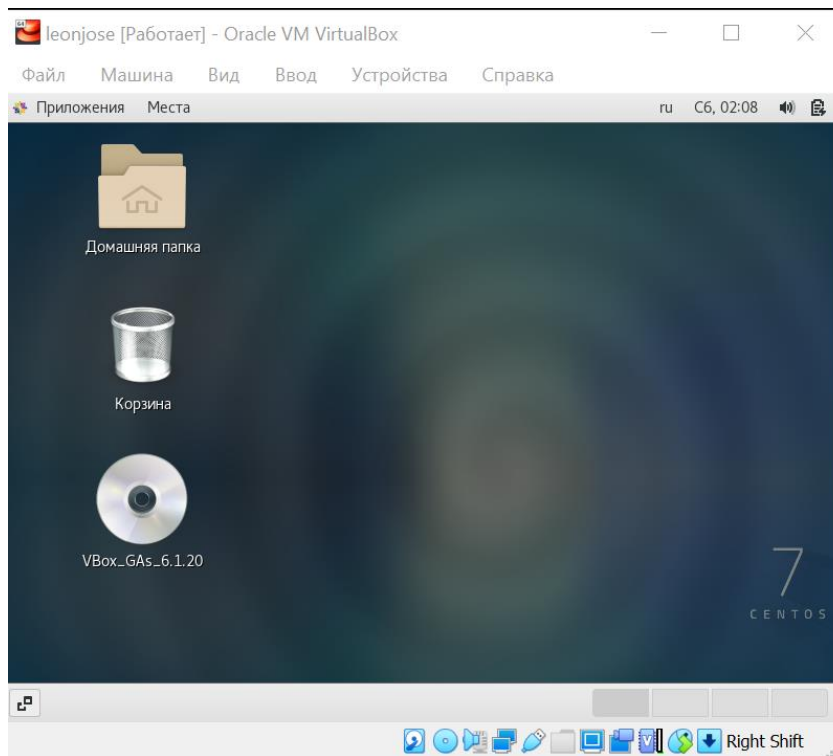


Рисунок 22

Из терминала мы можем извлечь информацию о системе. (рисунок 23)

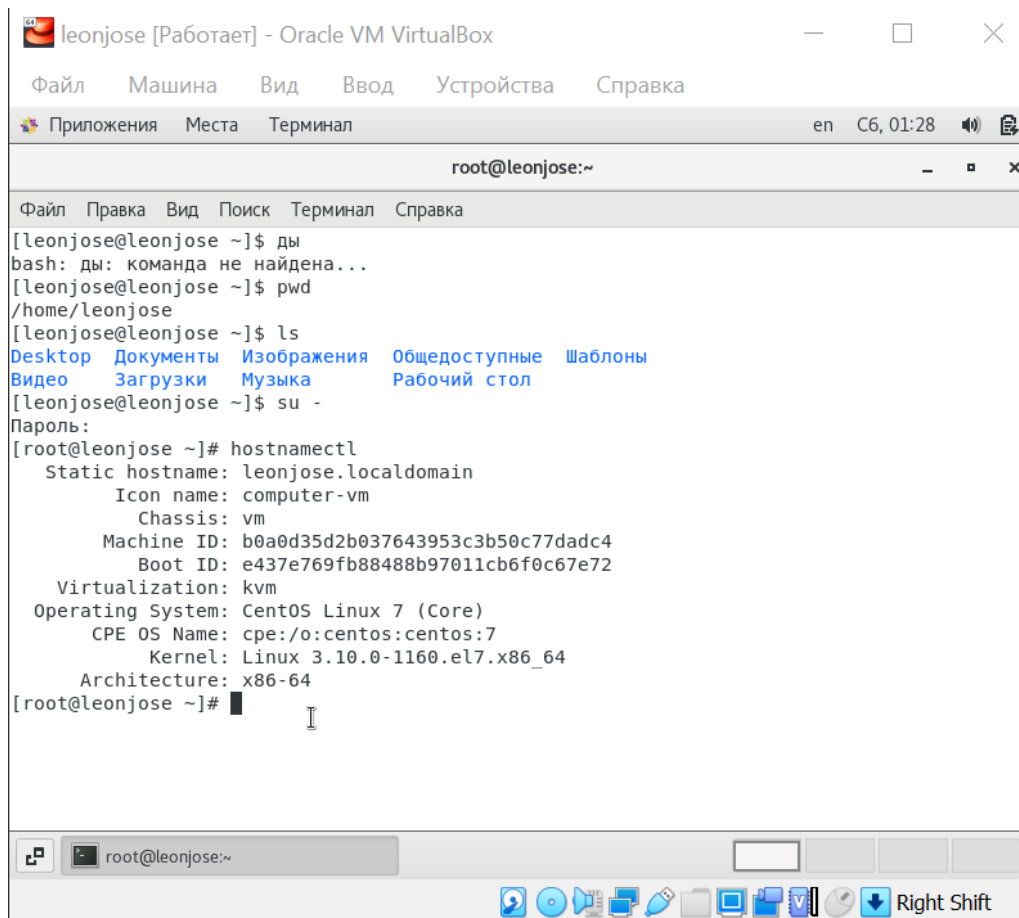


Рисунок 23

Открываем консоль и анализируем последовательность загрузки системы, используя команду «`sudo dmesg`» и введя пароль. (рисунок 24)

Домашнее задание

- 1) Версия ядра Linux: команда «`sudo dmesg | grep -i "Linux version"`» (рисунок 25)

```
[leonjose@leonjose ~]$ dmesg | grep -i "Linux version"
[ 0.000000] Linux version 3.10.0-1160.el7.x86_64 (mockbuild@kbuilder.bsys.centos.org) (gcc version 4.8.5 20150623 (Red Hat 4.8.5-44) (GCC) ) #1 SMP Mon Oct 19 16:18:59 UTC 2020
[leonjose@leonjose ~]$
```

Рисунок 25

- 2) Частота процессора : команда «`sudo dmesg | grep -i "MHz`» (рисунок 26)

```
[leonjose@leonjose ~]$ dmesg | grep -i "MHz"
[ 0.000000] tsc: Detected 2112.002 MHz processor
[ 1.523281] e1000 0000:00:03:0 eth0: (PCI:33MHz:32-bit) 08:00:27:8d:3c:5b
[ 1.633071] tsc: Refined TSC clocksource calibration: 2107.015 MHz
[leonjose@leonjose ~]$
```

Рисунок 26

- 3) Модель процессора : команда «`sudo dmesg | grep i CPU 0`» (рисунок 27)

```
[leonjose@leonjose ~]$ dmesg | grep -i "CPU0"
[ 0.131287] smpboot: CPU0: Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz (fam: 06, model: 8e, stepping: 0c)
[leonjose@leonjose ~]$
```

Рисунок 27

- 4) Объем доступной оперативной памяти : команда «`sudo dmesg | grep -i «Memory»`» (рисунок 28)

```
[leonjose@leonjose ~]$ dmesg | grep -i "Memory"
[ 0.000000] Base memory trampoline at [ffff9fc340099000] 99000 size 24576
[ 0.000000] crashkernel=auto resulted in zero bytes of reserved memory.
[ 0.000000] Early memory node ranges
[ 0.000000] PM: Registered nosave memory: [mem 0x0009f000-0x0009ffff]
[ 0.000000] PM: Registered nosave memory: [mem 0x000a0000-0x000effff]
[ 0.000000] PM: Registered nosave memory: [mem 0x000f0000-0x000fffff]
[ 0.000000] Memory: 981000k/1048512k available (7788k kernel code, 392k absent, 67120k reserved, 5954k data, 1984k init)
[ 0.000000] please try 'cgroup_disable=memory' option if you don't want memory cgroups
[ 0.042385] Initializing cgroup subsys memory
[ 0.229808] x86/mm: Memory block size: 128MB
[ 0.627525] Freeing initrd memory: 30816k freed
[ 0.654538] Non-volatile memory driver v1.3
[ 0.654646] crash memory driver: version 1.1
[ 0.714145] Freeing unused kernel memory: 1984k freed
[ 0.715000] Freeing unused kernel memory: 392k freed
[ 0.715906] Freeing unused kernel memory: 536k freed
[ 1.529219] [drm] Max dedicated hypervisor surface memory is 507904 kiB
[ 1.529220] [drm] Maximum display memory size is 16384 kiB
[ 1.529380] [TTM] Zone kernel: Available graphics memory: 507378 kiB
[leonjose@leonjose ~]$
```

Рисунок 28

- 5) Тип обнаруженного гипервизора : команда « `sudo dmesg grep -i «Hypervisor detected »(рисунок 29)`

```
[leonjose@leonjose ~]$ dmesg | grep -i "Hypervisor detected"
[ 0.000000] Hypervisor detected: KVM
[leonjose@leonjose ~]$
```

Рисунок 29

- 6) Тип файловой системы корневого раздела и последовательность монтирования файловых систем : команда « `sudo dmesg grep -i Mount »(рисунок 30)`

```
[leonjose@leonjose ~]$ dmesg | grep -i "Mount"
[ 0.042276] Mount-cache hash table entries: 2048 (order: 2, 16384 bytes)
[ 0.042278] Mountpoint-cache hash table entries: 2048 (order: 2, 16384 bytes)
[ 2.040406] XFS (dm-0): Mounting V5 Filesystem
[ 2.050008] XFS (dm-0): Ending clean mount
[ 3.316010] XFS (sda1): Mounting V5 Filesystem
[ 7.929195] XFS (sda1): Ending clean mount
[leonjose@leonjose ~]$
```

Рисунок 30

- 7) Последовательность монтирования файловых систем. Мы создаем новой точки монтирования с помощью утилиты `mkdir`. После того, как точка монтирования создана, а файловая система присутствует на разделе жесткого диска, утилита `mount` может быть использована для монтирования файловой системы в директорию, которая станет точкой монтирования. После монтирования пользователи получают доступ к файловой системе. (рисунок 31)

```
root@leonjose:~
Файл  Правка  Вид  Поиск  Терминал  Справка
[root@leonjose ~]# mkdir /home/project42
[root@leonjose ~]# mount -t ext2 /dev/sdb1 /home/project42/
mount: special device /dev/sdb1 does not exist
[root@leonjose ~]# ls
42  anaconda-ks.cfg  initial-setup-ks.cfg
[root@leonjose ~]# cat /etc/filesystems
xfs
ext4
ext3
ext2
nodev proc
nodev devpts
iso9660
vfat
rfs
rfsplus
*
[root@leonjose ~]#
```

Рисунок 31

Вывод

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

Контрольные вопросы

1. Какую информацию содержит учётная запись пользователя?

- Учетная запись пользователя содержит:
имя пользователя ()
- Идентификационный номер пользователя ()
- Идентификационный номер группы ()
- Пароль
- Полное имя
- Домашний каталог
- Начальную оболочку

2. Укажите команды терминала и приведите примеры:

– 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`»

– 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. Что такое файловая система? Приведите примеры с краткой характеристикой.

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.

Существует несколько типов файловых систем:

XFS, ReiserFS, JFS, ext, ext2, ext3, Reiser4, ext4, Btrfs, Tux2, Tux3, Xiafs, ZFS.

4. Как посмотреть, какие файловые системы подмонтированы в ОС?

Команда «`findmnt`» или «`findmnt --all`» будет отображать все подмонтированные файловые системы или искать файловую систему.

5. Как удалить зависший процесс?

Необходимо завершить дочерние процессы, удалить временные файлы, сокеты и так далее. Но в зависимости от сложности ситуации процесс может реагировать не на все сигналы. Рассмотрим основные сигналы, которые используются для завершения процесса:

SIGINT, SIGQUIT, SIGHUP, SIGTERM, SIGKILL