

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

**ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №1**

дисциплина: Операционные системы

Выполнила:

Егорова Александра

Группа: НПМбд-02-20

МОСКВА

2020 г.

Цель работы

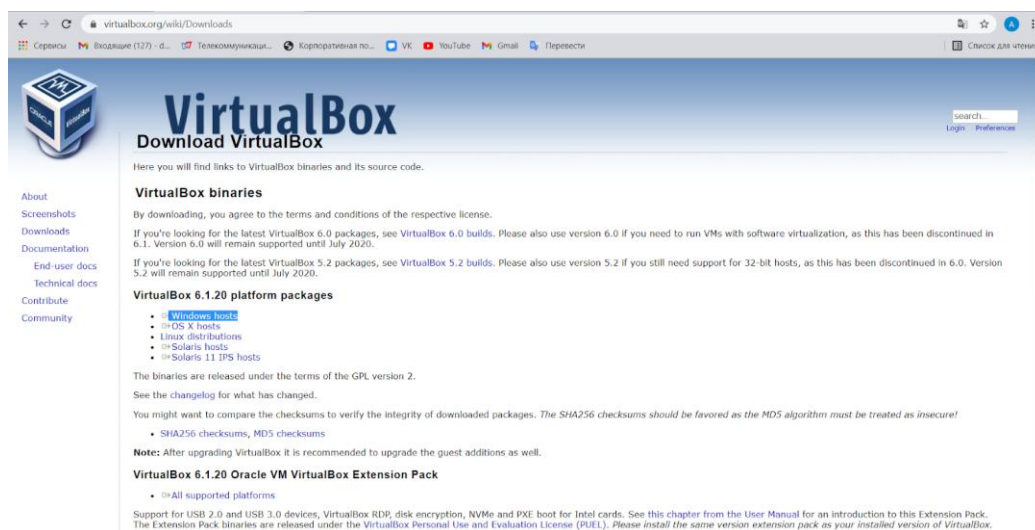
Целью данной работы является приобретение практических навыков установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов.

Ход выполнения работы

Для начала скачаем VirtualBox, необходимую для запуска виртуальных машин. Скачать можно на официальном сайте: <https://www.virtualbox.org> (Рис. 1). Необходимо выбрать версию своей операционной системы (Рис. 2).



(Рис.1)

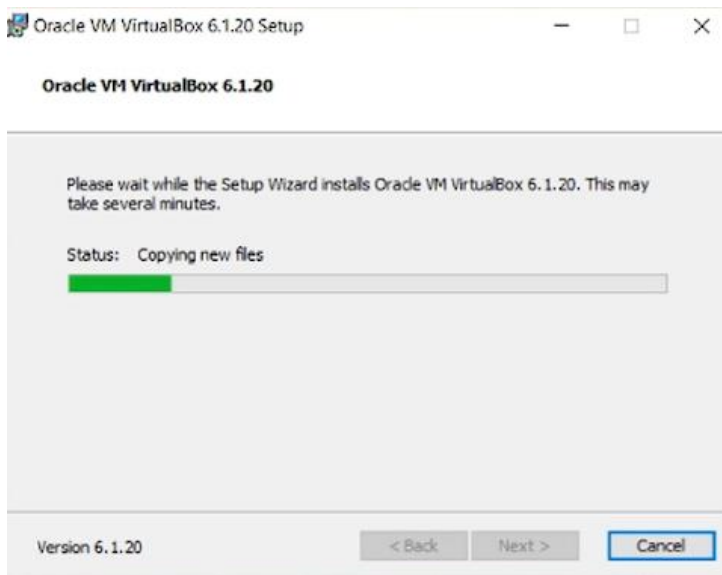


(Рис.2)

Далее выполняем установку скачанного файла

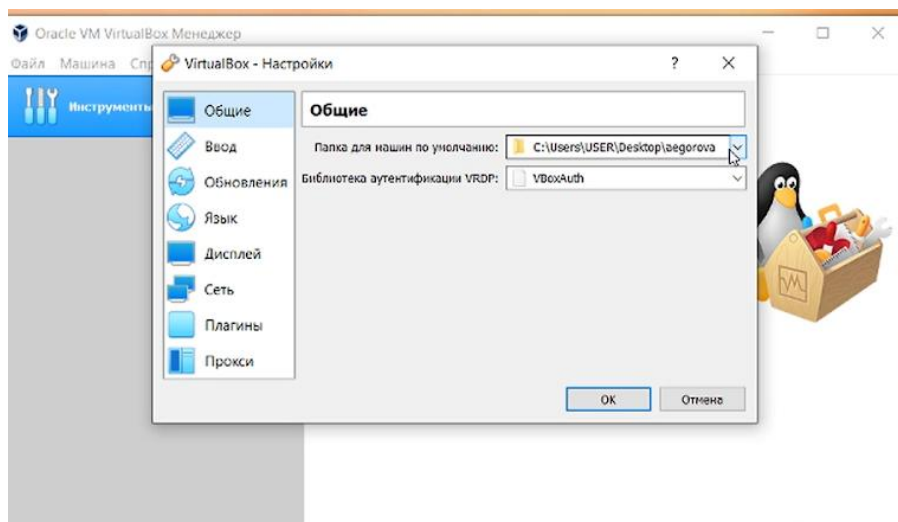


(Рис.3)



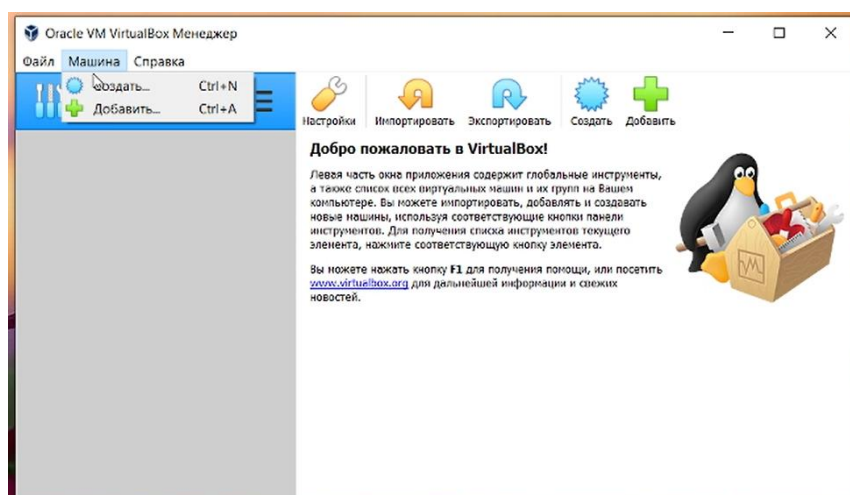
(Рис.4)

Создаём на рабочем столе папку, в которой будет храниться виртуальная машина. Имя папки – имя пользователя. В данном случае «aegorova». 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 (Рис.5)



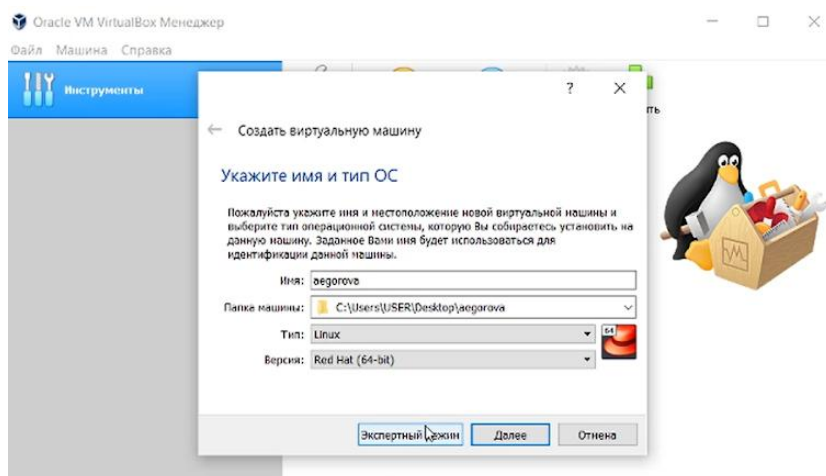
(Рис.5)

Переходим к созданию виртуальной машины. Для этого нажимаем «Машина» и «Создать» (Рис.6).



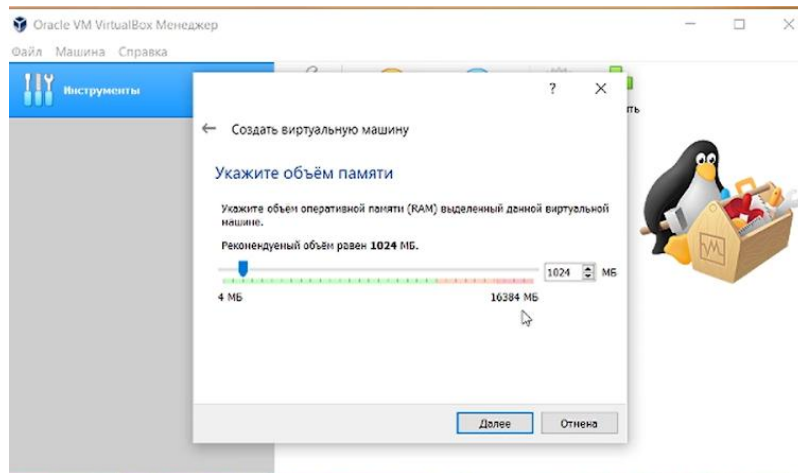
(Рис.6).

Укажем имя виртуальной машины и тип операционной системы – Linux, RedHat 64-bit (Рис.7)



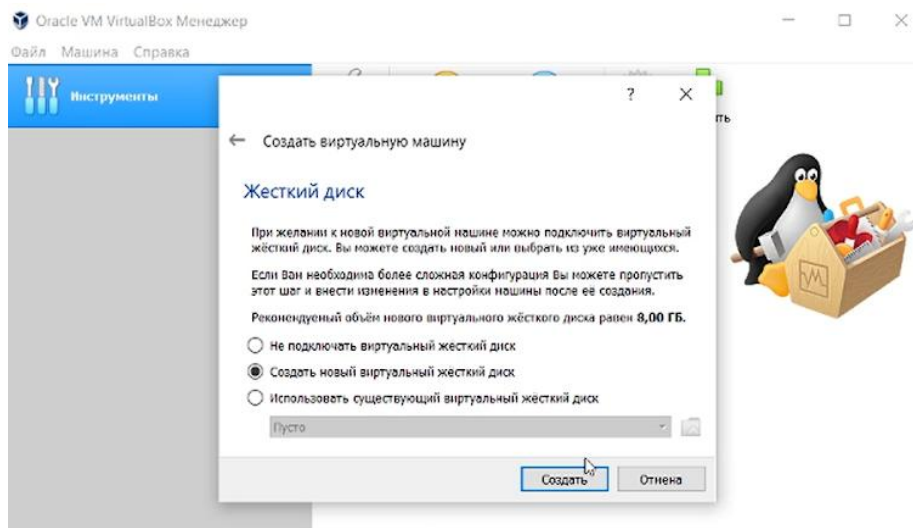
(Рис.7)

Указываем размер основной памяти виртуальной машины – 1024 МБ (Рис.8).



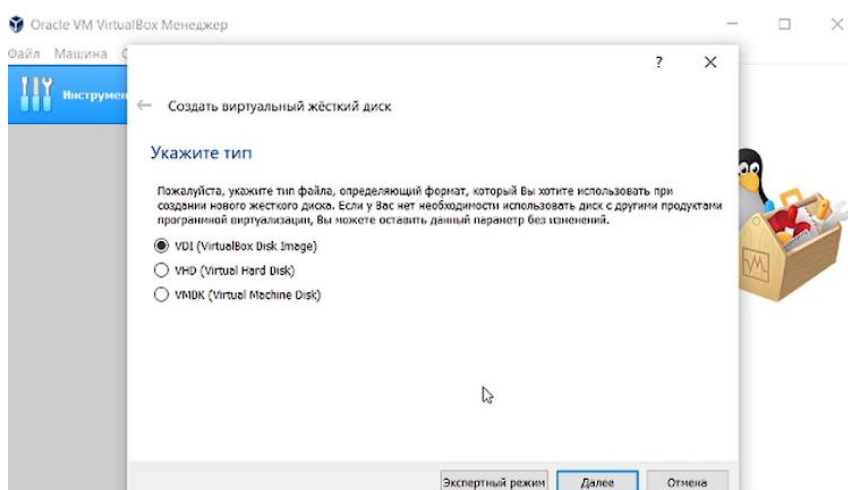
(Рис.8)

Создаём новый виртуальный жёсткий диск (Рис.9)

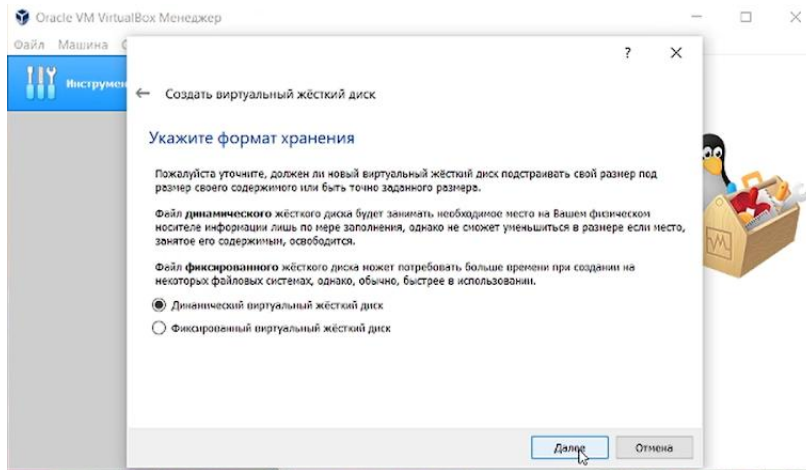


(Рис.9)

Задаём конфигурацию жёсткого диска – VDI (VirtualBox Disk Image), динамический виртуальный жёсткий диск (Рис.10, 11).

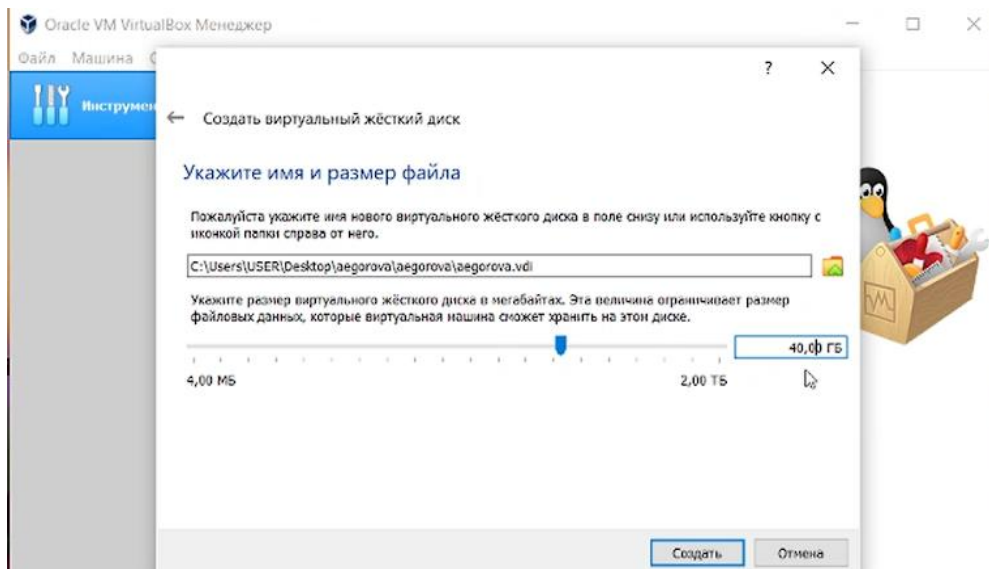


(Рис.10)



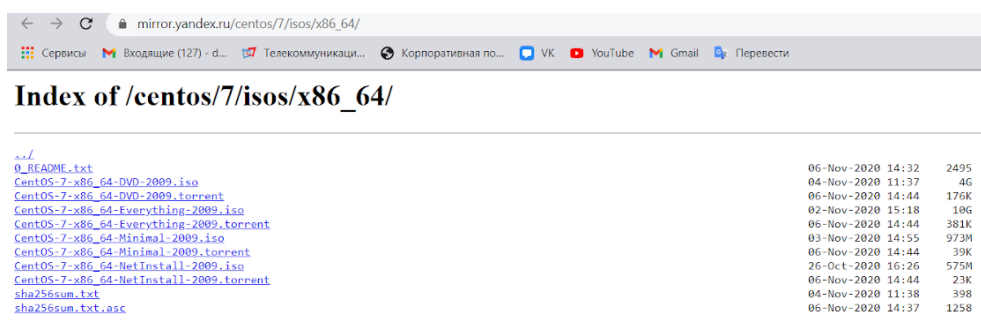
(Рис.11)

Задаём расположение и размер диска (Рис.12).



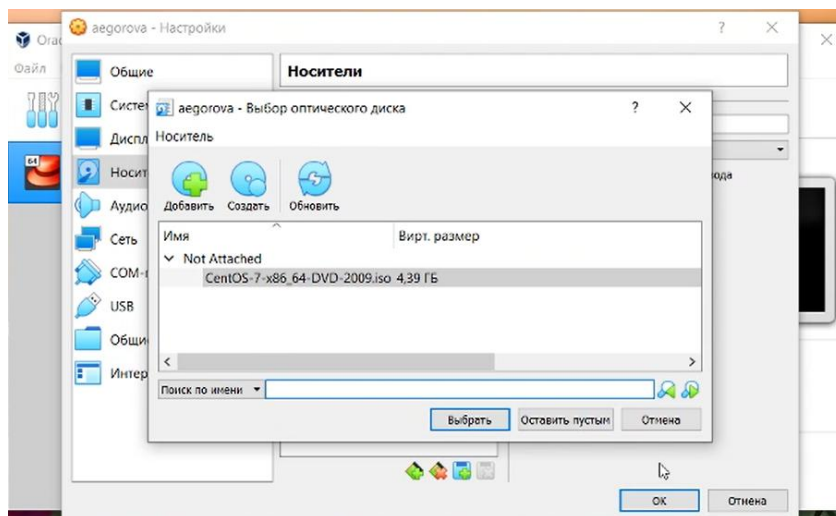
(Рис.12)

Далее 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/ (Рис.13).

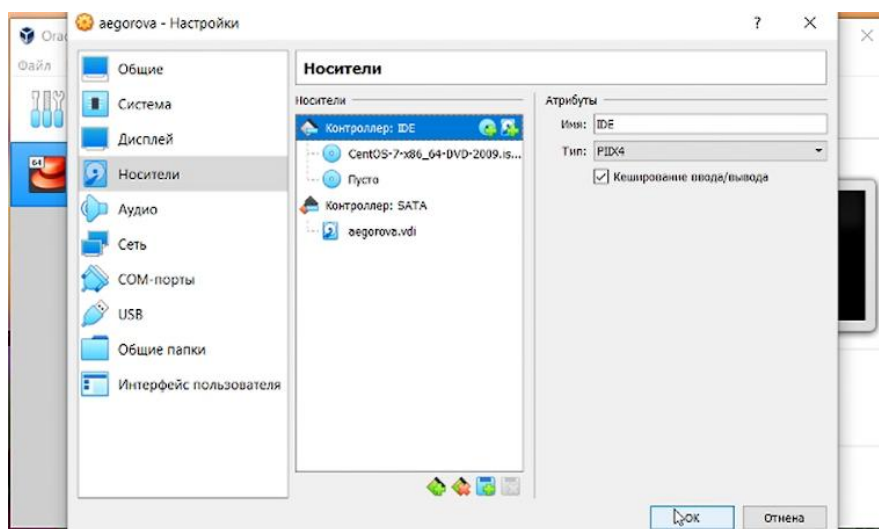


(Рис.13)

Теперь в VirtualBox для виртуальной машины выбираем «Свойства» и «Носители». Добавляем новый привод оптических дисков (Рис.14, 15).

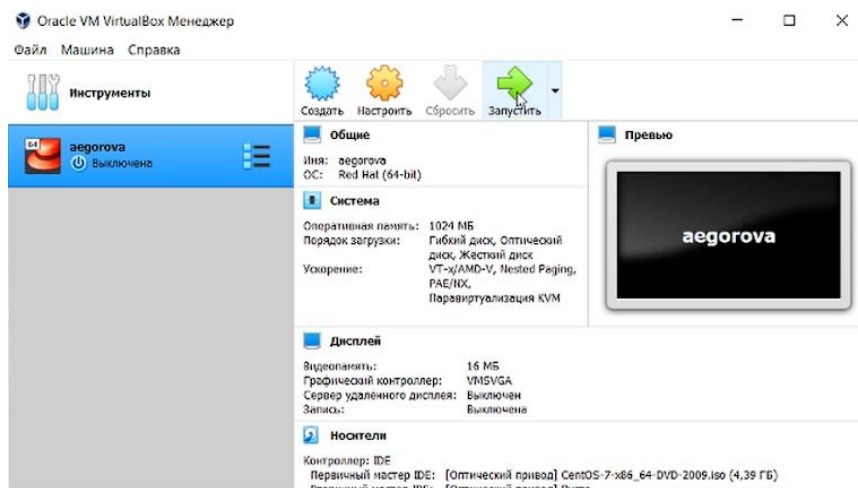


(Рис.14)



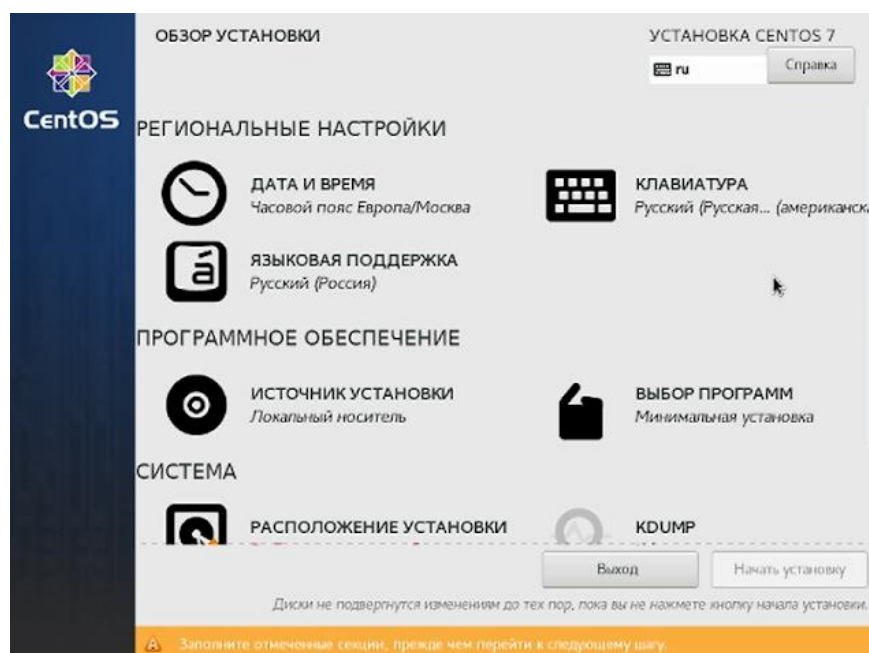
(Рис.15)

После этого необходимо запустить виртуальную машину и продолжить настройку (Рис.16).



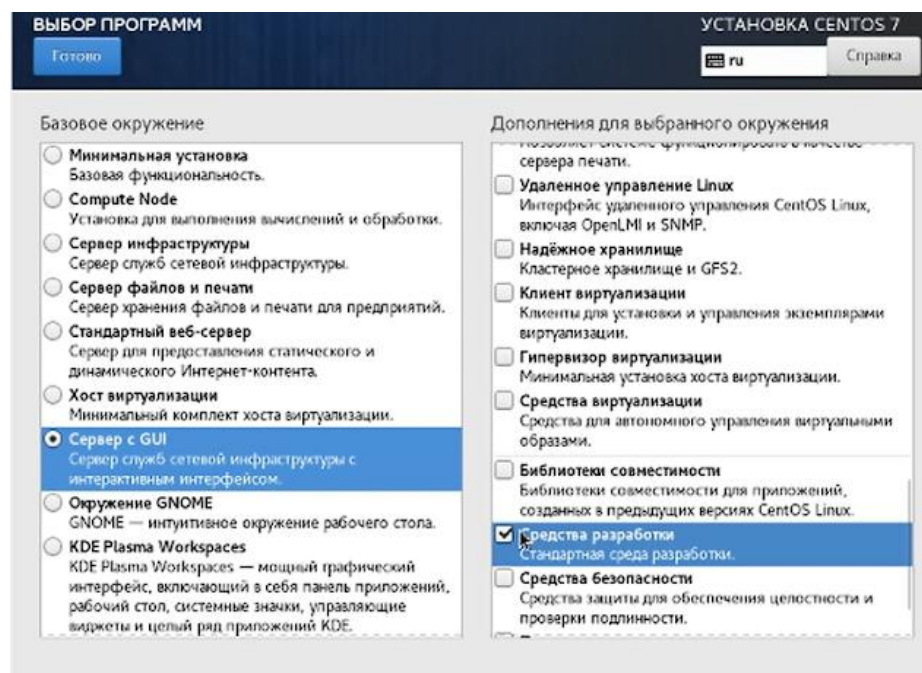
(Рис.16)

После запуска открылось окно настройки установки образа ОС (Рис.17)



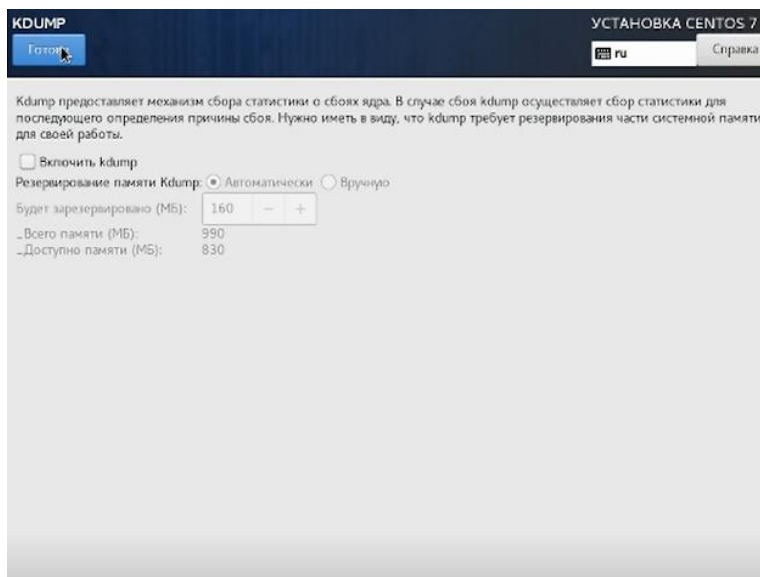
(Рис.17)

Заходим в окно настройки выбора программ: выбираем базовое окружение “Сервер с GUI” и дополнение к выбранному окружению “Средства разработки” (Рис.18).



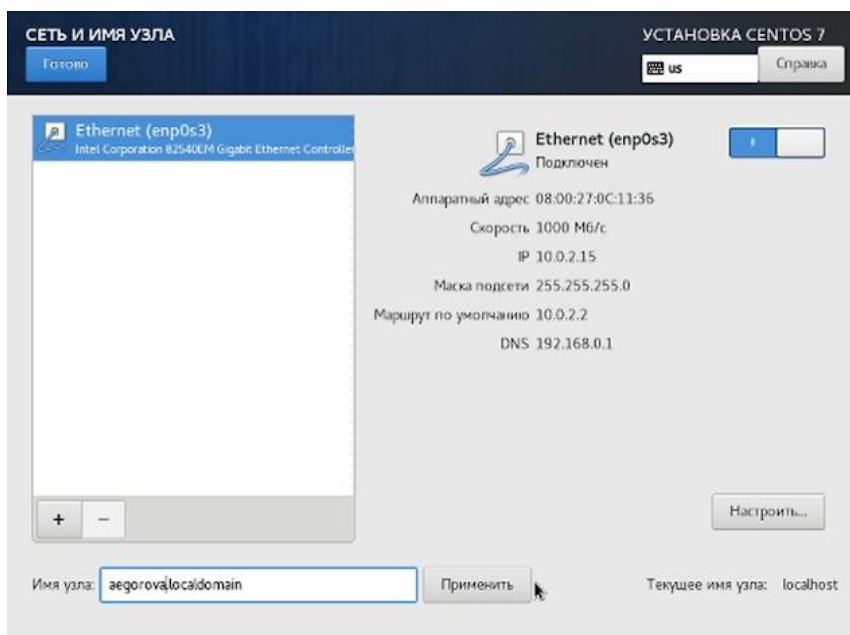
(Рис.18)

Далее отключаем KDUMP (Рис.19).



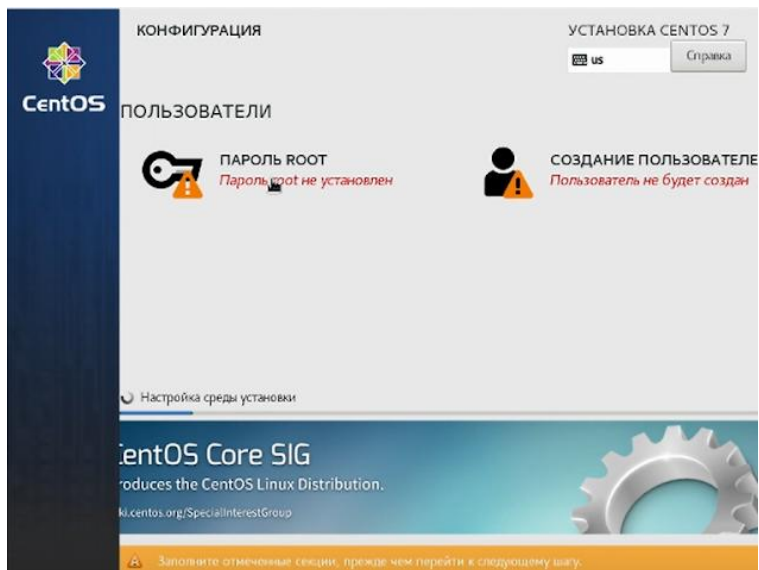
(Рис.19)

Открываем “Сеть и имя узла”: включаем сеть и указываем имя узла (в данном случае “aegorova”) (Рис.20).

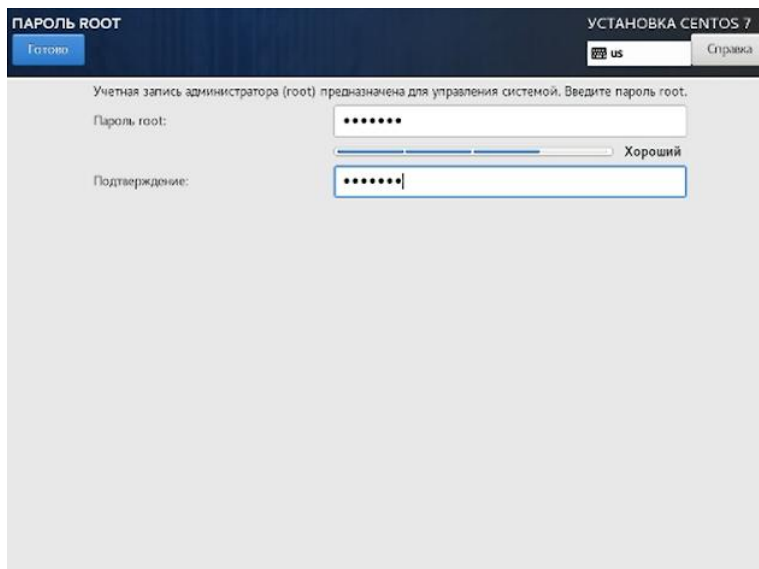


(Рис.20)

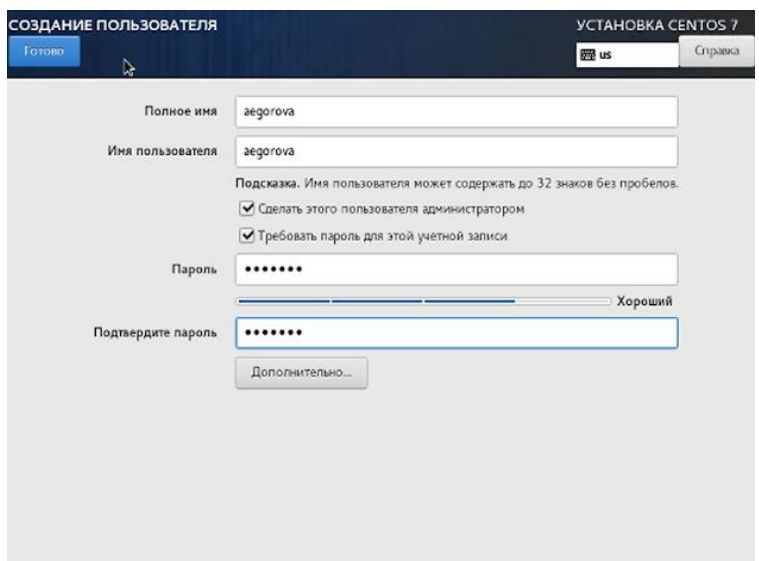
Начинаем установку (Рис.21). После этого нам нужно придумать пароль root (Рис.22) и создать пользователя (имя должно совпадать, т.е. указываем “aegorova”) и делаем пользователя администратором (Рис.23). Как только мы все указали, начинается процесс установки пакетов.



(Рис.21)

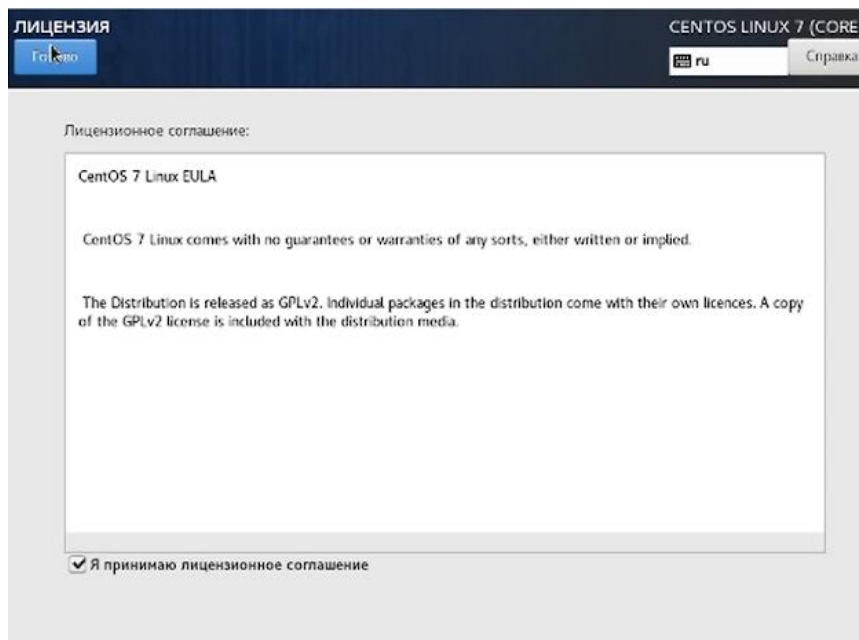


(Рис.22)



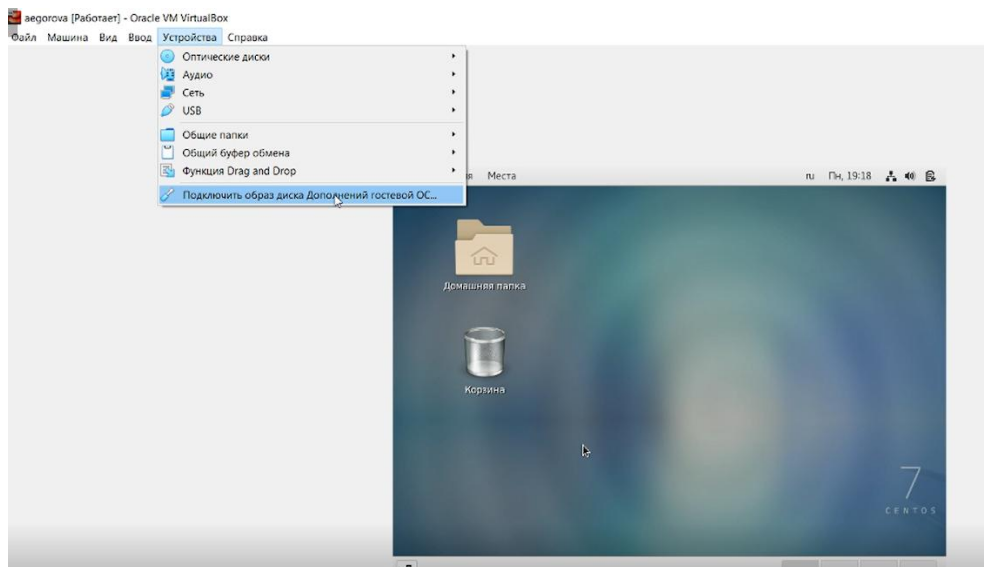
(Рис.23)

После перезагрузки виртуальной машины, нужно принять лицензию и завершить работу. После этого снова начинается перезагрузка машины (Рис.24)

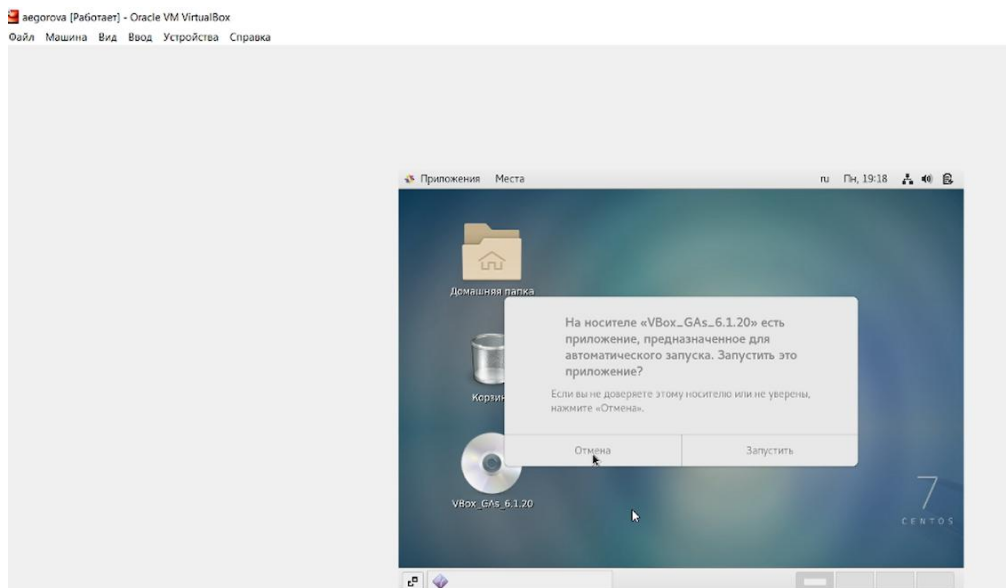


(Рис.24)

Заходим в ОС под заданной при установке учётной записью. В меню “Устройства” виртуальной машины подключаем образ диска дополнений гостевой ОС (Рис.25, 26), и вводим пароль пользователя root нашей виртуальной ОС.

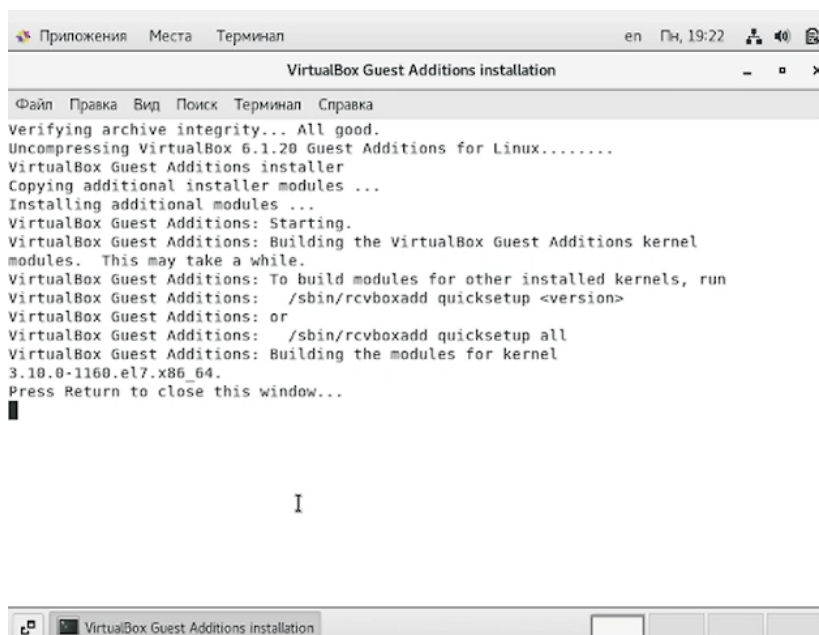


(Рис.25)



(Рис.26)

После загрузки дополнений нажимаем “Enter” (Рис.27) и корректно перезагружаем виртуальную машину.



(Рис.27)

Домашнее задание

Дождаемся загрузки графического окружения и открываем терминал. В окне терминала проанализируем последовательность загрузки системы, выполнив команду `dmesg`. Появляется большой список.

```
aegorova@aegorova:~$ dmesg
[ 0.000000] Initializing cgroup subsys cpuset
[ 0.000000] Initializing cgroup subsys cpu
[ 0.000000] Initializing cgroup subsys cpuct
[ 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)
Oct 19 16:18:59 UTC 2020
[ 0.000000] Command line: BOOT_IMAGE=/vmlinuz-3.10.0-1160.el7.x86_64 root=/dev/mapper/centos-root ro rd.lvm.lv=centos/root rd.lvm.lv=
et LANG=ru_RU.UTF-8
[ 0.000000] e820: 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 0x00000000000a0000-0x00000000000fffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000100000-0x00000000003fcfff] usable
[ 0.000000] BIOS-e820: [mem 0x00000000003fd000-0x00000000003fffff] ACPI data
[ 0.000000] BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000fec01000-0x00000000fec01fff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000ffffc000-0x00000000ffffcfff] reserved
[ 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] e820: update [mem 0x00000000-0x00000fff] usable ==> reserved
[ 0.000000] e820: remove [mem 0x00000000-0x00000fff] usable
[ 0.000000] e820: last_pfn = 0x3ffff max_arch_pfn = 0x400000000
[ 0.000000] MTRR default type: uncachable
[ 0.000000] MTRR variable ranges disabled:
[ 0.000000] PAT configuration [0-7]: WB WC UC- UC WB WP UC- UC
[ 0.000000] CPU MTRRs all blank - virtualized system.
[ 0.000000] found SMP MP-table at [mem 0x0009fff0-0x0009ffff] mapped at [ffffffffff200fff]
[ 0.000000] Base memory trampoline at [ffff881700099000] 99000 size 24576
[ 0.000000] BRK [0x1a674000, 0x1a674fff] PGTABLE
[ 0.000000] BRK [0x1a675000, 0x1a675fff] PGTABLE
[ 0.000000] BRK [0x1a676000, 0x1a676fff] PGTABLE
[ 0.000000] BRK [0x1a677000, 0x1a677fff] PGTABLE
[ 0.000000] BRK [0x1a678000, 0x1a678fff] PGTABLE
[ 0.000000] RAMDISK: [mem 0x343be000-0x361d6fff]
[ 0.000000] Early table checksum verification disabled
[ 0.000000] ACPI: RSDP 0000000000000000 00024 (v02 VBOX )
[ 0.000000] ACPI: XSDT 000000003ffff000 00030 (v01 VBOX VBOXXSDT 00000001 ASL 00000061)
[ 0.000000] ACPI: FACP 000000003ffff000 000F4 (v04 VBOX VBOXFACP 00000001 ASL 00000061)
```

Используем поиск с помощью `grep`: `dmesg | grep -i "то, что ищем"`.
Получите следующую информацию:

1. Версия ядра Linux (Linux version).

```
[aegorova@aegorova ~]$ 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
```

2. Частота процессора (Detected Mhz processor).

```
[aegorova@aegorova ~]$ dmesg | grep -i "Detected"
[ 0.000000] Hypervisor detected: KVM
[ 0.000000] Detected CPU family 6 model 126 stepping 5
[ 0.000000] tsc: Detected 1497.600 MHz processor
[ 0.707920] hub 1-0:1.0: 12 ports detected
[ 0.719003] systemd[1]: Detected virtualization kvm.
[ 0.719007] systemd[1]: Detected architecture x86_64.
```

3. Модель процессора (CPU0).

```
[aegorova@aegorova ~]$ dmesg | grep -i "CPU0"
[ 0.136630] smpboot: CPU0: Intel(R) Core(TM) i7-1065G7 CPU @ 1.30GHz (fam: 06, model: 7e, stepping: 05)
```

4. Объем доступной оперативной памяти (Memory available).

```
[aegorova@aegorova ~]$ dmesg | grep -i "Memory"
[ 0.000000] Base memory trampoline at [ffff881700099000] 99000 size 24576
[ 0.000000] Early memory node ranges
[ 0.000000] PM: Registered nosave memory: [mem 0x0009f000-0x0009ffff]
[ 0.000000] PM: Registered nosave memory: [mem 0x000a0000-0x000aefff]
[ 0.000000] PM: Registered nosave memory: [mem 0x000f0000-0x000fffff]
[ 0.000000] Memory: 980996k/1048512k available (7788k kernel code, 392k absent, 67124k reserved, 5954k data, 1984k init)
[ 0.000000] please try 'cgroup_disable=memory' option if you don't want memory cgroups
[ 0.041761] Initializing cgroup subsys memory
[ 0.236365] x86/mm: Memory block size: 128MB
[ 0.627385] Freeing initrd memory: 30820k freed
[ 0.655716] Non-volatile memory driver v1.3
[ 0.655842] crash memory driver: version 1.1
[ 0.714778] Freeing unused kernel memory: 1084k freed
[ 0.715452] Freeing unused kernel memory: 392k freed
[ 0.716129] Freeing unused kernel memory: 536k freed
[ 1.587615] [drm] Max dedicated hypervisor surface memory is 507904 kiB
[ 1.587616] [drm] Maximum display memory size is 16384 kiB
[ 1.587749] [TTM] Zone kernel: Available graphics memory: 507378 kiB
[aegorova@aegorova ~]$ dmesg
```

5. Тип обнаруженного гипервизора (Hypervisor detected).

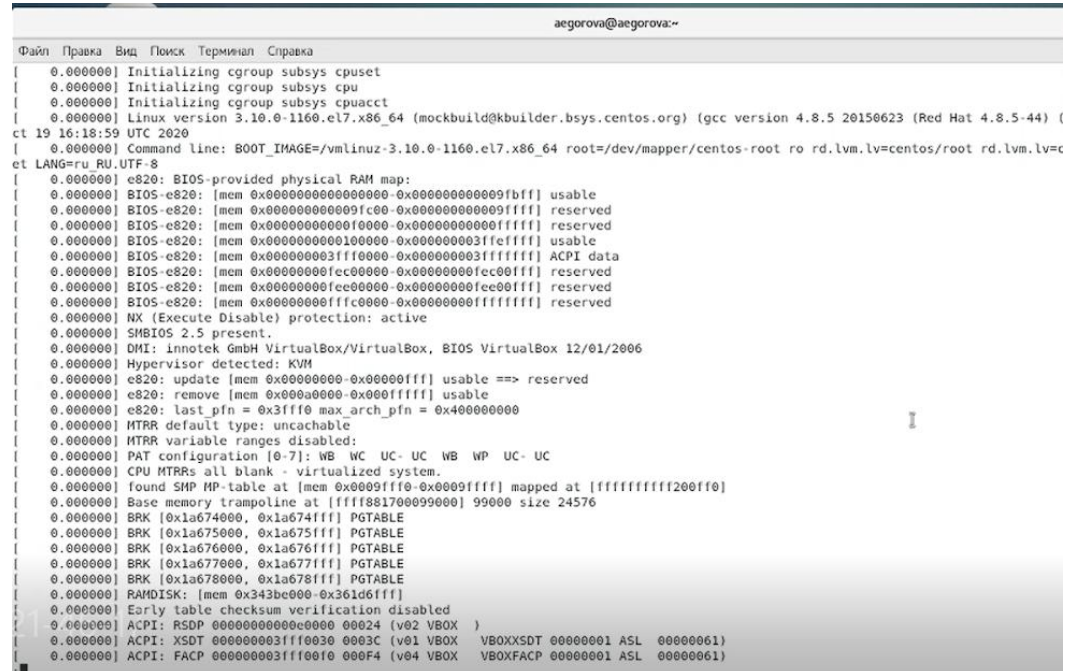
```
[aegorova@aegorova ~]$ dmesg | grep -i "Hypervisor"
[ 0.000000] Hypervisor detected: KVM
[ 1.587615] [drm] Max dedicated hypervisor surface memory is 507904 KiB
```

6. Тип файловой системы корневого раздела.

```
[aegorova@aegorova ~]$ dmesg | grep -i "Mount"
[ 0.041621] Mount-cache hash table entries: 2048 (order: 2, 16384 bytes)
[ 0.041623] Mountpoint-cache hash table entries: 2048 (order: 2, 16384 bytes)
[ 2.118946] XFS (dm-0): Mounting V5 Filesystem
[ 2.229820] XFS (dm-0): Ending clean mount
[ 3.587987] XFS (sda1): Mounting V5 Filesystem
[ 3.770085] XFS (sda1): Ending clean mount
```

7. Последовательность монтирования файловых систем.

```
[aegorova@aegorova ~]$ dmesg | less
```



The screenshot shows the output of the `dmesg | less` command. The output is a log of system boot messages. Key messages include: "Initializing cgroup subsys cpuset", "Initializing cgroup subsys cpu", "Initializing cgroup subsys cpuct", "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))", "Command line: BOOT_IMAGE=/vmlinuz-3.10.0-1160.el7.x86_64 root=/dev/mapper/centos-root ro rd.lvm.lv=centos/root rd.lvm.lv=centos LANG=ru_RU.UTF-8", "BIOS-e820: BIOS-provided physical RAM map:", "BIOS-e820: [mem 0x00000000-0x00000000] usable", "BIOS-e820: [mem 0x00000000-0x00000000] reserved", "BIOS-e820: [mem 0x00000000-0x00000000] reserved", "BIOS-e820: [mem 0x00000000-0x00000000] usable", "BIOS-e820: [mem 0x00000000-0x00000000] ACPI data", "BIOS-e820: [mem 0x00000000-0x00000000] reserved", "BIOS-e820: [mem 0x00000000-0x00000000] reserved", "BIOS-e820: [mem 0x00000000-0x00000000] reserved", "NX (Execute Disable) protection: active", "SMBIOS 2.5 present.", "DMI: Innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006", "Hypervisor detected: KVM", "e820: update [mem 0x00000000-0x00000000] usable ==> reserved", "e820: remove [mem 0x00000000-0x00000000] usable", "e820: last_pfn = 0x3ffff max_arch_pfn = 0x40000000", "MTRR default type: uncachable", "MTRR variable ranges disabled:", "PAT configuration [0-7]: WB WC UC- UC WB WP UC- UC", "CPU MTRRs all blank - virtualized system.", "found SMP MP-table at [mem 0x0009fff0-0x0009ffff] mapped at [ffffffffff200fff0]", "Base memory trampoline at [ffff801700099000] 99000 size 24576", "BRK [0x1a674000, 0x1a674fff] PGTABLE", "BRK [0x1a675000, 0x1a675fff] PGTABLE", "BRK [0x1a676000, 0x1a676fff] PGTABLE", "BRK [0x1a677000, 0x1a677fff] PGTABLE", "BRK [0x1a678000, 0x1a678fff] PGTABLE", "RAMDISK: [mem 0x343bc000-0x343d6fff]", "Early table checksum verification disabled", "ACPI: RSDP 0000000000000000 00024 (v02 VBOX)", "ACPI: XSDT 000000003ffff000 0003C (v01 VBOX VBOXXSDT 00000001 ASL 00000001)", "ACPI: FACP 000000003ffff000 000F4 (v04 VBOX VBOXFACP 00000001 ASL 00000001)

Контрольные вопросы

1) Какую информацию содержит учётная запись пользователя?

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) 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`» 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) Что такое файловая система? Приведите примеры с краткой характеристикой.

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.

- **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.
- **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*240 байт) для 32-х разрядных систем.

- **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 также 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.
- **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.

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?

Основные сигналы, 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.

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 приобрела практические навыки установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов.