

**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:

Джеффри Родригес Сантос

Группа:

НПМбд-02-20

МОСКВА

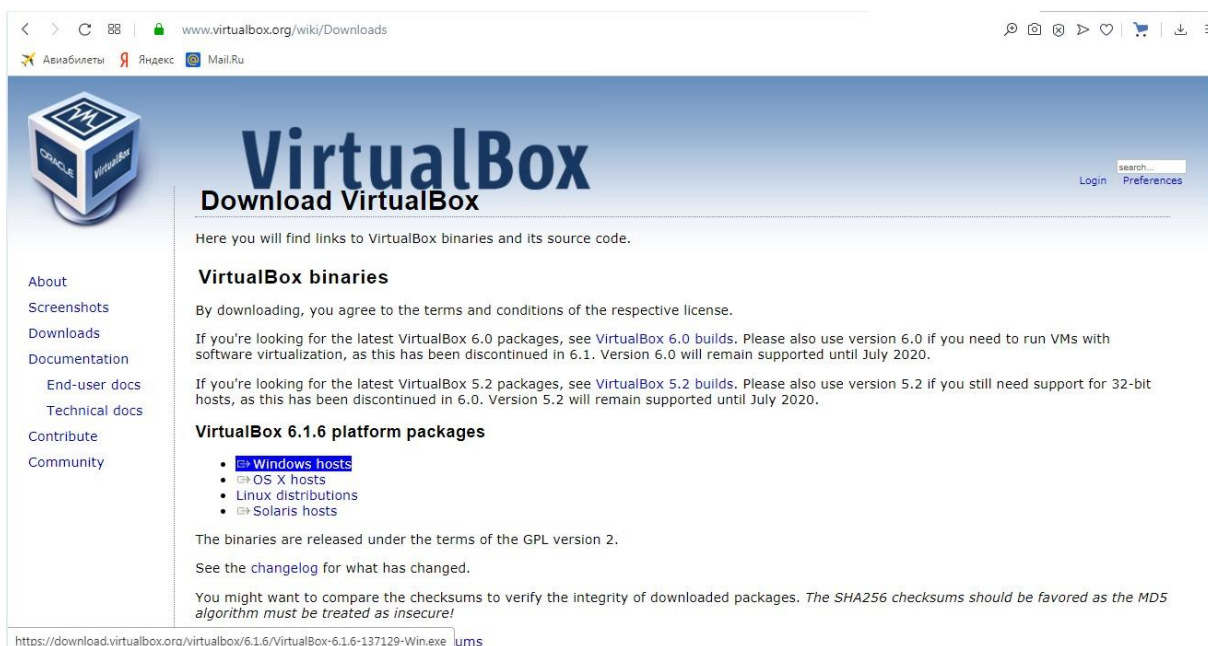
2020 г.

1. Цель работы:

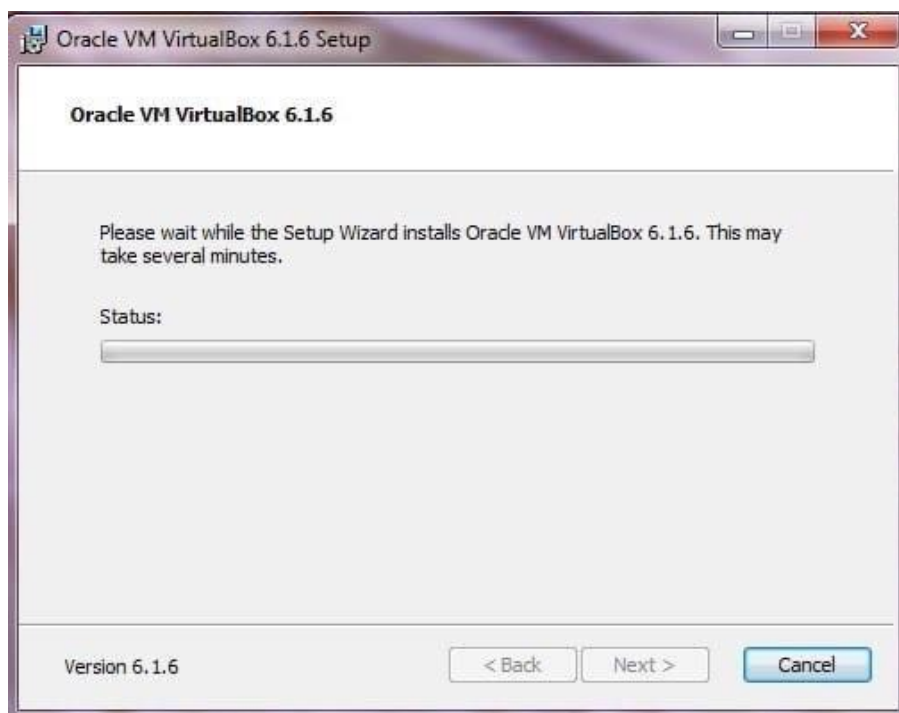
Целью данной работы является приобретение практических навыков установки системы на виртуальную машину.

2. Ход работы:

Для начала скачаем VirtualBox, предназначенную для запуска виртуальных машин. Скачать можно на официальном сайте: <https://www.virtualbox.org>. Необходимо выбрать версию своей операционной системы

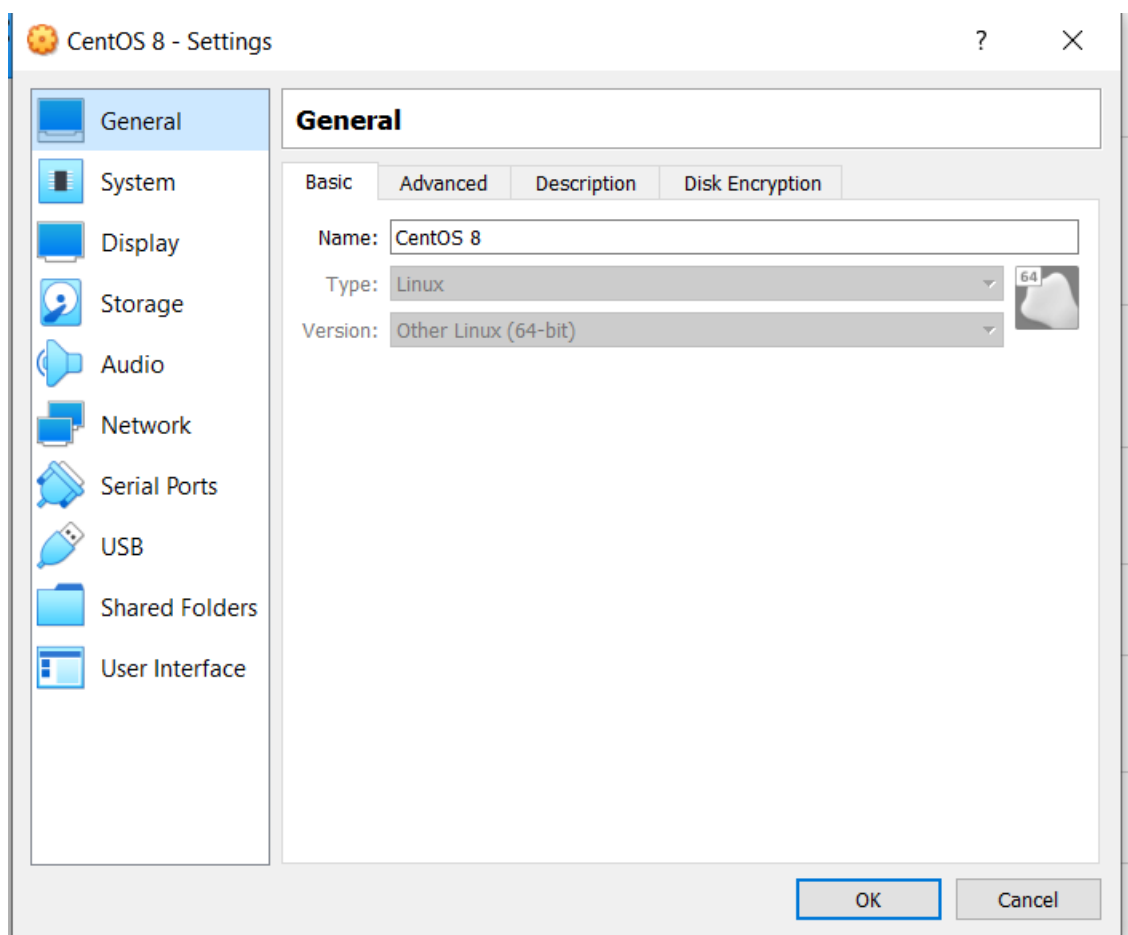


Далее выполняем установку скачанного файла.

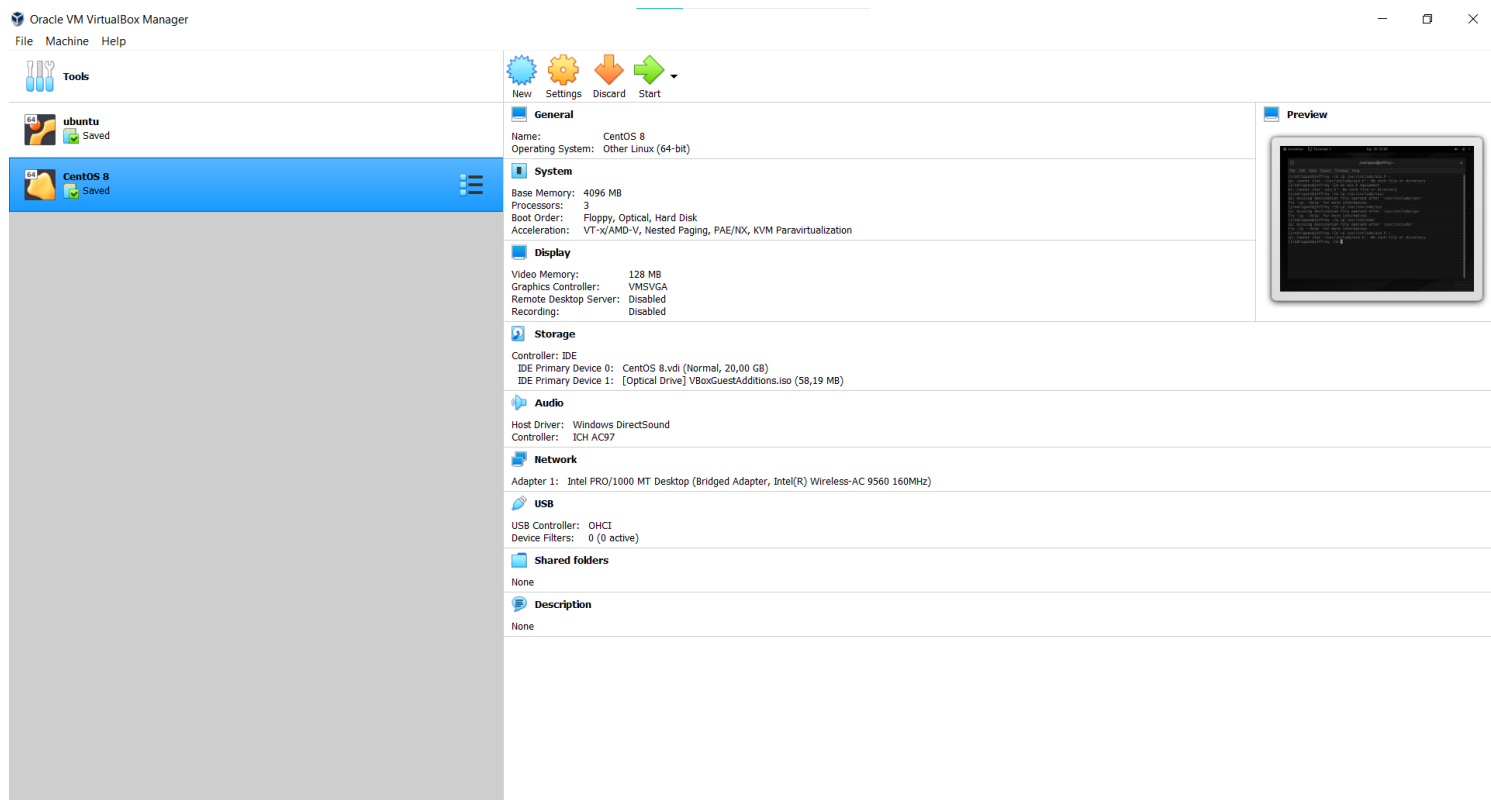


Далее создаём на рабочем столе папку, в которой будет храниться наша.

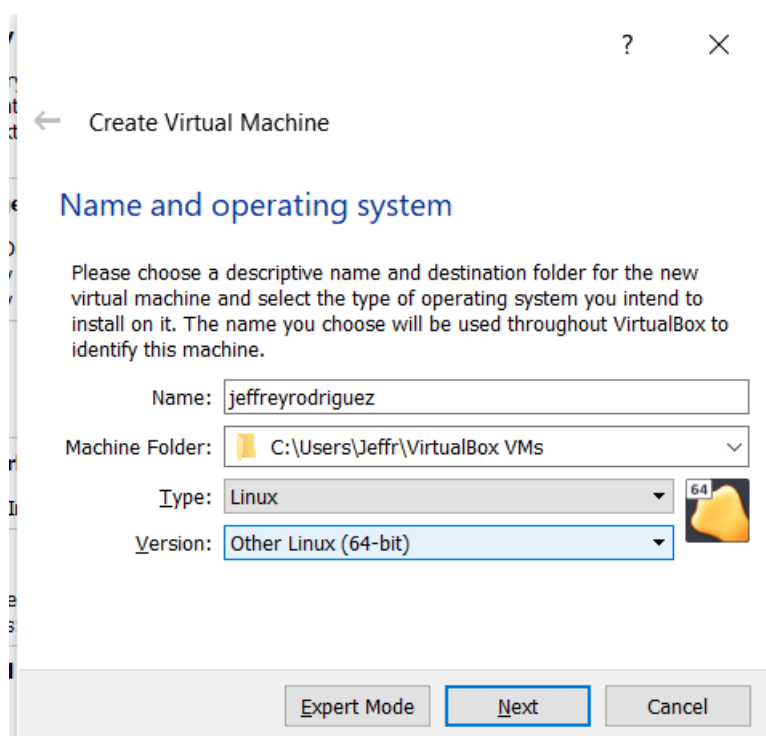
виртуальная машина. Имя папки – имя пользователя (логин студента в дисплейном классе). В данном случае «jeffreYROdriguez». Проверяем в свойствах VirtualBox месторасположение папки для виртуальных машин. Для этого открываем VirtualBox, далее «Файл» → «Свойства» → вкладка «Общие» и в поле «Папка для машин по умолчанию» указываем путь к папке, созданной ранее.



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



Указываем имя машины (логин в дисплейном классе, «jeffreyrodriguez») и тип операционной системы - Linux, other linux (64-бит, т. к. на компьютере установлен 64-битный процессор).



Укажите размер основной памяти виртуальной машины: 1024 МБ.

The screenshot shows the 'Create Virtual Machine' dialog box with the 'Memory size' step selected. The title bar has a question mark and a close button. The main heading is 'Memory size'. Below it, the text says: 'Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.' and 'The recommended memory size is 512 MB.' A horizontal slider is shown with a blue handle. The slider has a green segment on the left and a red segment on the right. The left end is labeled '4 MB' and the right end is labeled '8192 MB'. A text box to the right of the slider contains the value '1024' and is followed by 'MB'. At the bottom right, there are 'Next' and 'Cancel' buttons.

Создаём новый виртуальный жёсткий диск .

The screenshot shows the 'Create Virtual Machine' dialog box with the 'Hard disk' step selected. The title bar has a question mark and a close button. The main heading is 'Hard disk'. Below it, the text says: 'If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select one from the list or from another location using the folder icon.' and 'If you need a more complex storage set-up you can skip this step and make the changes to the machine settings once the machine is created.' and 'The recommended size of the hard disk is 8,00 GB.' There are three radio button options: 'Do not add a virtual hard disk', 'Create a virtual hard disk now' (which is selected), and 'Use an existing virtual hard disk file'. Below these options is a text box containing 'CentOS 8.vdi (Normal, 20,00 GB)' and a folder icon. At the bottom right, there are 'Create' and 'Cancel' buttons.

Задаём конфигурацию жёсткого диска - VDI (образ диска VirtualBox),

динамический виртуальный жёсткий диск.

? ×

← Create Virtual Hard Disk

Hard disk file type

Please choose the type of file that you would like to use for the new virtual hard disk. If you do not need to use it with other virtualization software you can leave this setting unchanged.

☒ VDI (VirtualBox Disk Image)

☐ VHD (Virtual Hard Disk)

☐ VMDK (Virtual Machine Disk)

Expert Mode

Next

Cancel

? ×

← Create Virtual Hard Disk

Storage on physical hard disk

Please choose whether the new virtual hard disk file should grow as it is used (dynamically allocated) or if it should be created at its maximum size (fixed size).

A **dynamically allocated** hard disk file will only use space on your physical hard disk as it fills up (up to a maximum **fixed size**), although it will not shrink again automatically when space on it is freed.

A **fixed size** hard disk file may take longer to create on some systems but is often faster to use.

☒ Dynamically allocated

☐ Fixed size

Next

Cancel

Задаём расположение и размер диска. В данном случае:
C:\Users\Jeffr\VirtualBox VMs\jeffreYROdriguez\jeffreYROdriguez.; 20 ГБ,
но рекомендуется 40 ГБ (или больше).

? ×

← Create Virtual Hard Disk

File location and size

Please type the name of the new virtual hard disk file into the box below or click on the folder icon to select a different folder to create the file in.

C:\Users\Jeffr\VirtualBox VMs\jeffreYROdriguez\jeffreYROdriguez.vdi

Select the size of the virtual hard disk in megabytes. This size is the limit on the amount of file data that a virtual machine will be able to store on the hard disk.

4,00 MB 2,00 TB 8,00 GB

Create Cancel

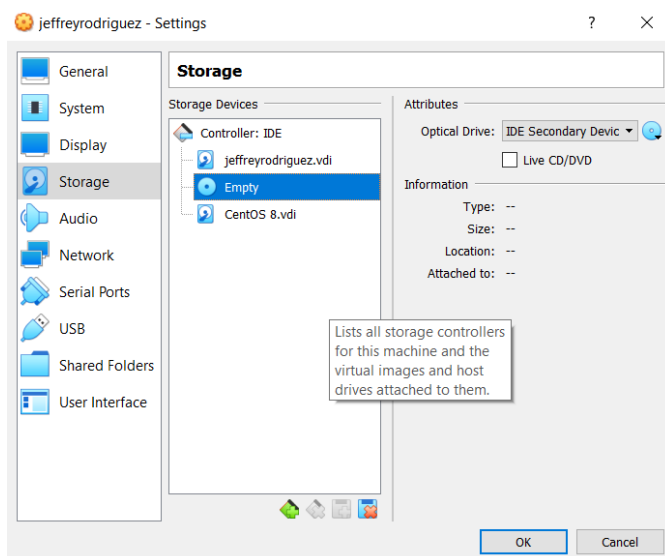
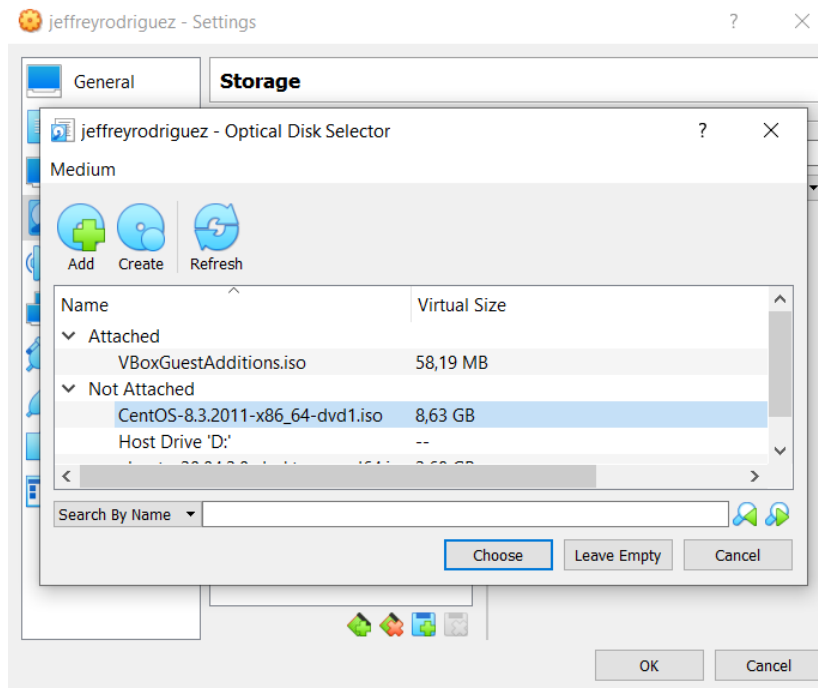
Далее нам необходимо скачать образ операционной системы. В данном случае - это "CentOS-8 -x86_64-DVD-2011.iso". Скачать можно на сайте: http://mirror.turbozoneinternet.net.br/centos/8.3.2011/isos/x86_64/

Index of /centos/8.3.2011/isos/x86_64

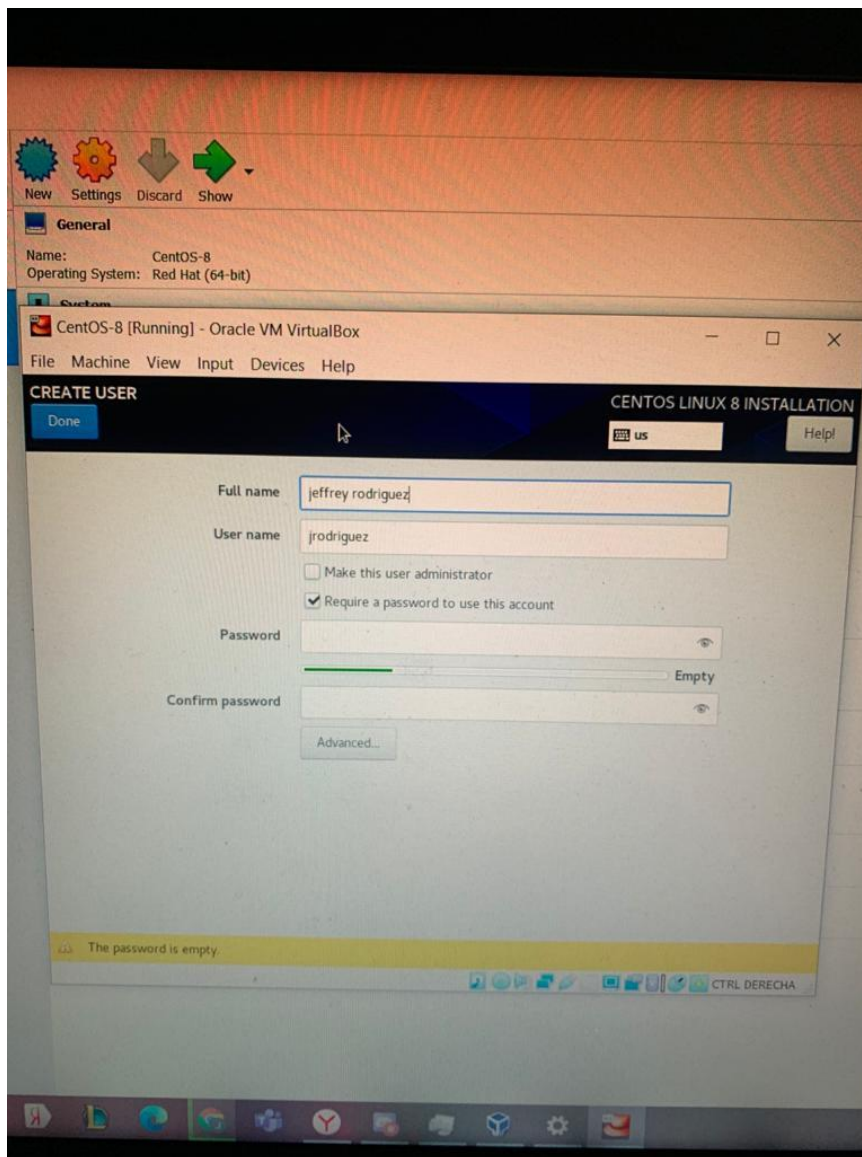
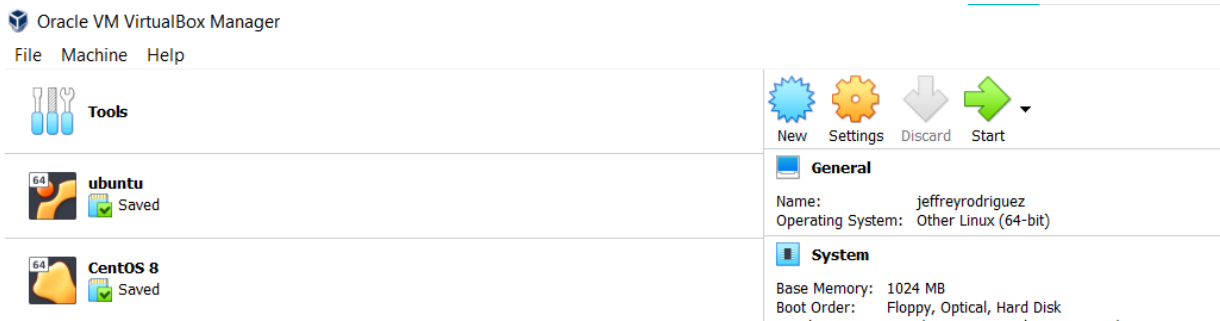
Name	Last modified	Size	Description
Parent Directory	-		
CHECKSUM	2020-12-08 10:59	485	
CHECKSUM.asc	2021-04-01 13:23	1.3K	
CentOS-8.3.2011-x86_64-DVD.iso	2020-11-18 18:01	683M	
CentOS-8.3.2011-x86_64-DVD.iso	2020-11-18 18:13	635	
CentOS-8.3.2011-x86_64-DVD.iso	2020-12-04 13:01	27K	
CentOS-8.3.2011-x86_64-DVD.iso	2020-11-18 18:43	8.6G	
CentOS-8.3.2011-x86_64-DVD.iso	2020-11-18 18:43	455K	
CentOS-8.3.2011-x86_64-DVD.iso	2020-12-04 13:01	372K	
CentOS-8.3.2011-x86_64-DVD.iso	2020-11-18 18:42	1.7G	
CentOS-8.3.2011-x86_64-DVD.iso	2020-11-18 18:42	93K	

Теперь в VirtualBox для нашей машины выбираем «Свойства» →

Носители ». Добавляем новый привод оптических дисков и выбираем образ
"CentOS-8-x86_64-DVD-2011.iso"



После этого необходимо запустить виртуальную машину продолжить настройку .



Предварительно мне пришлось создать ещё одну учётную запись "jeffreyrodriguez" и разрешить использовать "sudo" для данного пользователя (для этого добавила пользователя в файл / etc / sudoers.

```
[jrodriguez@localhost ~]$ su -l
Password:
[root@jeffrey ~]#
```

имя компьютера

```
[jrodriguez@jeffrey ~]$
```

Далее устанавливаем необходимое имя хоста, получив права администратора (команда «su -») и использовав команду «hostnamectl set-hostname нужное_имя». Удостоверяемся, что имя изменено, используя команду «hostnamectl».

```
[root@jeffrey ~]# su -
[root@jeffrey ~]# hostnamectl set-hostname jeffrey
[root@jeffrey ~]# hostnamectl
  Static hostname: jeffrey
        Icon name: computer-vm
        Chassis: vm
        Machine ID: 151764d4c7e94ca696e77aef0d05a50e
        Boot ID: 6290bd0e19a3487c8dc7a5678947a360
  Virtualization: oracle
  Operating System: CentOS Linux 8
        CPE OS Name: cpe:/o:centos:centos:8
        Kernel: Linux 4.18.0-240.el8.x86_64
  Architecture: x86_64
[root@jeffrey ~]#
```

3. Домашнее задание:

Загружаем графическое окружения и открываем консоль.

Анализируем последовательность введения системы, используя команду «sudo dmesg» и введя пароль .

```
[root@jeffrey ~]# sudo dmesg
[    0.000000] Linux version 4.18.0-240.el8.x86_64 (mockbuild@kbuilder.bsys.cent
os.org) (gcc version 8.3.1 20191121 (Red Hat 8.3.1-5) (GCC)) #1 SMP Fri Sep 25 1
9:48:47 UTC 2020
[    0.000000] Command line: BOOT_IMAGE=(hd0,msdos1)/vmlinuz-4.18.0-240.el8.x86_
64 root=/dev/mapper/cl-root ro crashkernel=auto resume=/dev/mapper/cl-swap rd.lvm
m.lv=cl/root rd.lvm.lv=cl/swap rhgb quiet
[    0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point regi
sters'
[    0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
[    0.000000] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
[    0.000000] x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]: 256
[    0.000000] x86/fpu: Enabled xstate features 0x7, context size is 832 bytes,
using 'standard' format.
[    0.000000] BIOS-provided physical RAM map:
[    0.000000] BIOS-e820: [mem 0x0000000000000000-0x00000000000009fbff] usable
[    0.000000] BIOS-e820: [mem 0x00000000000009fc00-0x00000000000009ffff] reserved
[    0.000000] BIOS-e820: [mem 0x0000000000000f0000-0x0000000000000fffff] reserved
[    0.000000] BIOS-e820: [mem 0x000000000000100000-0x0000000000000dffff] usable
[    0.000000] BIOS-e820: [mem 0x000000000dffff0000-0x000000000dffffffffff] ACPI data
[    0.000000] BIOS-e820: [mem 0x00000000fec00000-0x00000000fec000ffff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000fee00000-0x00000000fee000ffff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000fffc0000-0x00000000fffffffd] reserved
```



```

[ 0.000000] BIOS-e820: [mem 0x0000000010000000-0x0000000011ffffffff] usable
[ 0.000000] NX (Execute Disable) protection: active
[ 0.000000] SMBIOS 2.5 present.
[ 0.000000] DMI: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
[ 0.000000] Hypervisor detected: KVM
[ 0.000000] kvm-clock: Using msrs 4b564d01 and 4b564d00
[ 0.000000] kvm-clock: cpu 0, msr 6b001001, primary cpu clock
[ 0.000000] kvm-clock: using sched offset of 12595261152 cycles
[ 0.000000] clocksource: kvm-clock: mask: 0xffffffffffffffff max_cycles: 0x1d42e4dffb, max_idle_ns: 881590591483 ns
[ 0.000000] tsc: Detected 2304.000 MHz processor
[ 0.000000] e820: update [mem 0x00000000-0x00000fff] usable ==> reserved
[ 0.000000] e820: remove [mem 0x000a0000-0x000fffff] usable
[ 0.000000] last_pfn = 0x120000 max_arch_pfn = 0x400000000
[ 0.000000] MTRR default type: uncachable
[ 0.000000] MTRR variable ranges disabled:
[ 0.000000] Disabled
[ 0.000000] x86/PAT: MTRRs disabled, skipping PAT initialization too.
[ 0.000000] CPU MTRRs all blank - virtualized system.
[ 0.000000] x86/PAT: Configuration [0-7]: WB WT UC- UC WB WT UC- UC
[ 0.000000] last_pfn = 0xdfff0 max_arch_pfn = 0x400000000
[ 0.000000] found SMP MP-table at [mem 0x0009fff0-0x0009ffff]
[ 0.000000] kexec: Reserving the low 1M of memory for crashkernel

```

Просмотрим вывод этой команды, выполнив команду «`sudo dmesg | less`». В данном случае после каждого нажатия клавиши «Enter» в консоли отображается только одна команда.

```
[root@jeffrey ~]# sudo dmesg | less
```

```

[ 0.000000] Linux version 4.18.0-240.el8.x86_64 (mockbuild@kbuilder.bsys.centos.org) (gcc version 8.3.1 20191121 (Red Hat 8.3.1-5) (GCC)) #1 SMP Fri Sep 25 19:48:47 UTC 2020
[ 0.000000] Command line: BOOT_IMAGE=(hd0,msdos1)/vmlinuz-4.18.0-240.el8.x86_64 root=/dev/mapper/cl-root ro crashkernel=auto resume=/dev/mapper/cl-swap rd.lvm.lv=cl/root rd.lvm.lv=cl/swap rhgb quiet
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
[ 0.000000] x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]: 256
[ 0.000000] x86/fpu: Enabled xstate features 0x7, context size is 832 bytes, using 'standard' format.
[ 0.000000] BIOS-provided physical RAM map:
[ 0.000000] BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff] usable
[ 0.000000] BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000000f0000-0x00000000000fffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000100000-0x0000000000dfffff] usable
[ 0.000000] BIOS-e820: [mem 0x00000000dffff000-0x00000000dfffffff] ACPI data
[ 0.000000] BIOS-e820: [mem 0x00000000fec00000-0x00000000fec0ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000fee00000-0x00000000fee0ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000fffc0000-0x00000000ffffffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000100000000-0x000000011fffffffff] usable
:

```

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

1) Версия ядра Linux: команда «`sudo dmesg | grep -i "Linux version"`». Из рисунка видно, что в данном случае версия операционной системы.

```
[root@jeffrey ~]# sudo dmesg | grep -i "Linux version"
[ 0.000000] Linux version 4.18.0-240.el8.x86_64 (mockbuild@kbuilder.bsys.cent
os.org) (gcc version 8.3.1 20191121 (Red Hat 8.3.1-5) (GCC)) #1 SMP Fri Sep 25 1
9:48:47 UTC 2020
```

2) Частота процессора: команда «`sudo dmesg | grep -i «МГц»`».

Из рисунка видно, что частота процессора составляет 1995,390 МГц.

```
[root@jeffrey ~]# sudo dmesg | grep -i "MHz"
[ 0.000000] tsc: Detected 2304.000 MHz processor
[ 10.716304] e1000 0000:00:03:0 eth0: (PCI:33MHz:32-bit) 08:00:27:c9:45:d6
[root@jeffrey ~]#
```

3) Модель процессора: команда «`sudo dmesg | grep -i "CPU0"`» (Рисунок 31). Изучение моего рисунка процессора - Intel (R) Core (TM) i5- 8300NU CPU @ 2.30GHz

```
[root@jeffrey ~]# sudo dmesg | grep -i "CPU0"
[ 0.033000] smpboot: CPU0: Intel(R) Core(TM) i5-8300N CPU @ 2.30GHz (family:
0x6, model: 0x9e, stepping: 0xa)
[root@jeffrey ~]#
```

4) Объем доступной оперативной памяти: команда «`sudo dmesg | grep -i «Память»`». Из рисунка видно, что объем доступной оперативной памяти составляет 2096628 Кбайт ОЗУ.

```
[root@jeffrey ~]# sudo dmesg | grep -i "Memory"
[ 0.000000] kexec: Reserving the low 1M of memory for crashkernel
[ 0.000000] Reserving 192MB of memory at 3376MB for crashkernel (System RAM:
4095MB)
[ 0.000000] Early memory node ranges
[ 0.000000] PM: Registered nosave memory: [mem 0x00000000-0x00000fff]
[ 0.000000] PM: Registered nosave memory: [mem 0x00009f000-0x00009fff]
[ 0.000000] PM: Registered nosave memory: [mem 0x0000a0000-0x0000efff]
[ 0.000000] PM: Registered nosave memory: [mem 0x0000f0000-0x0000ffff]
[ 0.000000] PM: Registered nosave memory: [mem 0xdfff0000-0xdfffffff]
[ 0.000000] PM: Registered nosave memory: [mem 0xe0000000-0xfebfffff]
[ 0.000000] PM: Registered nosave memory: [mem 0xfec00000-0xfec00fff]
[ 0.000000] PM: Registered nosave memory: [mem 0xfec01000-0xfedfffff]
[ 0.000000] PM: Registered nosave memory: [mem 0xfef00000-0xfef00fff]
[ 0.000000] PM: Registered nosave memory: [mem 0xfef01000-0xffffbfff]
[ 0.000000] PM: Registered nosave memory: [mem 0xfffc0000-0xffffffff]
[ 0.000000] Memory: 3438480K/4193848K available (12292K kernel code, 2184K rw
data, 3960K rodata, 2428K init, 15440K bss, 429204K reserved, 0K cma-reserved)
[ 0.029461] Freeing SMP alternatives memory: 32K
[ 0.058075] x86/mm: Memory block size: 128MB
[ 1.370252] Freeing initrd memory: 54268K
[ 1.608464] Non-volatile memory driver v1.3
[ 2.314869] Freeing unused decrypted memory: 2040K
[ 2.316163] Freeing unused kernel memory: 2428K
```

5) Тип обнаруженного гипервизора: команда «`sudo dmesg | grep -i «Обнаружен гипервизор» »`» (Рисунок 33). Из рисунка видно, что тип данного гипервизора - VMware.

```
[root@jeffrey ~]# sudo dmesg | grep -i "Hypervisor detected"
[    0.000000] Hypervisor detected: KVM
```

6) Тип файловой системы корневого раздела и последовательность монтирования файловых систем: команда «`sudo dmesg | grep -i "Mount"»`» (Рисунок 34). Из рисунка видно, что тип файловой системы корневого раздела – EXT4.

```
[root@jeffrey ~]# sudo dmesg | grep -i "Mount"
[    0.026222] Mount-cache hash table entries: 8192 (order: 4, 65536 bytes)
[    0.026231] Mountpoint-cache hash table entries: 8192 (order: 4, 65536 bytes)
[   11.746894] XFS (dm-0): Mounting V5 Filesystem
[   13.407073] XFS (dm-0): Ending clean mount
[   24.543943] XFS (sda1): Mounting V5 Filesystem
[   25.943250] XFS (sda1): Ending clean mount
[root@jeffrey ~]#
```

4.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*260 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*240 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 пэббайта (32*250 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 Стивенем 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 теббайт (4*240 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 больший 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) - изначально начатый компанией Oracle проект, обеспечивающий поддержанным большинством Linux миством Linux. Ключевые особенности данной файловой системы являются технологиями: копирование при записи, позволяющая сделать снимки диска (снапшоты), которые могут пригодиться для последующего восстановления; контроль за целостностью 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 Btrfs. С августа 2008 года эта система выпускается под GNU GPL.

- Tux2 - известная, но так и не анонсированная публично файловая система. Создатель Дэниэл Филипс (Дэниел Филлипс). Система базируется на алгоритме «Фазового Древа», который как и журналы защиты файловую остубет файловую остую файловую остую мебую. Организована как надстройка на ext2.

- Tux3 - система создана на основе FUSE (Filesystem in Userspace), специального модуля для создания файловых систем на Unix плахем на Unix плахем на Unix плахорна. Данный проект ставит перед собой цель избавиться от привычного журналирования, использовать версионное восстановление (состояние в установленном промежутке времени). ?

файла создаётся изменённая копия, а не переписывается текущая версия.

- Xiafs - основа данной файловой системы, принадлежащая Фрэнку Ся, основанному на файловой системе MINIX. В настоящее время устаревшей и практически не используется. Наряду с ext2 Разработана, как замена системе доп. В декабре 1993 года была добавлена система в стандартное ядро Linux. Она оказалась слабее ext2, ведущая роли ограничения максимальных размеров файла и раздела, а так же способность к дальнейшему расширению.

- ZFS (Zettabyte File System) - изначально созданная файловая система Sun Microsystems для небезызвестной операционной системы Solaris 2005. Отличительные особенности - отсутствие фрагментации данных как таковой, возможности по управлению снапшотами (снимки), пулами хранения (пулы хранения), изменяемый размер блоков, 64-х разрядный механизм контрольных сумм, а так же способность адресовать 128 бит информации. В Linux "Моя любовь".

4) Команда «findmnt» или «findmnt --all» будет отображать все подмонтированные файловые системы или искать юусли.

5) Основные сигналы (каждый сигнал имеет свой номер), используются для завершения процесса:

- SIGINT - самый безобидный сигнал, завершение прерывания. Он отправляется, 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. Он сообщает ей что нужно завершиться и программа может выполнить корректное завершение или прерогие. В отличие от

предыдущего, она генерирует дампы памяти. Сочетание клавиш Ctrl + /;

- **SIGHUP** - сообщает процедуру, что соединение с управляющим терминалом разорвано, отправляется; соединение с управляющим терминалом разорвано, отправляется; соединение сенозерияноснозерия, осноснозерия, осноснозерия, осноснозерия
- **SIGTERM** - немедленно завершает процесс, но обрабатывается системой, поэтому позволяет процессу, но обрабатывается программой, позволяет ей завершить работу

- **SIGKILL** - тоже немедленно завершает процесс, но, в отличие от предыдущего варианта, он не передается процессу, процессу не передается, процессу не передается, процессу не передается. Поэтому ресурсы и дочерние процессы остаются запущенными.

Также для передачи сигналов процессам в Linux используется утилита `kill`, её синтаксис: `kill [-сигнал] [pid_пеноцесаципеномерипенодитакитановатановатаксиперипериподиталья - синтаксис? Сигнал представляет собой один из выше перечисленных сигналов для завершения процесса. Перед тем, выполнить остановку процесса, нужно определить его PID. Для этого используйте команды ps и grep. Команда ps для вывода списка активных процессов в системе информации о них. Команда grep запускается одновременно с ps (в канале) и будет выполнять поиск по результатам команды ps.`

Утилита `kill` - это оболочка для `kill`, она ведет себя точно так же, и имеет тот же синтаксис, только в качестве системы 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.

5. Вывод: В данной лабораторной работе я изучила, как установить операционную систему на виртуальной машине и минимально необходимые для дальнейшей работы сервисы, а также приобрела навыки поиска информации об установленной операционной системе, используя консоль.

