

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: Бабина Юлия Олеговна

Группа: НПИМбд-01-21

МОСКВА

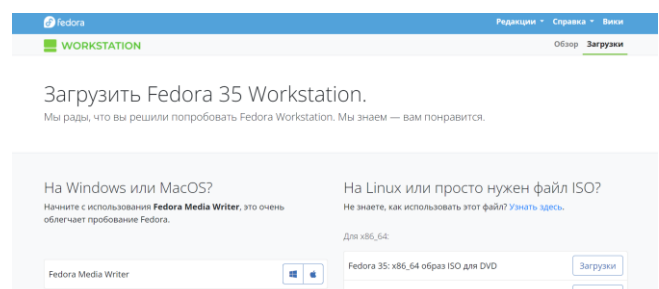
2022г.

1. Цель работы:

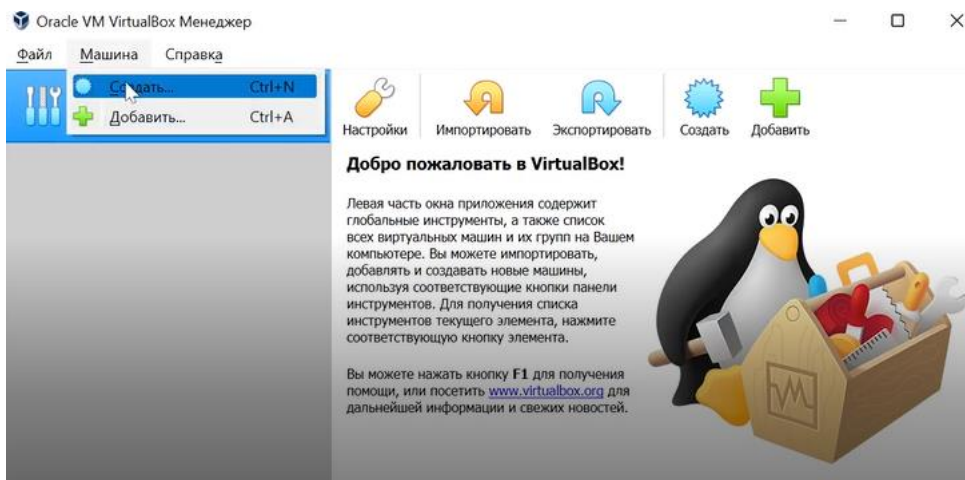
Целью данной работы является приобретение практических навыков установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов.

2. Ход работы:

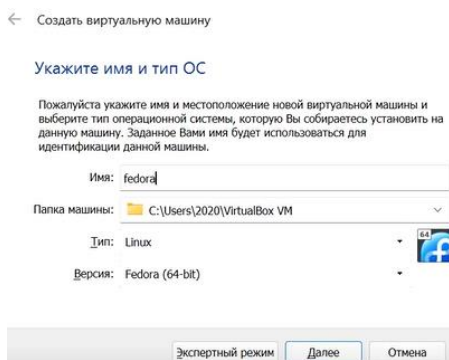
Для начала скачаем VirtualBox, необходимую для запуска виртуальных машин. Скачать можно на официальном сайте: <https://www.virtualbox.org>. Необходимо выбрать версию своей операционной системы (*Windows*). Также скачаем ISO файл дистрибутива Fedora 35 Workstation.



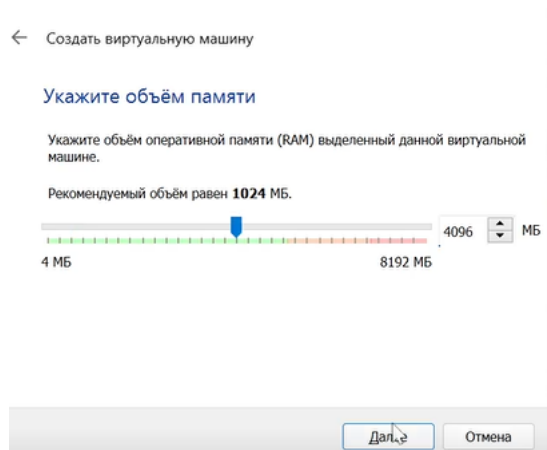
Переходим к созданию виртуальной машины. Для этого нажимаем «Машина» → «Создать».



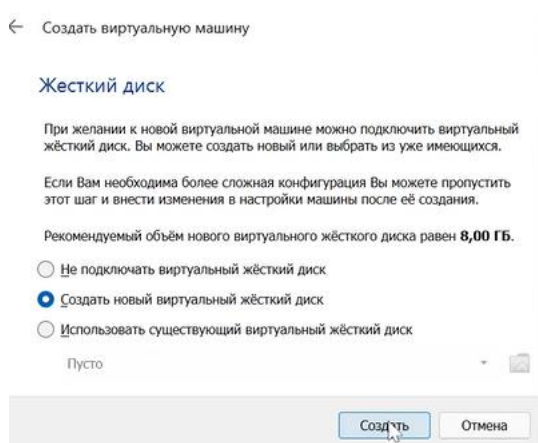
Указываем имя виртуальной машины «fedora» и тип операционной системы — Linux, Redhat (64-bit, т.к. на компьютере установлен 64 битный процессор).



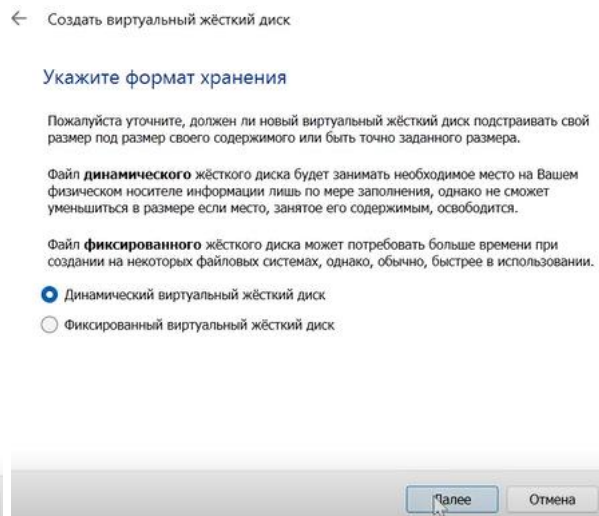
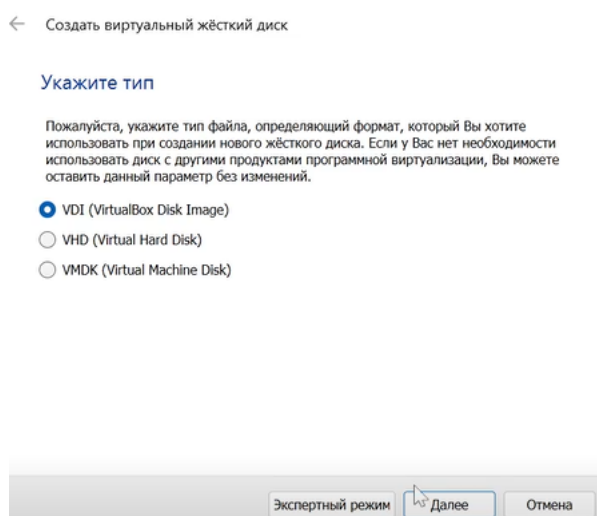
Устанавливаем размер основной памяти виртуальной машины – 4096 МБ.



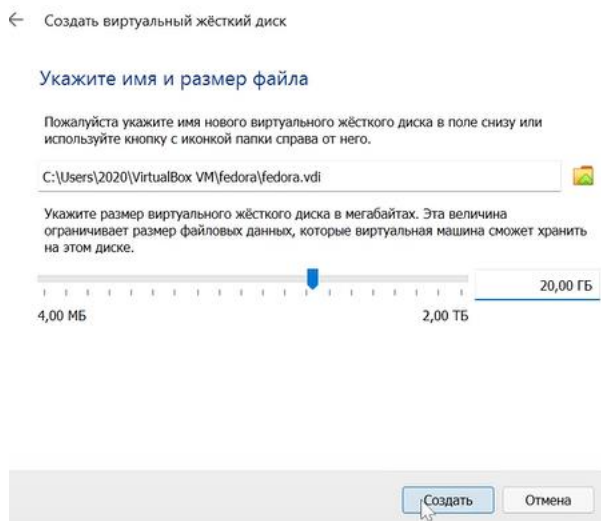
Создаем новый виртуальный жесткий диск.



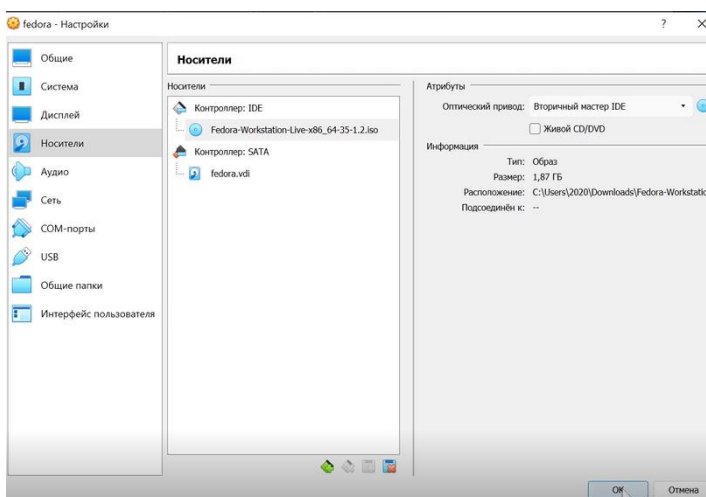
Задаем конфигурацию жесткого диска – VDI (VirtualBox Disk Image), динамический виртуальный жесткий диск.



Задаем расположение и размер диска.



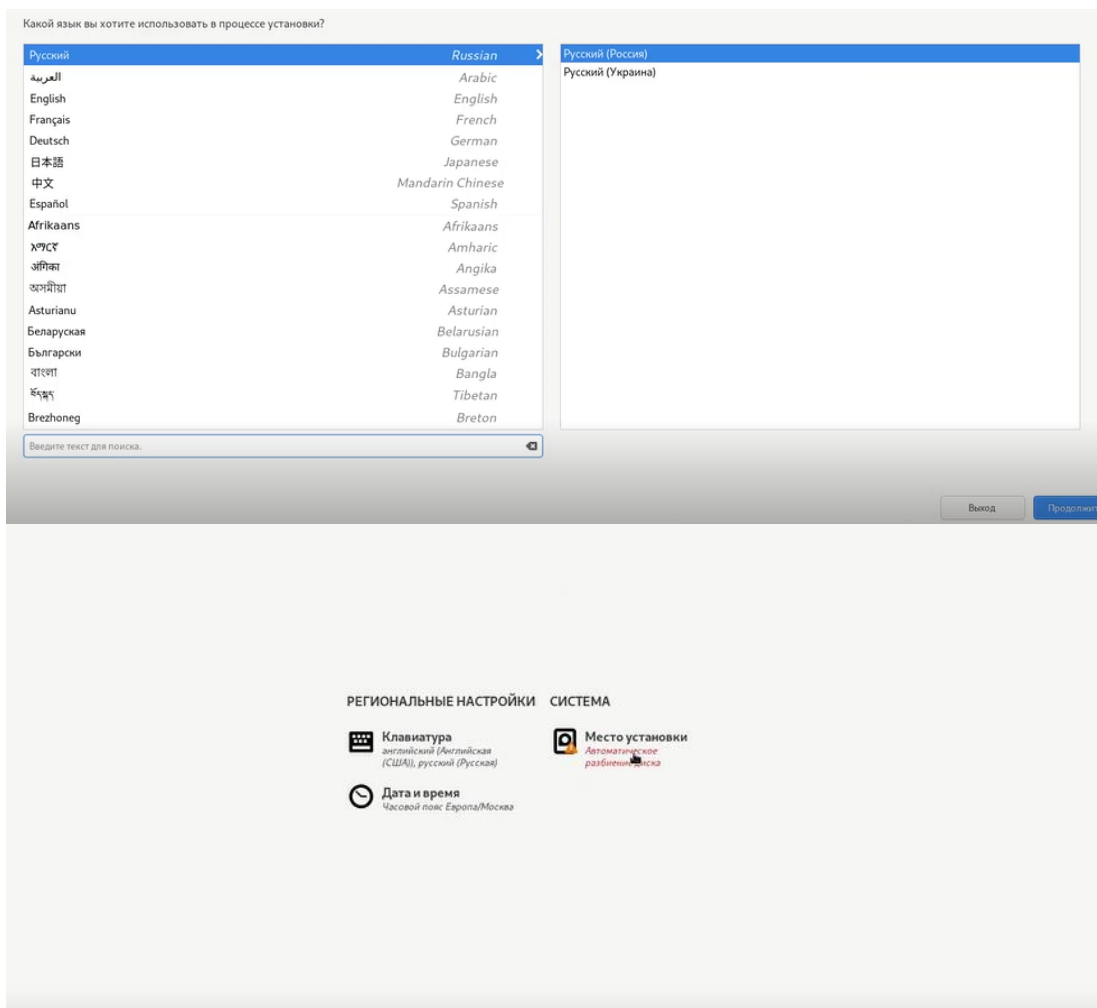
Теперь в VirtualBox для нашей виртуальной машины выбираем «Настроить» → «Носители». В пункте «Контролер IDE» выберем скачанный ISO файл.



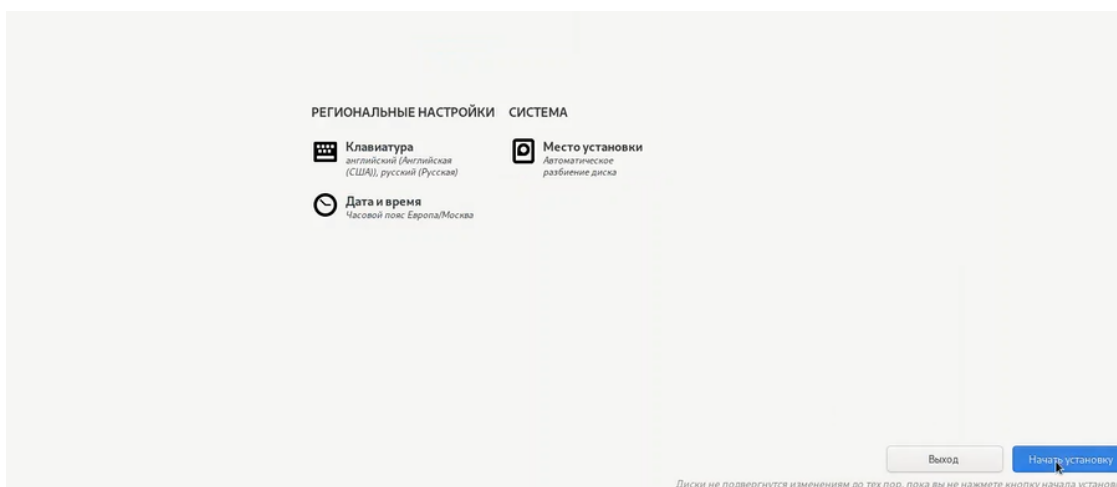
Запускаем виртуальную машину. После загрузки устанавливаем fedora на жесткий диск.



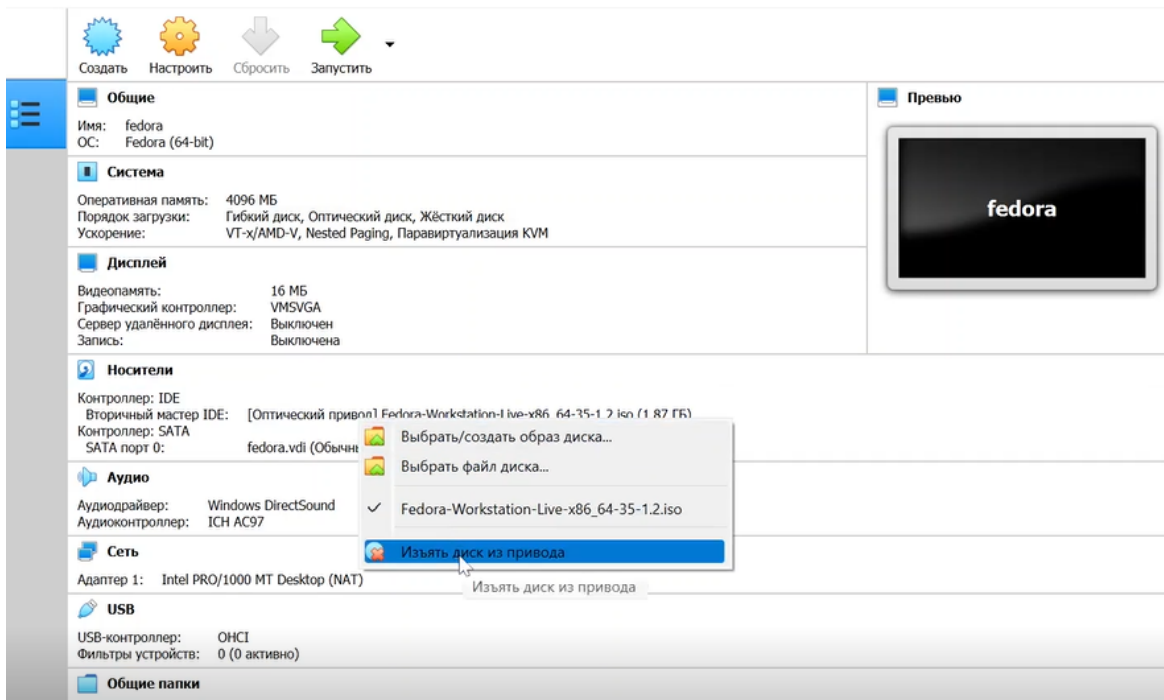
После этого выбираем язык и место установки.



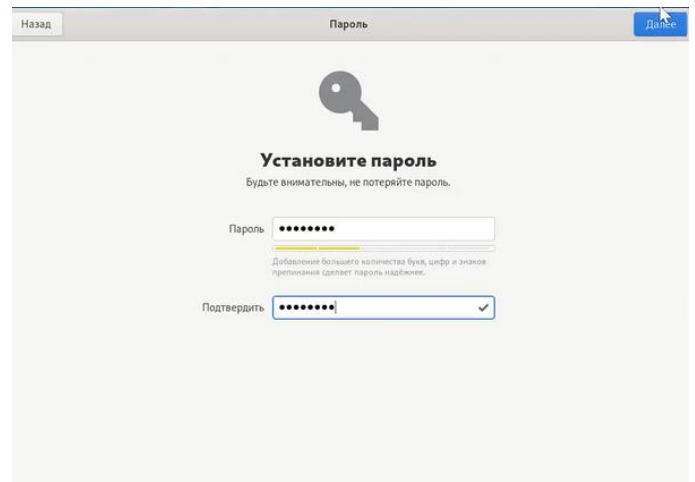
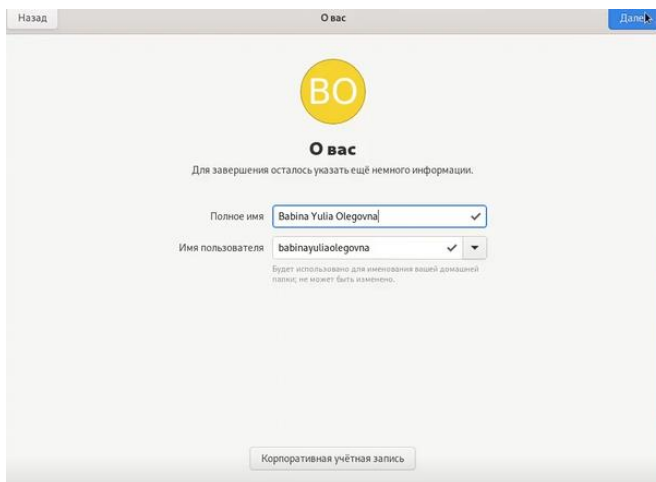
В пункте «Место установки» оставим настройки по умолчанию. Далее нажимаем на кнопку «Начать установку» и ожидаем загрузку файла.



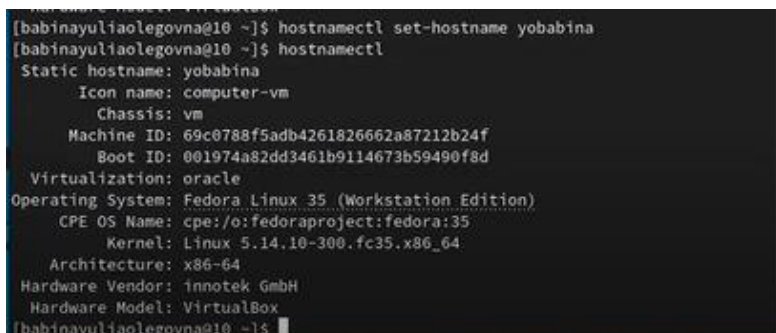
По окончании установки выключаем виртуальную машину и извлекаем диск из привода в разделе «Носители».



Запускаем виртуальную машину. Заполняем окно конфигурации пользователей. Устанавливаем пароль.



Устанавливаем имя хоста, использовав команду «hostnamectl set-hostname username». Удостоверяемся, что имя изменено, используя команду «hostnamectl».

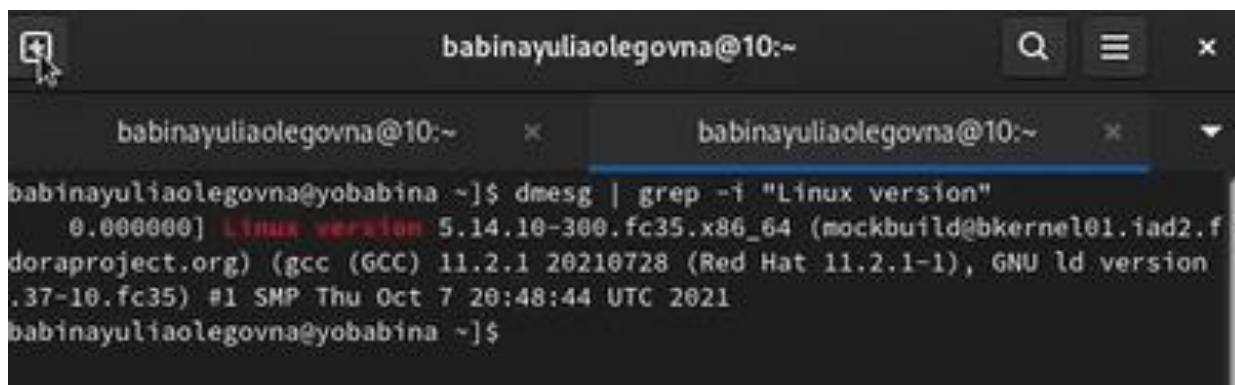


3. Домашнее задание:

В окне терминала используем команду «`dmesg | grep -i "то, что ищем"`», чтобы найти необходимую информацию.

1. Версия ядра Linux: команда «`dmesg | grep -i "Linux version"`».

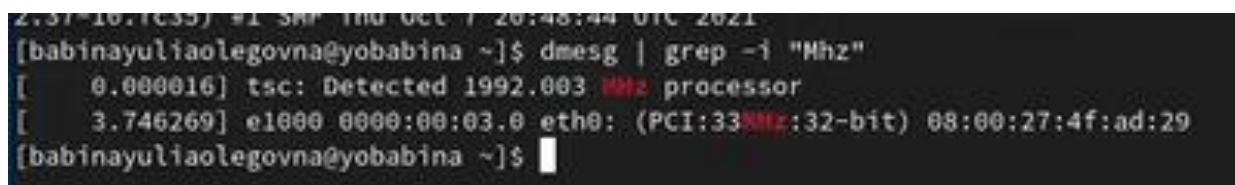
Из рисунка видно, что в данном случае версия операционной системы – 5.14.10-300.fc35.x86_64



```
babinayuliaolegovna@10:~  
babinayuliaolegovna@yobabina ~]$ 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  
babinayuliaolegovna@yobabina ~]$
```

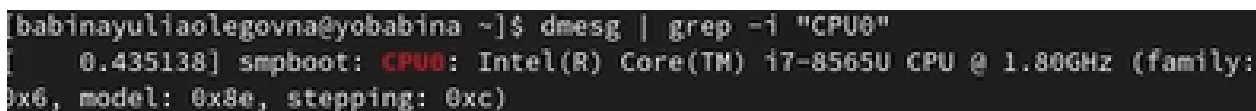
2. Частота процессора: команда «`dmesg | grep -i "Mhz"`».

Из рисунка видно, что частота процессора составляет 1992.003 МГц.



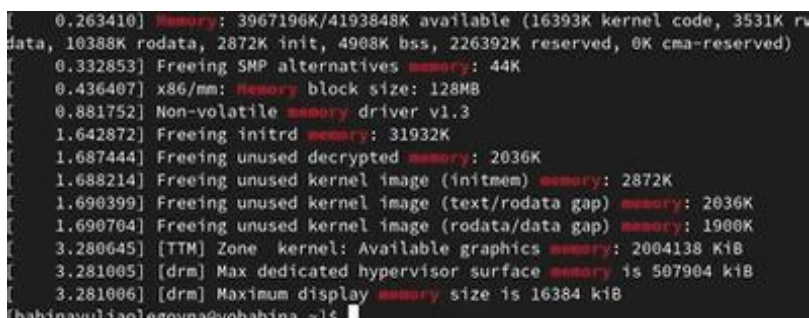
```
babinayuliaolegovna@yobabina ~]$ dmesg | grep -i "Mhz"  
[ 0.000016] tsc: Detected 1992.003 MHz processor  
[ 3.746269] e1000 0000:00:03:0 eth0: (PCI:33MHz:32-bit) 08:00:27:4f:ad:29  
babinayuliaolegovna@yobabina ~]$
```

3. Модель процессора: команда «`sdmesg | grep -i "CPU0"`». Из рисунка видно, что модель моего процессора – Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz



```
babinayuliaolegovna@yobabina ~]$ dmesg | grep -i "CPU0"  
[ 0.435138] smpboot: CPU0: Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz (family: 0x6, model: 0x8e, stepping: 0xc)
```

4. Объем доступной оперативной памяти: команда «`dmesg | grep -i "memory"`» (Рисунок 32). Из рисунка видно, что объем доступной оперативной памяти составляет 4193848 Кбайт ОЗУ.



```
[ 0.263410] Memory: 3967196K/4193848K available (16393K kernel code, 3531K r  
data, 10388K rodata, 2872K init, 4908K bss, 226392K reserved, 0K cma-reserved)  
[ 0.332853] Freeing SMP alternatives memory: 44K  
[ 0.436407] x86/mm: Memory block size: 128MB  
[ 0.881752] Non-volatile memory driver v1.3  
[ 1.642872] Freeing initrd memory: 31932K  
[ 1.687444] Freeing unused decrypted memory: 2036K  
[ 1.688214] Freeing unused kernel image (initmem) memory: 2872K  
[ 1.690399] Freeing unused kernel image (text/rodata gap) memory: 2036K  
[ 1.690704] Freeing unused kernel image (rodata/data gap) memory: 1900K  
[ 3.280645] [TTM] Zone kernel: Available graphics memory: 2004138 KiB  
[ 3.281005] [drm] Max dedicated hypervisor surface memory is 507904 KiB  
[ 3.281006] [drm] Maximum display memory size is 16384 KiB  
babinayuliaolegovna@yobabina ~]$
```


5. Тип обнаруженного гипервизора: команда «`dmesg|grep-i "hypervisor"`». Из рисунка видно, что тип данного гипервизора –KVM.

```
babinayuliaolegovna@yobabina ~]$ dmesg | grep -i "hypervisor"
0.000000] Hypervisor detected: KVM
3.281005] [drm] Max dedicated hypervisor surface memory is 507904 kiB
```

6. Тип файловой системы корневого раздела и последовательность монтирования файловых систем: команда «`dmesg grep -i «mount»`». Из рисунка видно, что тип файловой системы корневого раздела EXT 4.

```
babinayuliaolegovna@yobabina ~]$ dmesg | grep -i "mount"
0.318774] Mount-cache hash table entries: 8192 (order: 4, 65536 bytes, linear)
0.318794] Mountpoint-cache hash table entries: 8192 (order: 4, 65536 bytes, linear)
5.358232] systemd[1]: Set up automount Arbitrary Executable File Formats File System Automount Point.
5.366896] systemd[1]: Mounting Huge Pages File System...
5.368364] systemd[1]: Mounting POSIX Message Queue File System...
5.379323] systemd[1]: Mounting Kernel Debug File System...
5.381849] systemd[1]: Mounting Kernel Trace File System...
5.476438] systemd[1]: Starting Remount Root and Kernel File Systems...
5.516591] systemd[1]: Mounted Huge Pages File System.
5.517797] systemd[1]: Mounted POSIX Message Queue File System.
5.521749] systemd[1]: Mounted Kernel Debug File System.
5.530392] systemd[1]: Mounted Kernel Trace File System.
8.533021] EXT4-fs (sda1): mounted filesystem with ordered data mode. Opts: null). Quota mode: none.
babinayuliaolegovna@yobabina ~]$
```

4. Контрольные вопросы:

1. Учетная запись пользователя –это необходимая для системы информация о пользователе, хранящаяся в специальных файлах.

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:

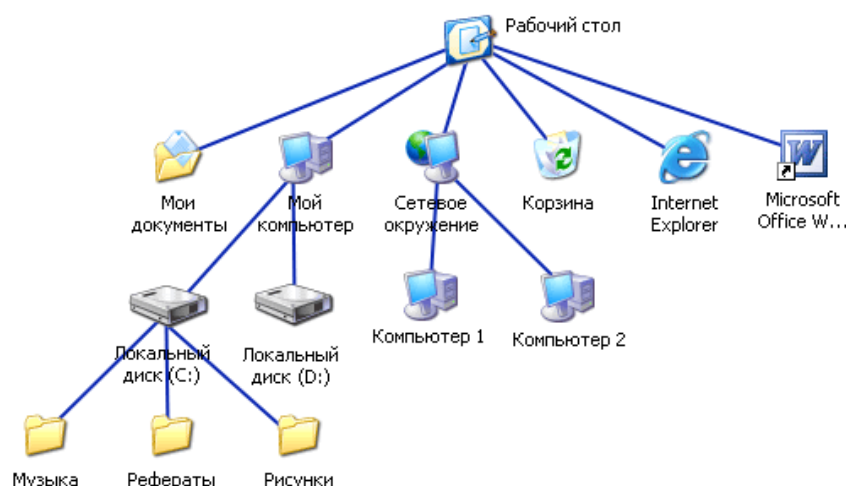
man [команда].

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 6» покажет список последних 6 команд.

3. Файловая система-порядок, определяющий способ организации, хранения и именования данных на носителях информации в компьютерах, а также в другом электронном оборудовании: цифровых фотоаппаратах, мобильных телефонах, мобильных телефонах и т.п.

Файловая система Windows



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+Q;
- **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 gtop. Команда ps предназначена для вывода списка активных процессов в системе и информации о них. Команда gtop 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 утилитами. 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.

5. 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.