

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: Морозова Анастасия Владимировна

Группа: НПИМбд-02-20

МОСКВА

2021 г.

Цель работы: Приобретение практических навыков установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов.

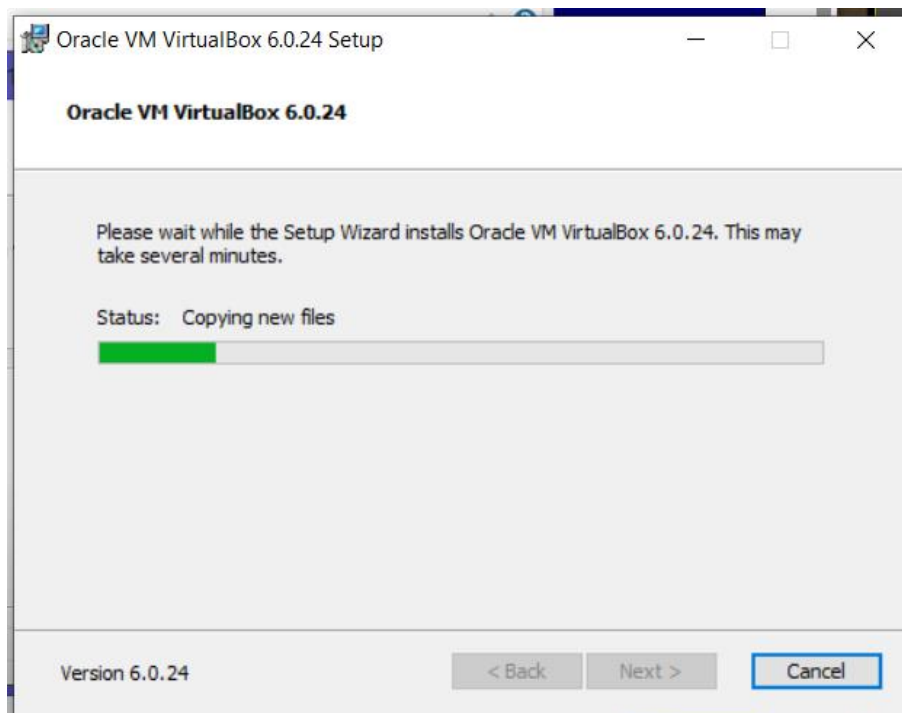
Ход работы:

- 1) Скачиваю VirtualBox, необходимая для запуска виртуальной машины.
Официальный сайт (<https://www.virtualbox.org/>). Выбираю необходимую операционную систему

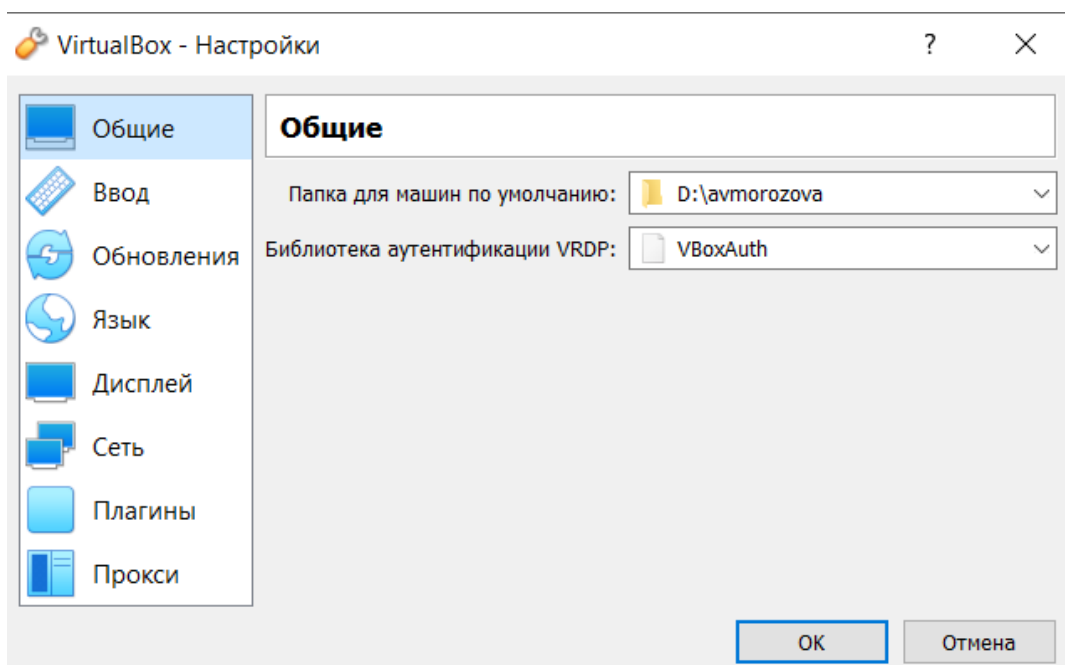


- 2) Выполняю установку скачанного файла:





- 3) Создаю папку на диске D, в котором будет храниться виртуальная машина. Имя папки – это имя пользователя (“avmorozova”). Проверяю в общих настройках поле «Папка для машин по умолчанию», указывая путь к папке, созданной ранее:



4) Создаю виртуальную машину: «Машина» → «Создать»:

Создать виртуальную машину

Укажите имя и тип ОС

Пожалуйста укажите имя и местоположение новой виртуальной машины и выберите тип операционной системы, которую Вы собираетесь установить на данную машину. Заданное Вами имя будет использоваться для идентификации данной машины.

Имя:

Папка машины:

Тип:

Версия:

Экспертный режим

Указываю размер основной памяти виртуальной машины – 1024 МБ или больше:

Создать виртуальную машину

Укажите объём памяти

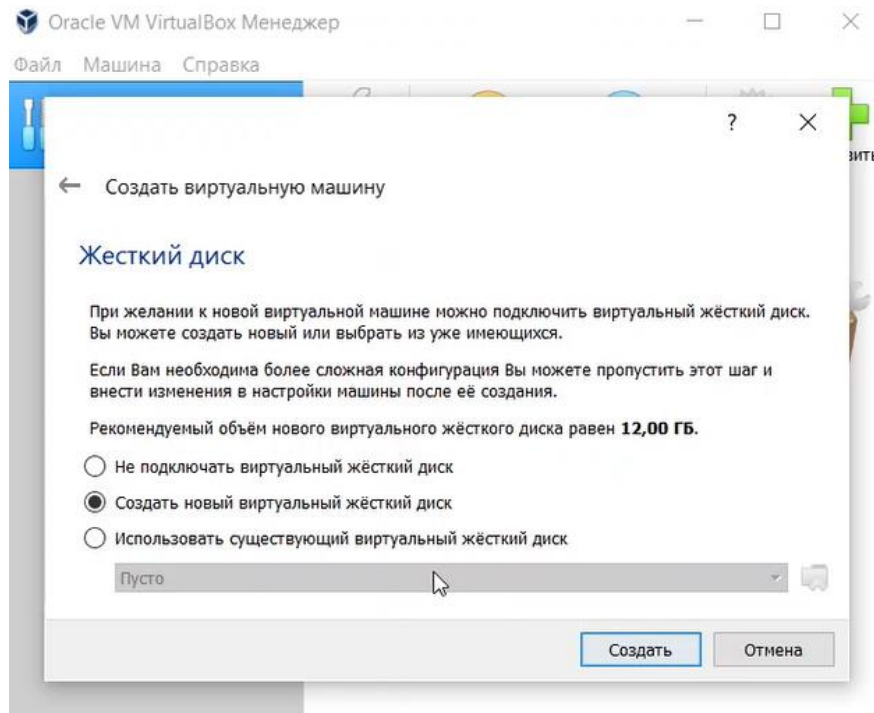
Укажите объём оперативной памяти (RAM) выделенный данной виртуальной машине.

Рекомендуемый объём равен **1024** МБ.

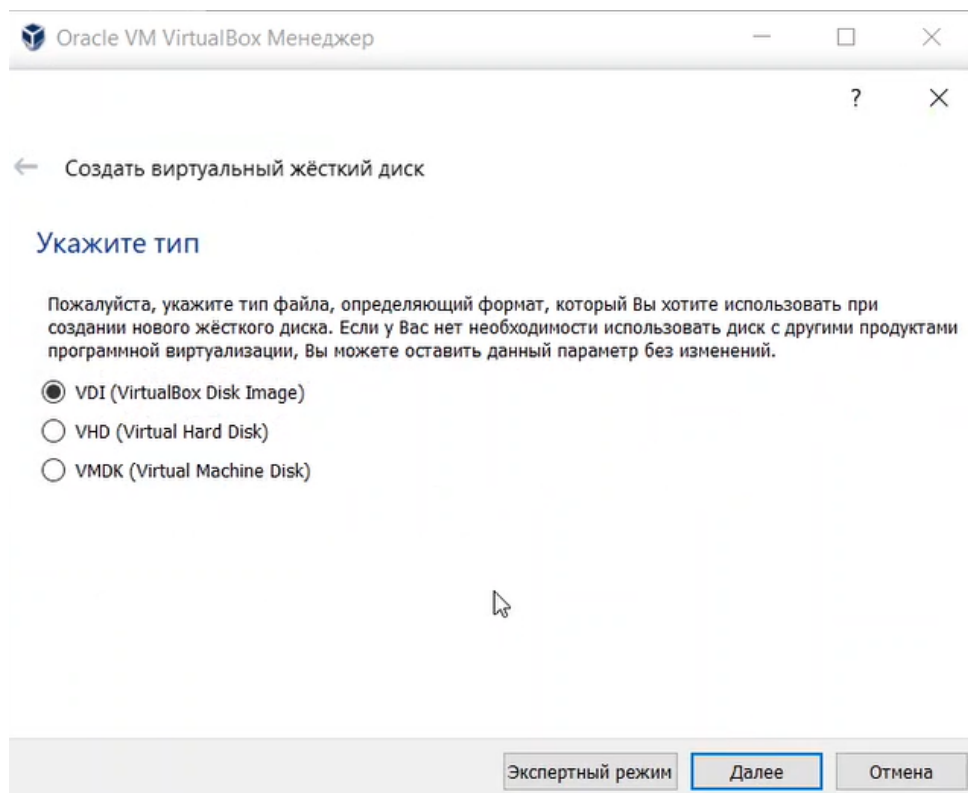
МБ

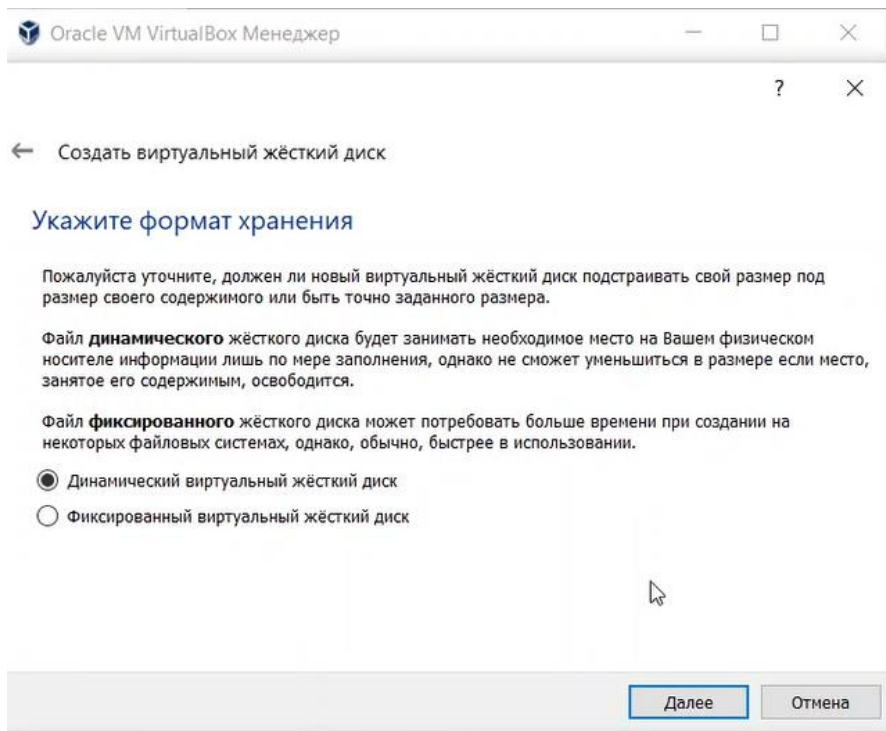
4 МБ 8192 МБ

Создаю новый виртуальный жесткий диск:



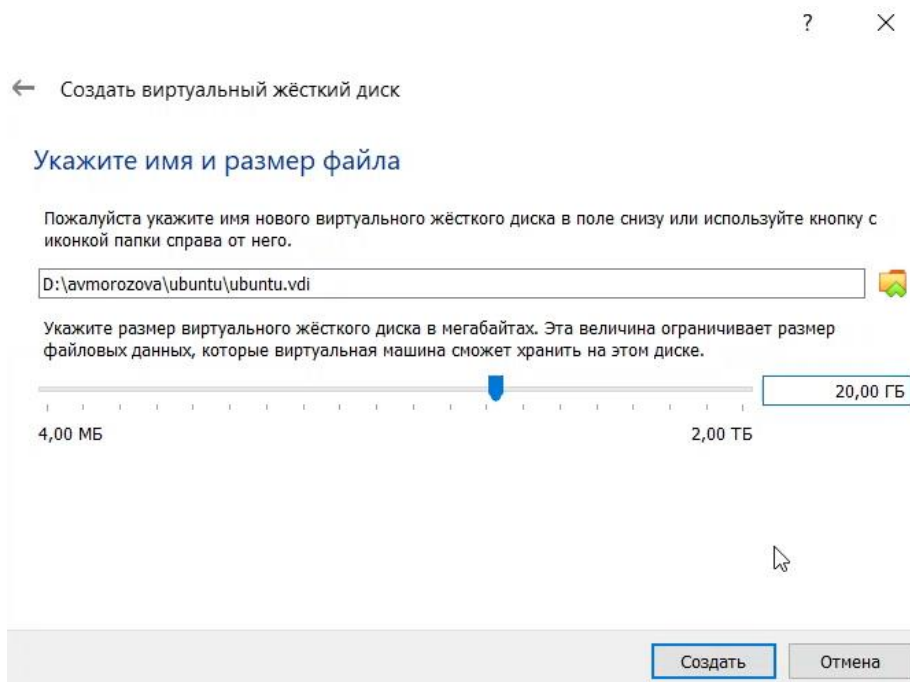
Задаю конфигурацию – VDI, динамический виртуальный жесткий диск:





Задаю расположение и размер диска: «D:\avmorozova\ubuntu\ubuntu.vdi».

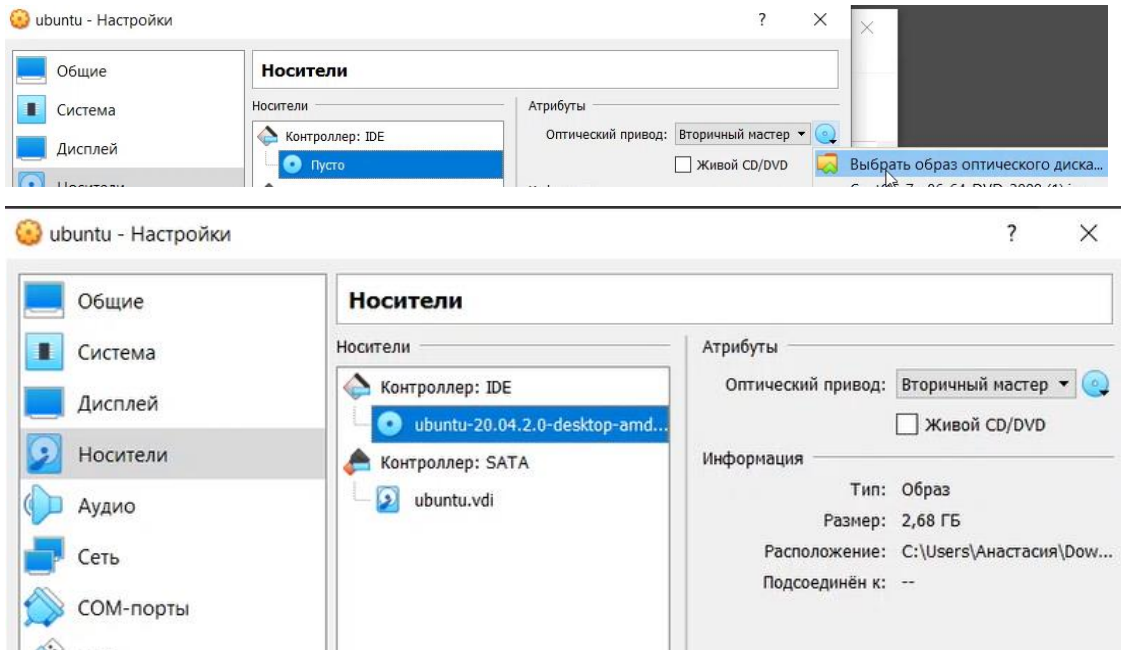
Размер 20 Гб или больше:



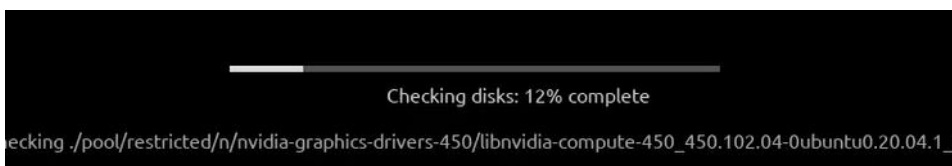
5) Скачиваю образ операционной системы. В моем случае – «ubuntu-20.04.2.0-desktop-amd64.iso».

Теперь в VirtualBox для виртуальной машины выбираем «Свойства» → «Носители». Добавляем новый привод оптических дисков и выбираем образ

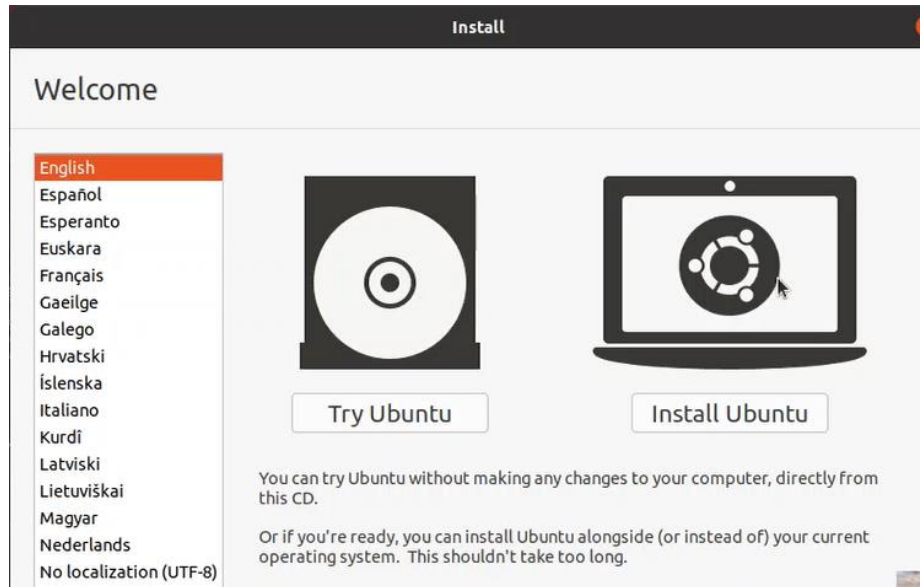
«ubuntu-20.04.2.0-desktop-amd64.iso»



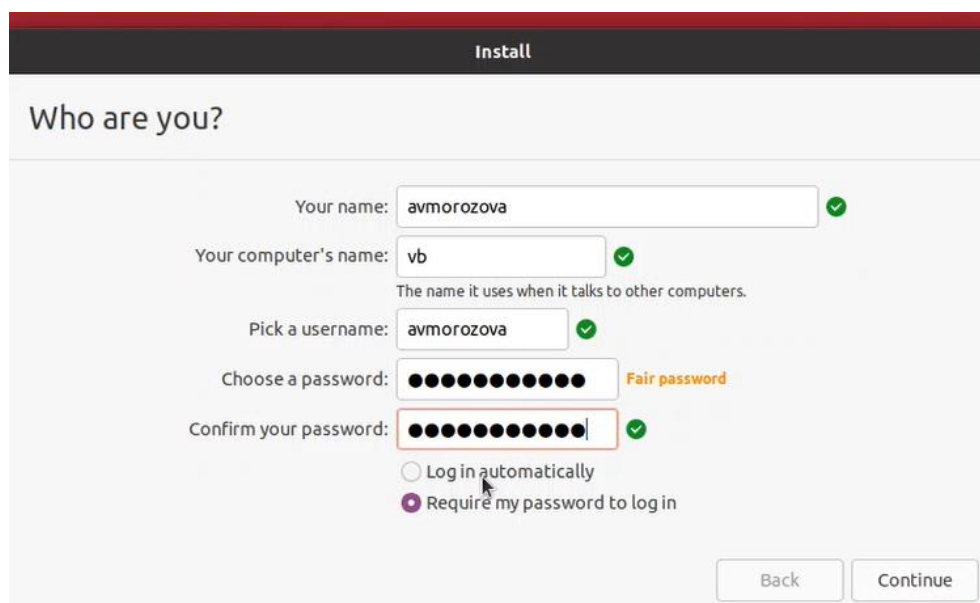
б) Запускаю виртуальную машину и продолжаю настройку



Устанавливаю Ubuntu



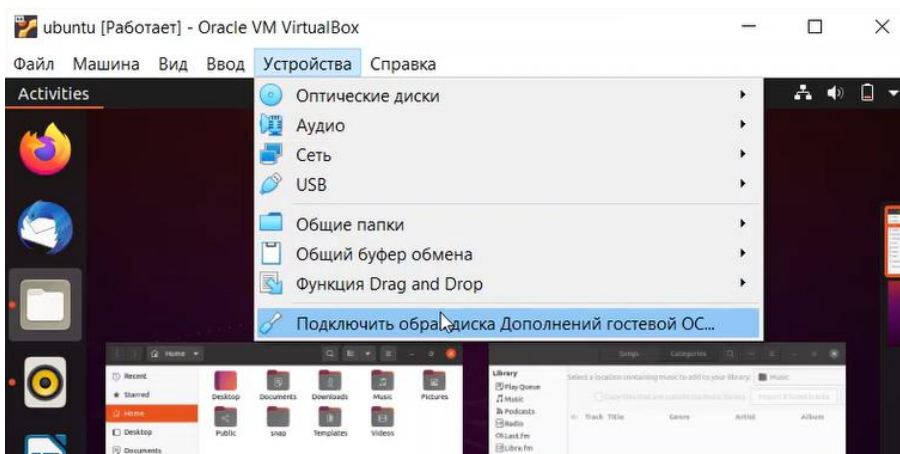
Устанавливаю имя и пароль для пользователя:



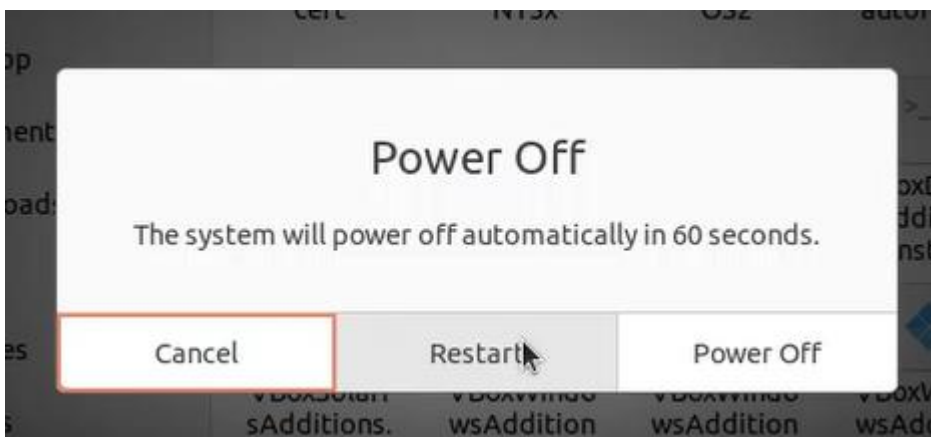
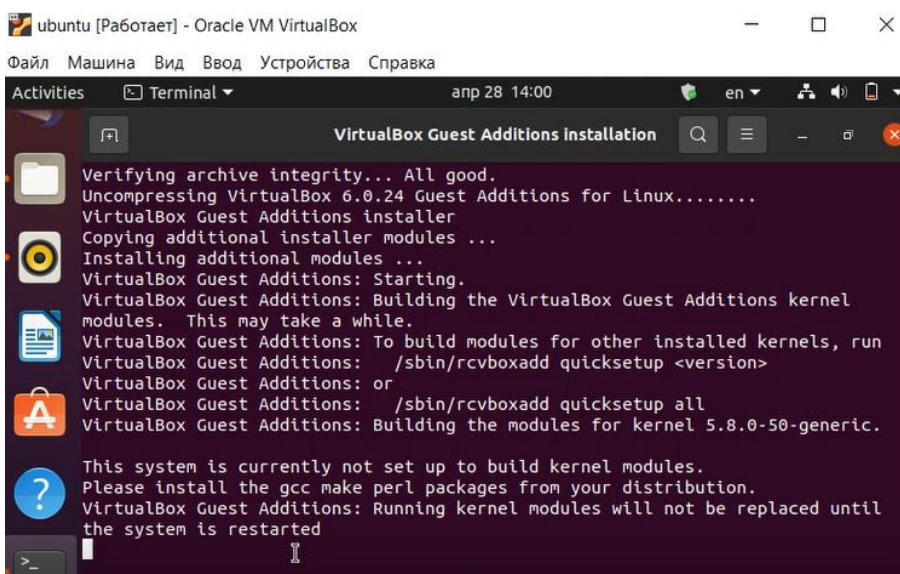
The 'Who are you?' screen of the Ubuntu installer. It contains several input fields and checkboxes. The 'Your name' field is filled with 'avmorozova'. The 'Your computer's name' field is filled with 'vb'. The 'Pick a username' field is filled with 'avmorozova'. The 'Choose a password' field is filled with dots, and the 'Confirm your password' field is also filled with dots. A 'Fair password' indicator is shown next to the password field. At the bottom, there are two radio buttons: 'Log in automatically' and 'Require my password to log in'. The 'Require my password to log in' option is selected. 'Back' and 'Continue' buttons are at the bottom right.



Подключаю образ диска дополнений гостевой Операционной Системы (меню «Устройства»):



По рекомендации, делаю перезагрузку:



7) Установка имени пользователя и названия хоста:

```
avmorozova@avmorozova:~$ sudo -i
[sudo] password for avmorozova:
root@avmorozova:~# adduser -G wheel avmorozova
Option g is ambiguous (gecos, gid, group)
adduser [--home DIR] [--shell SHELL] [--no-create-home] [--uid ID]
[--firstuid ID] [--lastuid ID] [--gecos GECOS] [--ingroup GROUP | --gid ID]
[--disabled-password] [--disabled-login] [--add_extra_groups]
[--encrypt-home] USER
    Add a normal user

adduser --system [--home DIR] [--shell SHELL] [--no-create-home] [--uid ID]
[--gecos GECOS] [--group | --ingroup GROUP | --gid ID] [--disabled-password]
[--disabled-login] [--add_extra_groups] USER
    Add a system user

adduser --group [--gid ID] GROUP
addgroup [--gid ID] GROUP
    Add a user group

addgroup --system [--gid ID] GROUP
    Add a system group

adduser USER GROUP
    Add an existing user to an existing group

general options:
  --quiet | -q          don't give process information to stdout
  --force-badname       allow usernames which do not match the
                        NAME_REGEX[_SYSTEM] configuration variable
  --extrausers          uses extra users as the database
  --help | -h          usage message
  --version | -v       version number and copyright
  --conf | -c FILE     use FILE as configuration file

root@avmorozova:~# passwd avmorozova
New password:
Retype new password:
passwd: password updated successfully
```

Таким образом у меня был создан пользователь с именем, удовлетворяющим соглашению об именовании

Далее устанавливаем необходимое имя хоста, получив права

администратора (команда «sudo -i») и использовав команду «hostnamectl set-hostname avmorozova». Удостоверяемся, что имя изменено, используя команду «hostnamectl»

```
ubuntu [Работает] - Oracle VM VirtualBox
Файл Машина Вид Ввод Устройства Справка
Activities Terminal apr 28 14:38 en
root@vb: ~
avmorozova@vb:~$ sudo -i
[sudo] password for avmorozova:
root@vb:~# hostnamectl set-hostname avmorozova
root@vb:~# hostnamectl
  Static hostname: avmorozova
            Icon name: computer-vm
            Chassis: vm
            Machine ID: 0df7a6d1ca794f278093ae316ed1f309
            Boot ID: 317c8f6cb31e4cfdbbe14d298193929e
    Virtualization: oracle
  Operating System: Ubuntu 20.04.2 LTS
            Kernel: Linux 5.8.0-50-generic
      Architecture: x86_64
root@vb:~#
```

Домашняя работа:

Загружаю графическое окружение и открываем консоль. Анализирую последовательность загрузки системы, используя команду «`sudo dmesg`» и введя пароль (т.к. команду использует обычный пользователь)

```
avmorozova@avmorozova:~$ sudo dmesg
[sudo] password for avmorozova:
[ 0.000000] Linux version 5.8.0-50-generic (buildd@lgw01-amd64-030) (gcc (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0, GNU ld (GNU Binutils for Ubuntu) 2.34) #56~20.04.1-Ubuntu SMP Mon Apr 12 21:46:35 UTC 2021 (Ubuntu 5.8.0-50.56~20.04.1-generic 5.8.18)
[ 0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-5.8.0-50-generic root=UUID=64b1ff00-c316-4b34-911c-3ead42e5f3d9 ro quiet splash
[ 0.000000] KERNEL supported cpus:
[ 0.000000]   Intel GenuineIntel
[ 0.000000]   AMD AuthenticAMD
[ 0.000000]   Hygon HygonGenuine
[ 0.000000]   Centaur CentaurHauls
[ 0.000000]   zhaoxin Shanghai
[ 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
```

Просмотрю вывод этой команды, выполнив команду «`sudo dmesg | less`»

```
avmorozova@avmorozova:~$ dmesg | less
```



```
[ 0.000000] Linux version 5.8.0-50-generic (bulld@lgw01-amd64-030) (gcc (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0, GNU ld (GNU Binutils for Ubuntu) 2.34) #56~20.04.1-Ubuntu SMP Mon Apr 12 21:46:35 UTC 2021 (Ubuntu 5.8.0-50.56~20.04.1-generic 5.8.18)
[ 0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-5.8.0-50-generic root=UUID=64b1ff00-c316-4b34-911c-3ead42e5f3d9 ro quiet splash
[ 0.000000] KERNEL supported cpus:
[ 0.000000] Intel GenuineIntel
[ 0.000000] AMD AuthenticAMD
[ 0.000000] Hygon HygonGenuine
[ 0.000000] Centaur CentaurHauls
[ 0.000000] zhaoxin Shanghai
[ 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 0x0000000000000000-0x0000000000009fbff] usable
[ 0.000000] BIOS-e820: [mem 0x0000000000009fc00-0x0000000000009ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x000000000000f0000-0x000000000000ffffff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000000100000-0x000000000000dffff] usable
[ 0.000000] BIOS-e820: [mem 0x00000000000dffff000-0x00000000000dffffff] ACPI data
[ 0.000000] BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000fee00000-0x00000000fee00fff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000fffc0000-0x00000000ffffffff] reserved
```

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

1) Версия ядра Linux: команда «`sudo dmesg | grep -i "Linux version"`». В данном случае версия операционной системы – 5.8.0-50-generic

```
avmorofova@avmorofova:~$ sudo dmesg | grep -i "Linux version"
[sudo] password for avmorofova:
Sorry, try again.
[sudo] password for avmorofova:
[ 0.000000] Linux version 5.8.0-50-generic (bulld@lgw01-amd64-030) (gcc (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0, GNU ld (GNU Binutils for Ubuntu) 2.34) #56~20.04.1-Ubuntu SMP Mon Apr 12 21:46:35 UTC 2021 (Ubuntu 5.8.0-50.56~20.04.1-generic 5.8.18)
```

2) Частота процессора: команда «`sudo dmesg | grep -i "MHz"`». Частота процессора составляет 2112 МГц

```
avmorofova@avmorofova:~$ sudo dmesg | grep -i "Mhz"
[ 0.000005] tsc: Detected 2112.000 MHz processor
[ 1.967254] e1000 0000:00:03.0 eth0: (PCI:33MHz:32-bit) 08:00:27:c9:69:58
```

3) Модель процессора: команда «`sudo dmesg | grep -i "CPU0"`». Из рисунка видно, что модель моего процессора – Intel(R) Core(TM) i5- 10210U CPU @ 1.60GHz

```
avmorofova@avmorofova:~$ sudo dmesg | grep -i "CPU0"
[ 0.474854] smpboot: CPU0: Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz (family: 0x6, model: 0x8e, stepping: 0xc)
```

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

```
avmorozova@avmorozova:~$ sudo dmesg | grep -i "Memory"
[ 0.092850] check: Scanning 1 areas for low memory corruption
[ 0.096803] Early memory node ranges
[ 0.206576] PM: hibernation: Registered nosave memory: [mem 0x00000000-0x000000fff]
[ 0.206578] PM: hibernation: Registered nosave memory: [mem 0x0009f000-0x0009ffff]
[ 0.206578] PM: hibernation: Registered nosave memory: [mem 0x000a0000-0x000aefff]
[ 0.206578] PM: hibernation: Registered nosave memory: [mem 0x000f0000-0x000fffff]
[ 0.206579] PM: hibernation: Registered nosave memory: [mem 0xdfff0000-0xdfffff]
[ 0.295165] Memory: 3971216K/4193848K available (14339K kernel code, 2537K wdata, 5452K rodata, 2644K init, 4916K bss, 222632K reserved, 0K cma-reserved)
[ 0.363789] Freeing SMP alternatives memory: 40K
[ 0.477199] x86/mm: Memory block size: 128MB
```

5) Тип обнаруженного гипервизора: команда «`sudo dmesg | grep -i "Hypervisor detected"`». Тип данного гипервизора – KVM

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

6-7) Тип файловой системы корневого раздела и последовательность монтирования файловых систем: команда «`sudo dmesg | grep -i "Mount"`». Тип файловой системы корневого раздела – EXT4.

```
avmorozova@avmorozova:~$ sudo dmesg | grep -i "Mount"
[ 0.355848] Mount-cache hash table entries: 8192 (order: 4, 65536 bytes, linear)
[ 0.355856] Mountpoint-cache hash table entries: 8192 (order: 4, 65536 bytes, linear)
[ 2.568298] EXT4-fs (sda5): mounted filesystem with ordered data mode. Opts: (null)
[ 3.354457] systemd[1]: Set up automount Arbitrary Executable File Formats File System Automount Point.
[ 3.356646] systemd[1]: Mounting Huge Pages File System...
[ 3.357681] systemd[1]: Mounting POSIX Message Queue File System...
[ 3.358812] systemd[1]: Mounting Kernel Debug File System...
```

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.

3) 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 пэбибайта (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 исходной 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. На данный момент является наиболее стабильной и поддерживаемой файловой системой в среде Linux.
- 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.
- 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.
- 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 делать 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 небезызвестной 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 дампы 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 выполнения работы я приобрела практические навыки установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов, поиска информации об установленной ОС, используя консоль