

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: Хусяинова Адиля Фаритовна

Группа: НПИМбд-03-21

МОСКВА

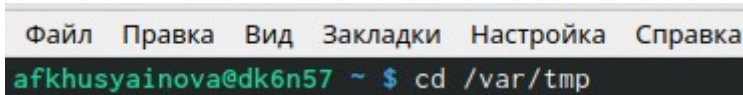
2022 г.

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

Задание: Необходимо установить на VirtualBox операционную систему Linux.

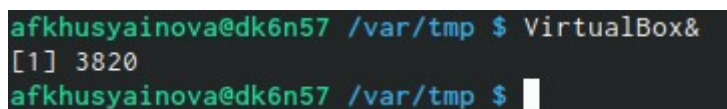
Ход работы:

- 1) Загрузим в дисплейном классе операционную систему Linux.
- 2) Осуществим вход в систему и запустим терминал, перейдя при этом в каталог `cd /var/tmp`



```
Файл Правка Вид Закладки Настройка Справка
afkhusyainova@dk6n57 ~ $ cd /var/tmp
```

- 3) Запустим виртуальную машину, для этого необходимо ввести в командной строке `VirtualBox&`



```
afkhusyainova@dk6n57 /var/tmp $ VirtualBox&
[1] 3820
afkhusyainova@dk6n57 /var/tmp $
```

Проверим в свойствах VirtualBox месторасположение каталога для виртуальных машин. Для этого в VirtualBox выберем Файл Свойства , вкладка Общие . В поле Папка для машин (рис. 1) должно стоять `var/tmp/имя_пользователя`

Здесь имя_пользователя — логин (учётная запись) студента в дисплейном классе.

Если указан другой каталог, то требуется изменить его, как указано выше

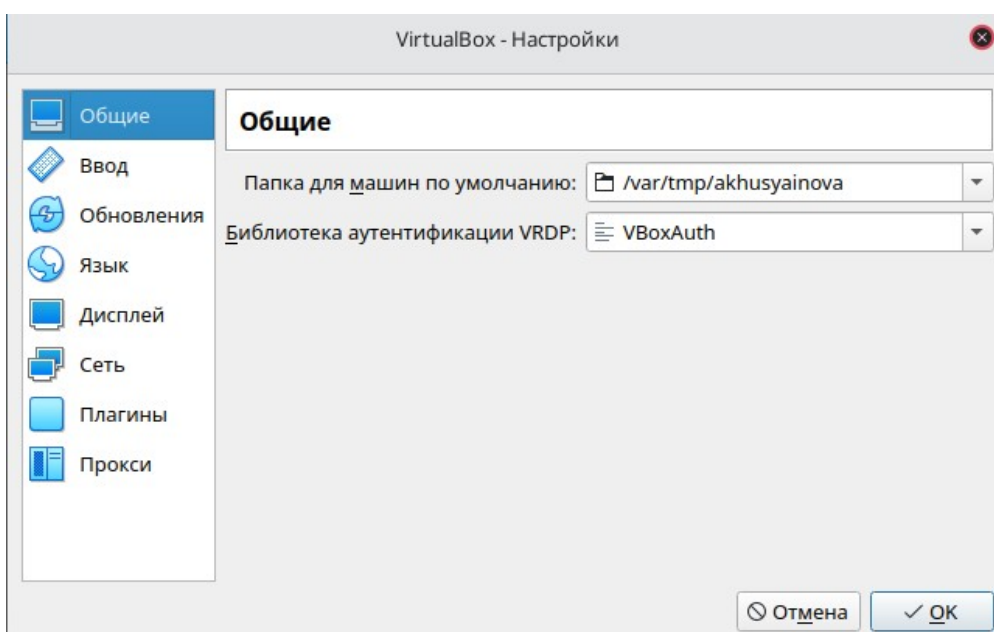


рис 1.

Создадим новую виртуальную машину. Для этого в VirtualBox выберем **Машина Создать** . Укажем имя виртуальной машины (ваш логин в дисплейном классе), тип операционной системы — **Linux, Fedora** (рис. 2)

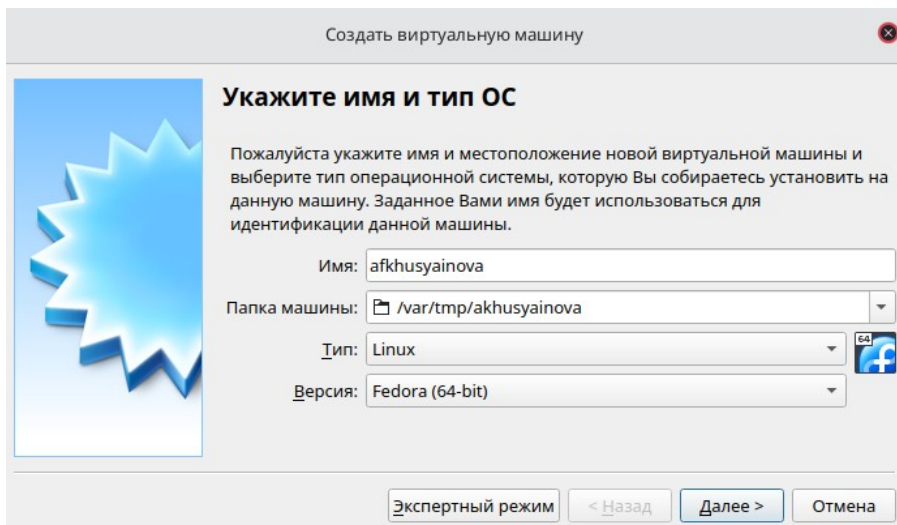


рис 2

Укажем размер основной памяти виртуальной машины от 2048 МБ (рис 3).

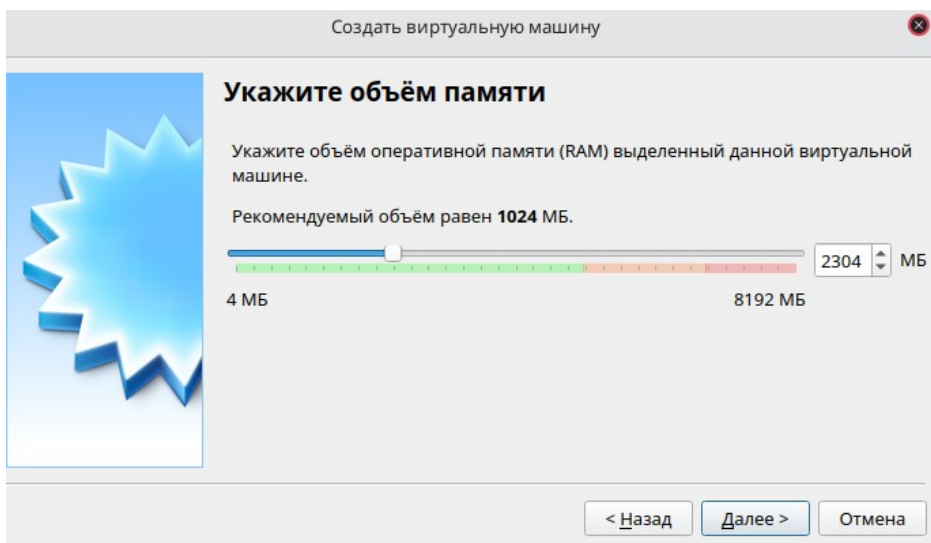
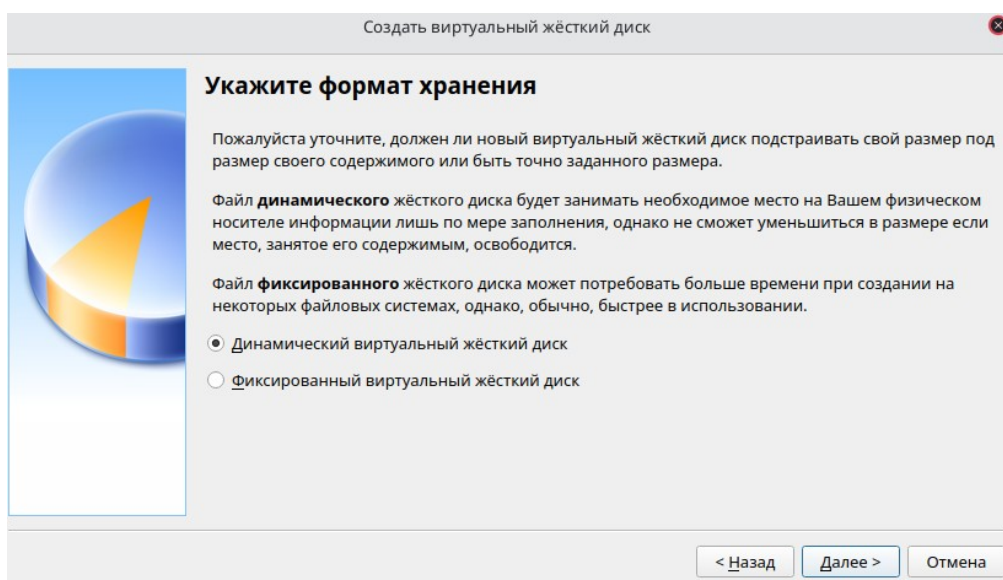
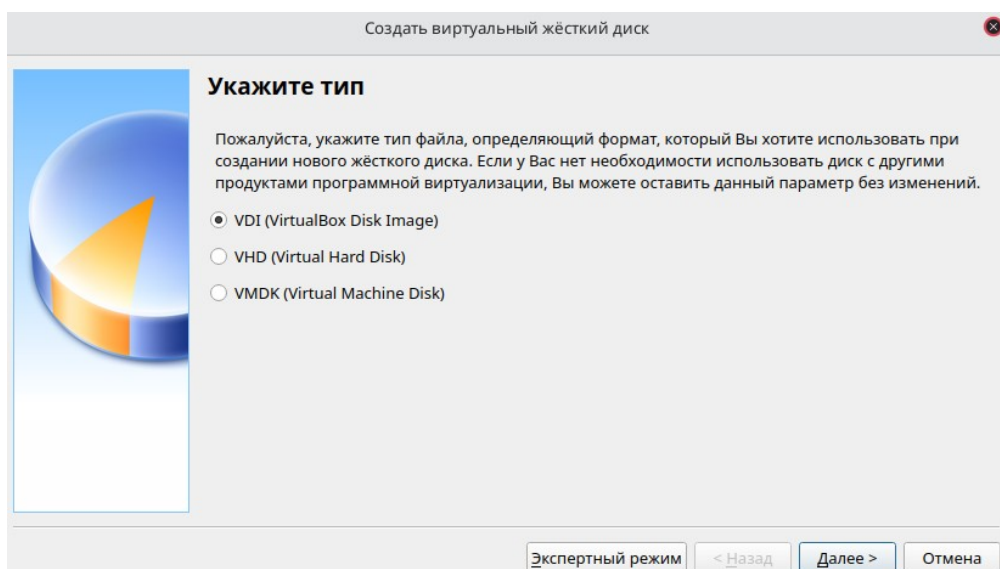
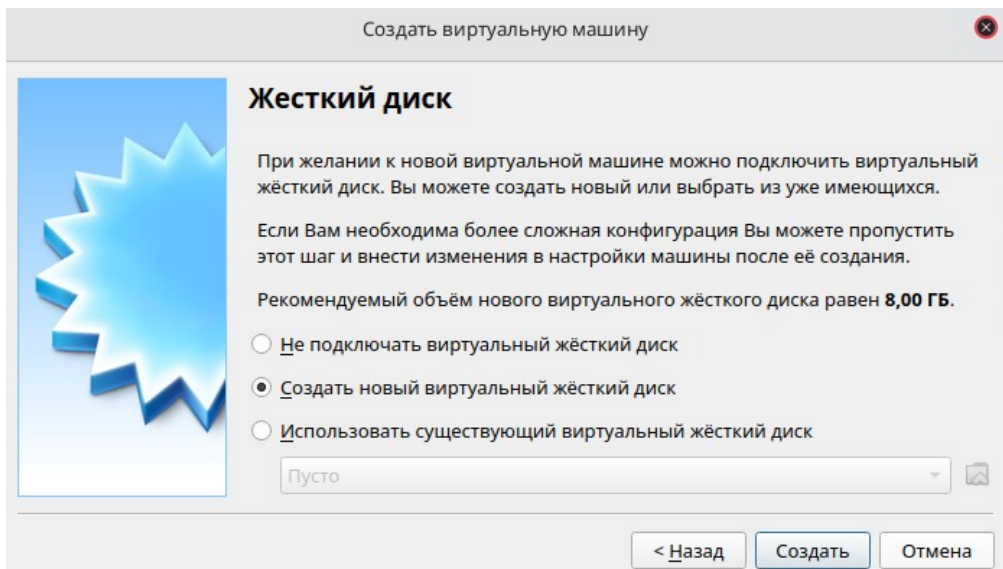


рис 3

Зададим конфигурацию жёсткого диска — загрузочный, VDI (VirtualBox Disk Image), динамический виртуальный диск.



Зададим размер диска — 80 ГБ (или больше), его расположение — в данном случае /var/tmp/имя_пользователя/fedora.vdi (рис 4)

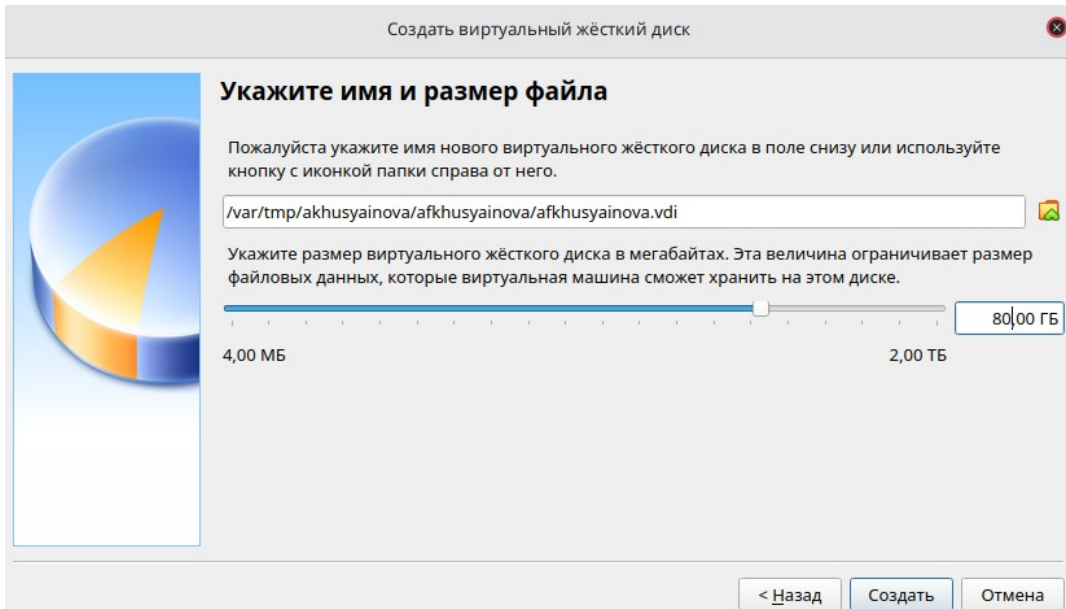


рис 4

Выберем в VirtualBox Свойства Носители нашей виртуальной машины. Добавим новый привод оптических дисков и выберите образ afs dk.sci.pfu.edu.ru common files iso Fedora-Live-Desktop-i686-19-1.iso (рис. 5).

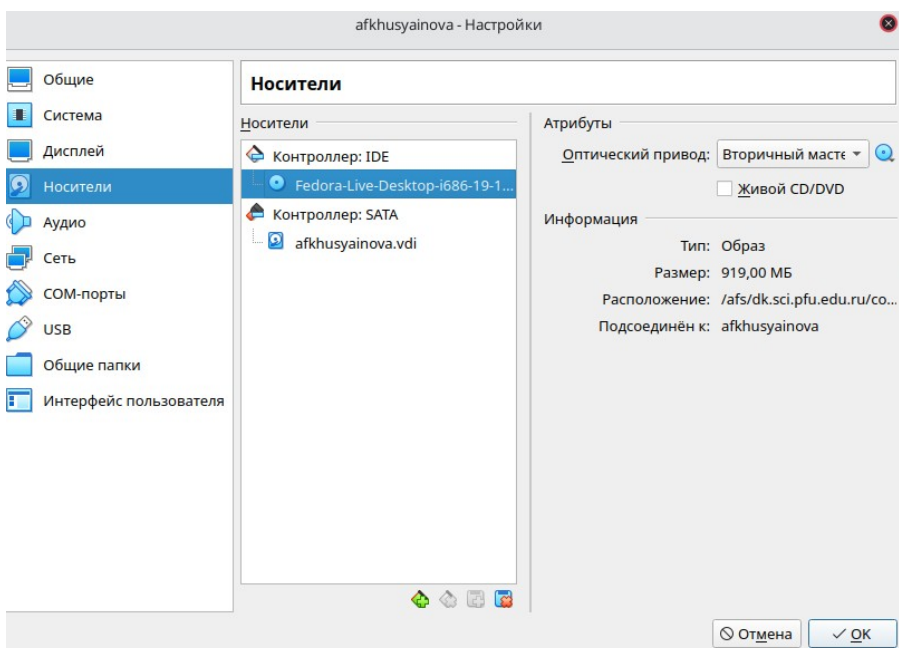
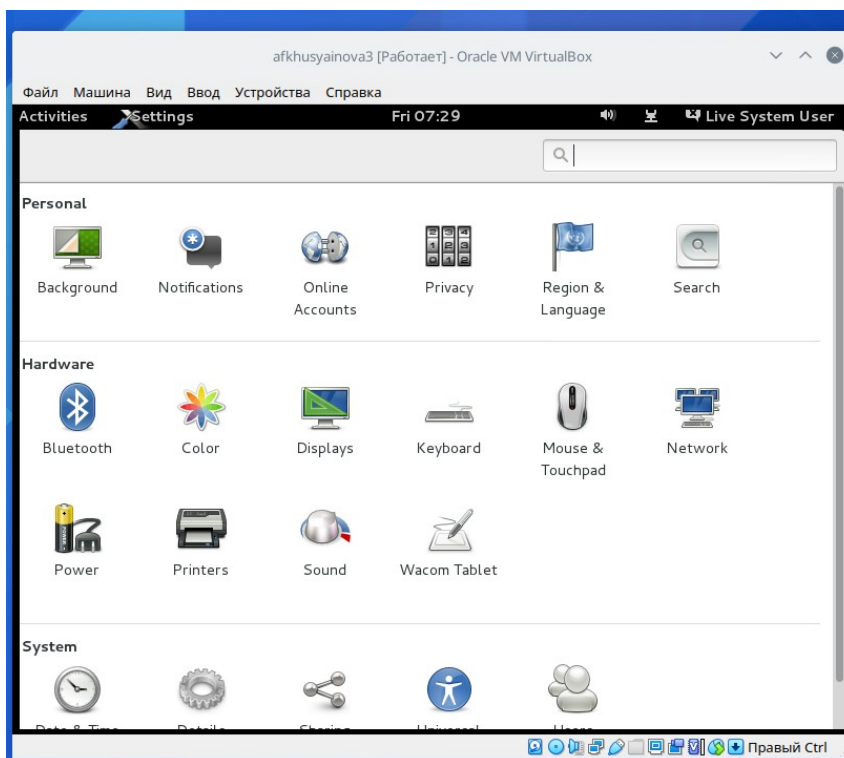
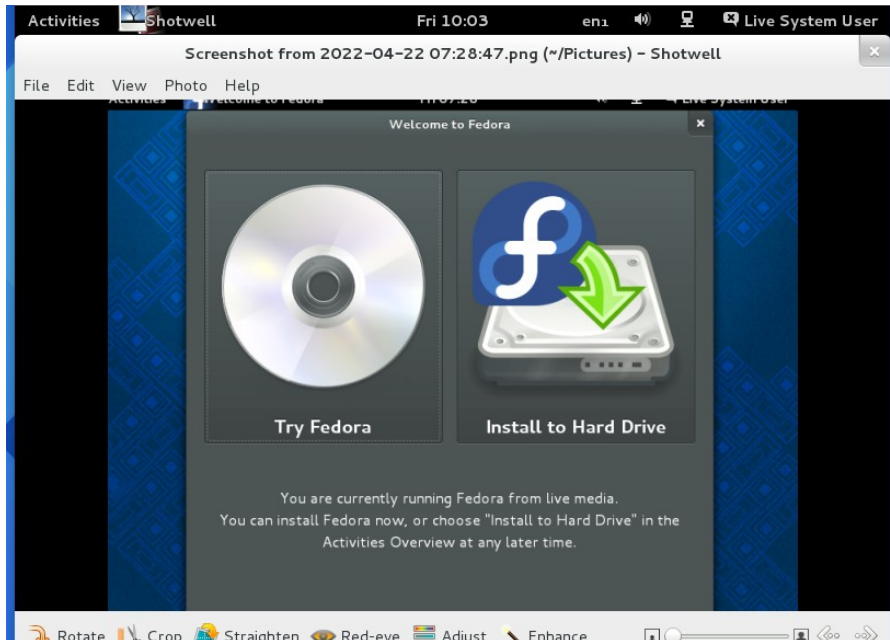


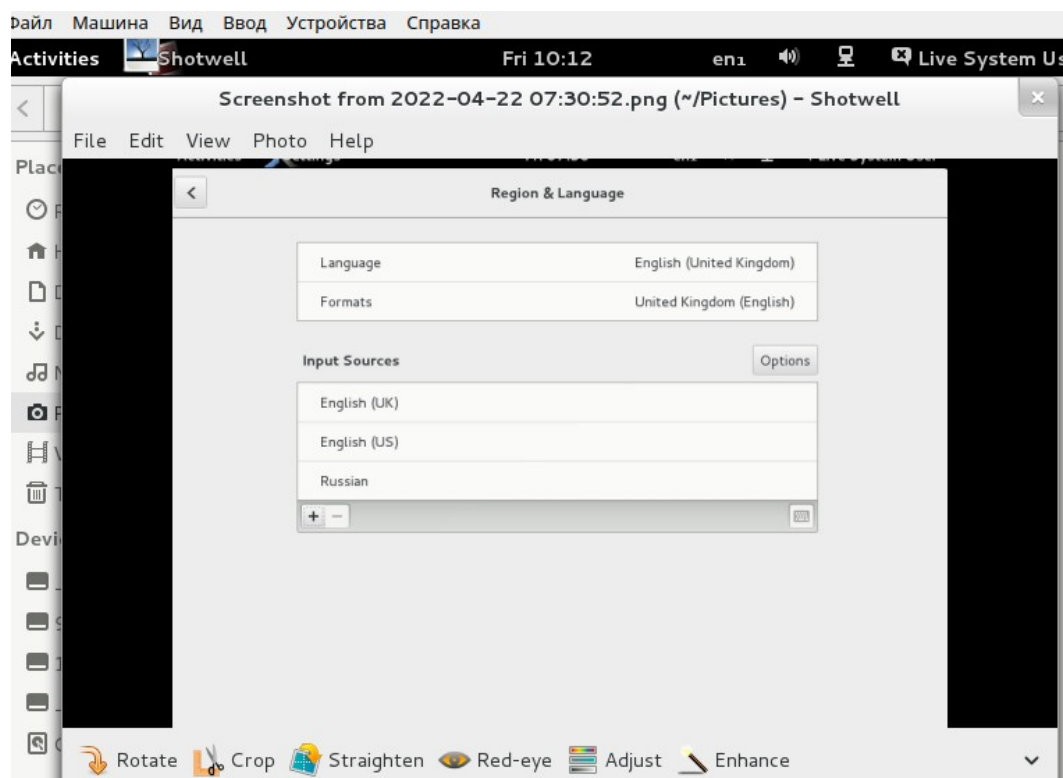
рис 5

Запустим виртуальную машину, выберем язык интерфейса и перейдем к настройкам установки операционной системы (рис. 6). При необходимости скорректируйте часовой пояс, раскладку клавиатуры

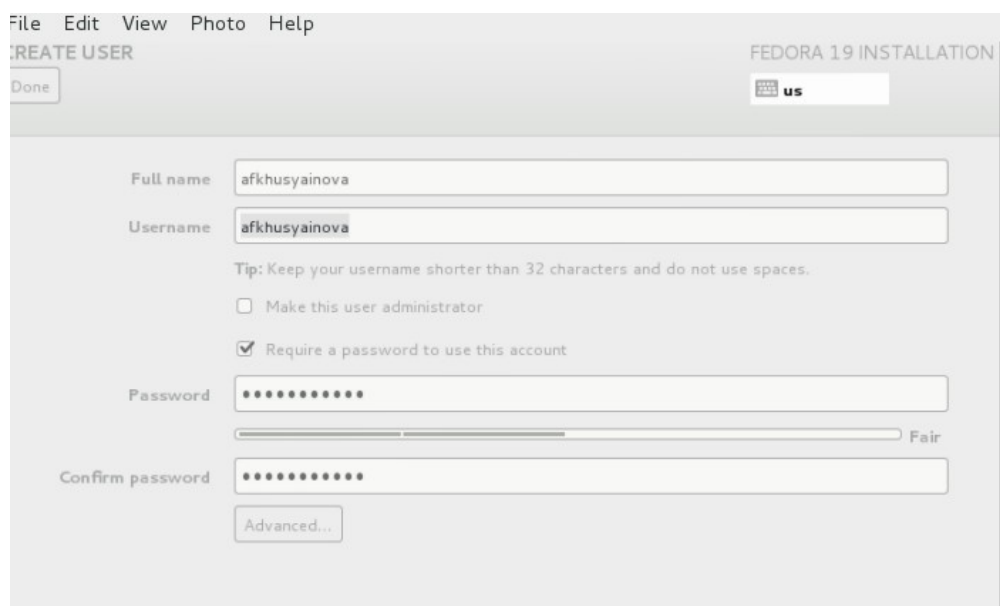


Окно настройки установки образа ОС

рис 6



Установка пароля для пользователя



Устанавливаем имя пользователя и называем хост:

1. Запустим виртуальную машину и необходимо залогиниться.
2. Запустим терминал и получим полномочия администратора с помощью su-
3. Создайте пользователя (вместо username укажем наш логин в дисплейном классе):`adduser -G wheel username`
4. Зададим пароль для пользователя (вместо username укажем наш логин в дисплейном классе) `passwd username`
5. Установим имя хоста (вместо username укажем наш логин в дисплейном классе):`hostnamectl set-hostname username`
- 6.Проверим, что имя хоста установлено верно: `hostnamectl`

```
root@localhost:~  
File Edit View Search Terminal Help  
liveuser@localhost ~]$ su -  
root@localhost ~]# adduser -G wheel afkhusyainova  
root@localhost ~]# passwd afkhusyainova  
Changing password for user afkhusyainova.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
root@localhost ~]# hostnamectl set-hostname afkhusyainova  
root@localhost ~]# hostnamectl  
Static hostname: afkhusyainova  
Icon name: computer-vm  
Chassis: vm  
Machine ID: b80e6b9c48db4202a3c54db1b32f6602  
Boot ID: 54339592506544028f721807cce24436  
Virtualization: kvm  
Operating System: Fedora 19 (Schrödinger's Cat)  
CPE OS Name: cpe:/o:fedoraproject:fedora:19  
Kernel: Linux 3.9.5-301.fc19.i686  
Architecture: i686  
root@localhost ~]#
```

5. Контрольные вопросы:

1) Учётная запись пользователя содержит следующую информацию: 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) Основные 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 2`» покажет список последних 2 команд

3) Файловая система – 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 у него прекрасная поддержка больших файлов и файловых томов, 8 эксбибайт (8*260 байт) для 64-х битных систем.
- ReiserFS (Reiser3) – Максимальный объём тома для этой системы равен 16 тебибайт (16*240 байт).
- JFS (Journaled File System) 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) – Разработана Remy Card с целью преодолеть ограничения файловой системы Minix.
- EXT2 (second extended file system) – не журналируемая файловая система, это был основной её недостаток, который исправит EXT3.
- EXT3 - расширение исконной для Linux EXT2, способное к журналированию. 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 – имеет 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. • TUX2 – известная, но так и не анонсированная публично файловая система. Система базируется на алгоритме «Фазового Древа», который как и журналирование защищает файловую систему от сбоев. Организована как надстройка на EXT2. • TUX3 – 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.

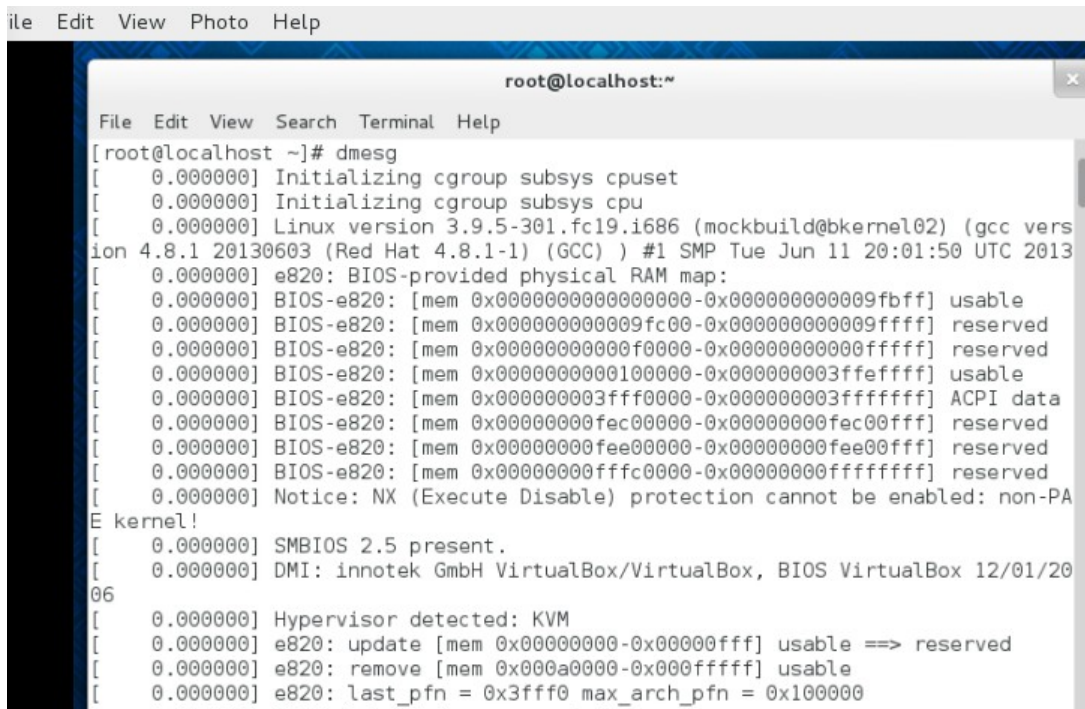
4) Команда «findmnt» или «findmnt --all» будет отображать все подмонтированные файловые системы или искать файловую систему.

5) 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+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).

6. Выполнение домашней работы:

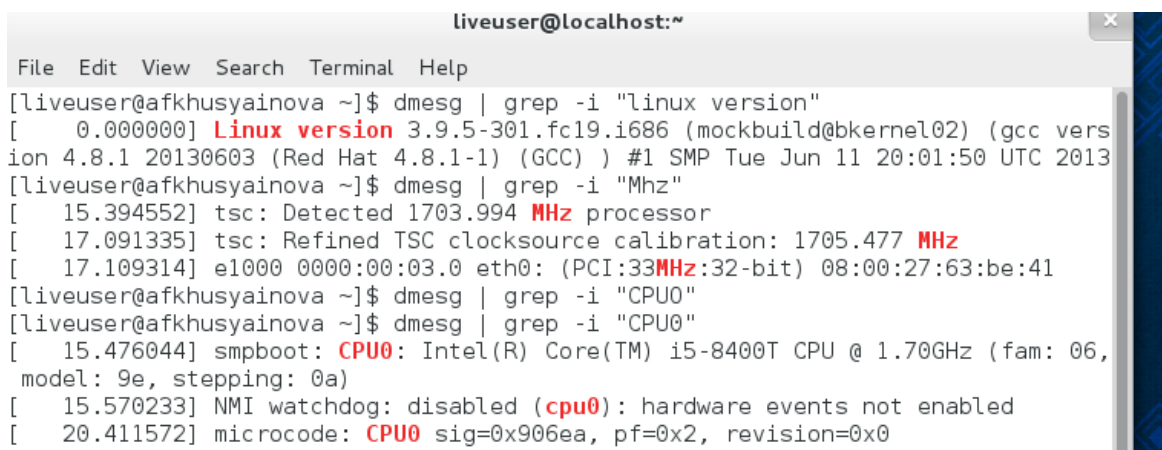
В окне терминала проанализируем последовательность загрузки системы, выполнив команду `dmesg`. Просмотрим вывод этой команды:



```
root@localhost:~  
File Edit View Search Terminal Help  
[root@localhost ~]# dmesg  
[ 0.000000] Initializing cgroup subsys cpuset  
[ 0.000000] Initializing cgroup subsys cpu  
[ 0.000000] Linux version 3.9.5-301.fc19.i686 (mockbuild@bkernel02) (gcc version 4.8.1 20130603 (Red Hat 4.8.1-1) (GCC) ) #1 SMP Tue Jun 11 20:01:50 UTC 2013  
[ 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 0x00000000000f0000-0x00000000000fffff] reserved  
[ 0.000000] BIOS-e820: [mem 0x0000000000100000-0x00000000003ffeffff] usable  
[ 0.000000] BIOS-e820: [mem 0x00000000003fff0000-0x00000000003ffffff] ACPI data  
[ 0.000000] BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff] reserved  
[ 0.000000] BIOS-e820: [mem 0x00000000fee00000-0x00000000fee00fff] reserved  
[ 0.000000] BIOS-e820: [mem 0x00000000fffc0000-0x00000000ffffff] reserved  
[ 0.000000] Notice: NX (Execute Disable) protection cannot be enabled: non-PAE kernel!  
[ 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 0x000a0000-0x000fffff] usable  
[ 0.000000] e820: last_pfn = 0x3fff0 max_arch_pfn = 0x100000
```

Используем поиск с помощью `dmesg | grep -i "то, что ищем"` и найдем следующее:

1. Версия ядра Linux (Linux version).
2. Частота процессора (Detected Mhz processor).
3. Модель процессора (CPU0).



```
liveuser@localhost:~  
File Edit View Search Terminal Help  
[liveuser@afkhusyainova ~]$ dmesg | grep -i "linux version"  
[ 0.000000] Linux version 3.9.5-301.fc19.i686 (mockbuild@bkernel02) (gcc version 4.8.1 20130603 (Red Hat 4.8.1-1) (GCC) ) #1 SMP Tue Jun 11 20:01:50 UTC 2013  
[liveuser@afkhusyainova ~]$ dmesg | grep -i "Mhz"  
[ 15.394552] tsc: Detected 1703.994 MHz processor  
[ 17.091335] tsc: Refined TSC clocksource calibration: 1705.477 MHz  
[ 17.109314] e1000 0000:00:03:0 eth0: (PCI:33MHz:32-bit) 08:00:27:63:be:41  
[liveuser@afkhusyainova ~]$ dmesg | grep -i "CPU0"  
[liveuser@afkhusyainova ~]$ dmesg | grep -i "CPU0"  
[ 15.476044] smpboot: CPU0: Intel(R) Core(TM) i5-8400T CPU @ 1.70GHz (fam: 06, model: 9e, stepping: 0a)  
[ 15.570233] NMI watchdog: disabled (cpu0): hardware events not enabled  
[ 20.411572] microcode: CPU0 sig=0x906ea, pf=0x2, revision=0x0
```

4. Объем доступной оперативной памяти (Memory available).

```
File Edit View Search Terminal Help
liveuser@afkhusyainova ~]$ dmesg | grep -i "Memory"
0.000000] initial memory mapped: [mem 0x00000000-0x00ffffff]
0.000000] Base memory trampoline at [c009b000] 9b000 size 16384
0.000000] init_memory_mapping: [mem 0x00000000-0x00ffffff]
0.000000] init_memory_mapping: [mem 0x36c00000-0x36ffffff]
0.000000] init_memory_mapping: [mem 0x30000000-0x36bffffff]
0.000000] init_memory_mapping: [mem 0x00100000-0x2fffffff]
0.000000] init_memory_mapping: [mem 0x37000000-0x373fdfff]
15.349445] Early memory node ranges
15.356417] PM: Registered nosave memory: 00000000009f000 - 0000000000a0000
15.356418] PM: Registered nosave memory: 0000000000a0000 - 0000000000f0000
15.356419] PM: Registered nosave memory: 0000000000f0000 - 000000000100000
15.359382] please try 'cgroup_disable=memory' option if you don't want memory
/ cgroups
15.383620] Memory: 998744k/1048512k available (5627k kernel code, 49376k reserved, 2784k data, 652k init, 143304k highmem)
15.383624] virtual kernel memory layout:
15.394741] Initializing cgroup subsys memory
16.091853] Freeing initrd memory: 28076k freed
16.112868] Non-volatile memory driver v1.3
16.421113] Freeing unused kernel memory: 652k freed
17.109978] [drm] Max dedicated hypervisor surface memory is 507904 kiB
17.112486] [TTM] Zone kernel: Available graphics memory: 442096 kiB
17.112488] [TTM] Zone highmem: Available graphics memory: 513748 kiB
```

5. Тип обнаруженного гипервизора (Hypervisor detected).

```
liveuser@afkhusyainova ~]$ dmesg | grep -i "Hypervisor detected"
0.000000] Hypervisor detected: KVM
liveuser@afkhusyainova ~]$ dmesg
0.000000] Initializing cgroup subsys cpuset
0.000000] Initializing cgroup subsys cpu
0.000000] Linux version 3.9.5-301.fc19.i686 (mockbuild@bkernel02) (gcc version 4.8.1 20130603 (Red Hat 4.8.1-1) (GCC) ) #1 SMP Tue Jun 11 20:01:50 UTC 2013
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 0x0000000000f00000-0x0000000000ffffff] reserved
0.000000] BIOS-e820: [mem 0x0000000001000000-0x0000000003ffefff] usable
```

6. Тип файловой системы корневого раздела.

7. Последовательность монтирования файловых систем.

```
[liveuser@afkhusyainova ~]$ dmesg | grep -i "Mount"
[ 15.394625] Mount-cache hash table entries: 512
[ 17.439849] EXT4-fs (dm-0): mounted filesystem with ordered data mode. Opts: (null)
[ 19.717599] EXT4-fs (dm-0): re-mounted. Opts: (null)
[liveuser@afkhusyainova ~]$
```

Вывод: В ходе данной лабораторной работы я изучила, как устанавливать операционную систему на виртуальную машину и как настраивать минимально необходимые для работы серверы.