

Лабораторная работа №1

Информационная безопасность

Леонтьева Ксения Андреевна | НПМбд-01-19

Содержание

| | | |
|---|--------------------------------|----|
| 1 | Цель работы | 4 |
| 2 | Выполнение лабораторной работы | 5 |
| 3 | Домашнее задание | 26 |
| 4 | Контрольные вопросы | 29 |
| 5 | Выводы | 32 |

Список иллюстраций

| | | |
|------|---|----|
| 2.1 | Имя и тип ОС | 5 |
| 2.2 | Объем памяти | 6 |
| 2.3 | Установка жёсткого диска | 7 |
| 2.4 | Тип жёсткого диска | 8 |
| 2.5 | Формат хранения жёсткого диска | 9 |
| 2.6 | Имя и размер файла | 10 |
| 2.7 | Выбор оптического диска | 11 |
| 2.8 | Запуск машины | 11 |
| 2.9 | Запуск машины | 12 |
| 2.10 | Выбор языка | 13 |
| 2.11 | Выбор языков раскладки | 14 |
| 2.12 | Выбор дополнительного языка | 15 |
| 2.13 | Выбор программ | 16 |
| 2.14 | Отключение KDUMP | 17 |
| 2.15 | Место установки ОС | 18 |
| 2.16 | Сеть и имя узла | 19 |
| 2.17 | Пароль для root | 20 |
| 2.18 | Создание пользователя | 21 |
| 2.19 | Завершение установки | 22 |
| 2.20 | Вход в систему | 23 |
| 2.21 | Удаление устройства | 23 |
| 2.22 | Подключение образа диска дополнений гостевой ОС | 24 |
| 2.23 | Запуск образ диска дополнений гостевой ОС | 24 |
| 3.1 | Команда <code>sudo dmesg</code> | 26 |
| 3.2 | Команда <code>sudo dmesg less</code> | 27 |
| 3.3 | Команда <code>sudo dmesg less</code> | 27 |
| 3.4 | Поиск информации с помощью <code>grep</code> | 28 |
| 3.5 | Поиск информации с помощью <code>grep</code> | 28 |

1 Цель работы

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

2 Выполнение лабораторной работы

Для начала с официального сайта была скачана и установлена VirtualBox.

Далее запускаем VirtualBox, выбираем “Создать”. В появившемся окне указываем имя ОС (kaleontjeva) и тип ОС (Linux, Red Hat (64-bit)) (рис. 2.1).

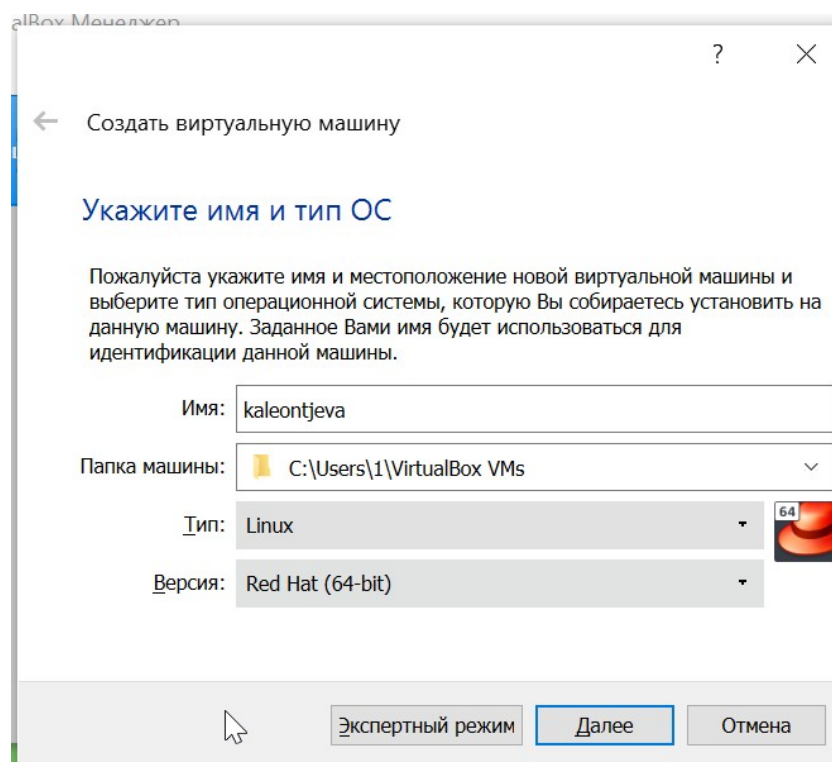


Рис. 2.1: Имя и тип ОС

Указываем объём памяти - 2048 МБ (рис. 2.2).

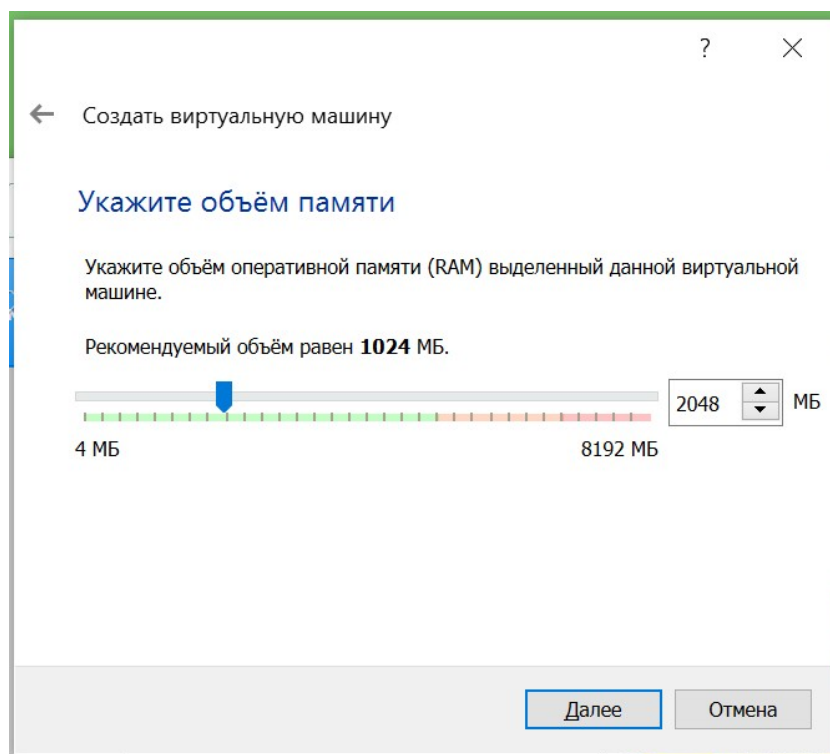


Рис. 2.2: Объем памяти

Создаем новый динамический виртуальный жёсткий диск: задаем его тип - VDI, формат хранения - динамический и размер файла - 40 ГБ (рис. 2.3-2.6).

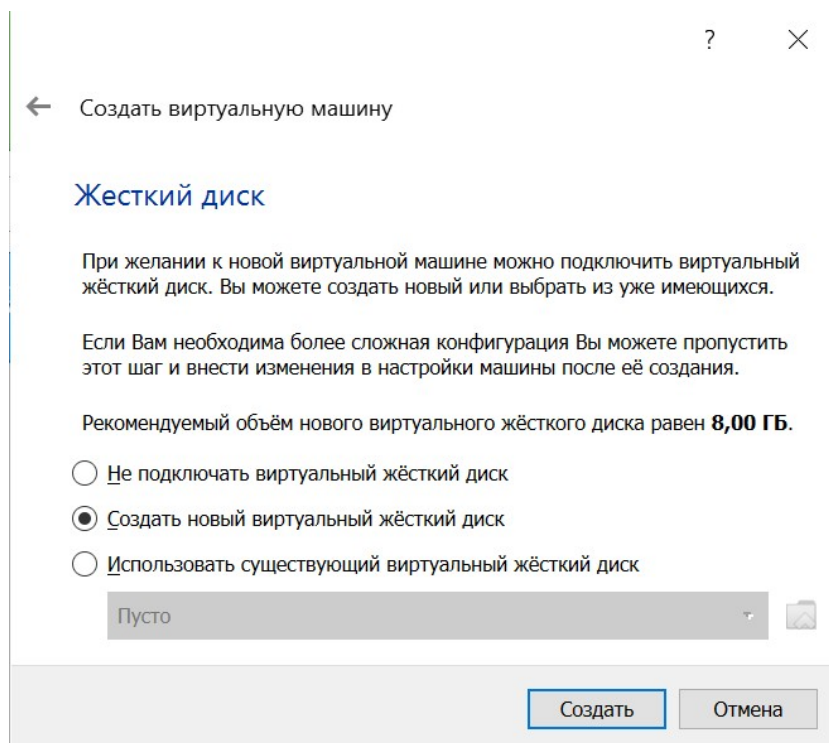


Рис. 2.3: Установка жёсткого диска

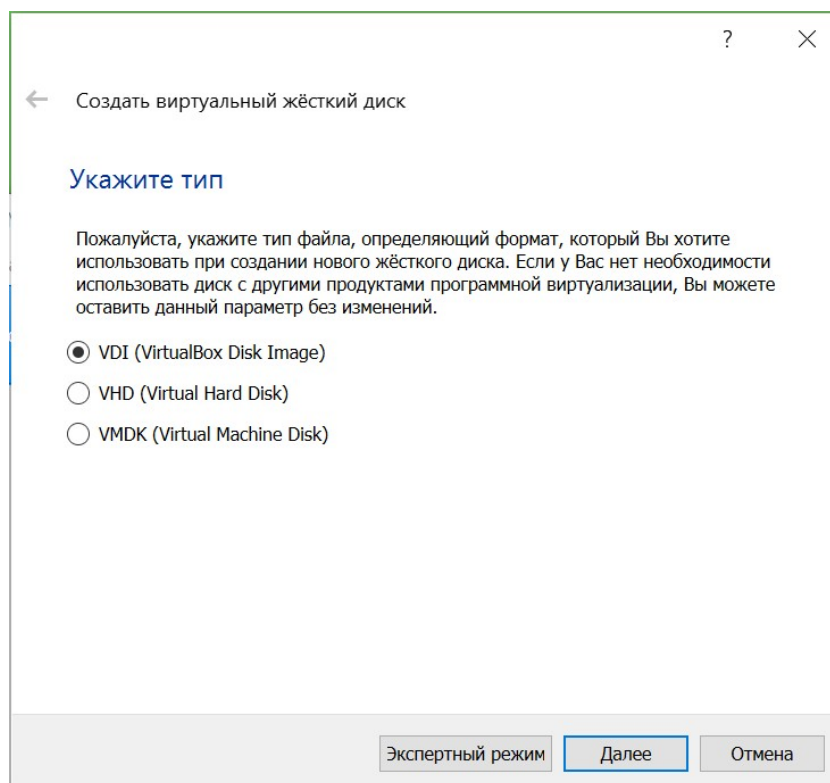


Рис. 2.4: Тип жёсткого диска

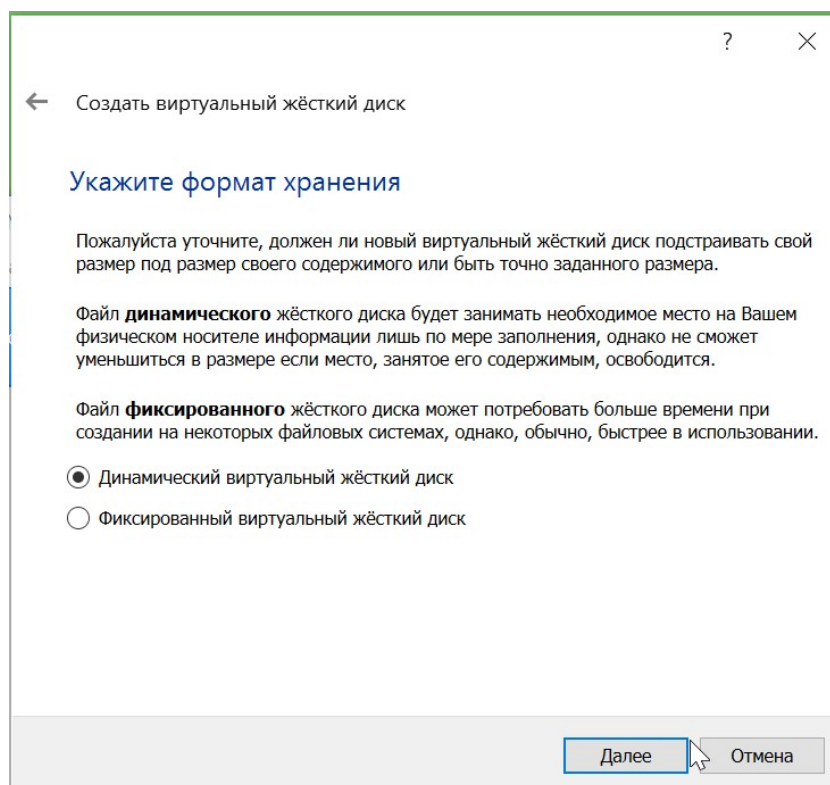


Рис. 2.5: Формат хранения жёсткого диска

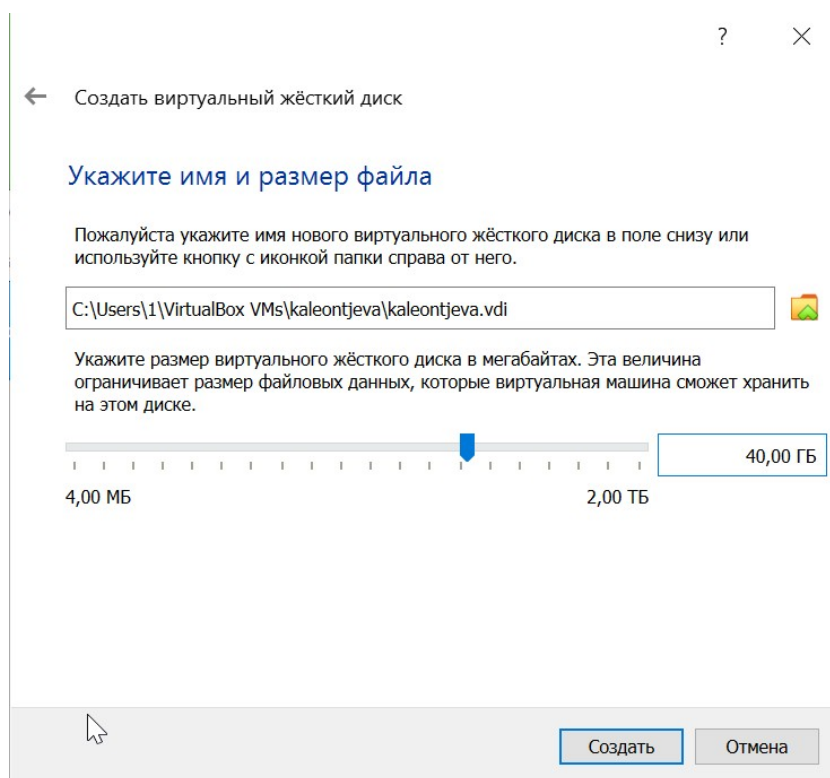


Рис. 2.6: Имя и размер файла

Затем нажимаем “Настроить”, переходим в раздел “Носители” и выбираем оптический диск - ранее скачанный с официального сайта дистрибутив “Rocky” (рис. 2.7).

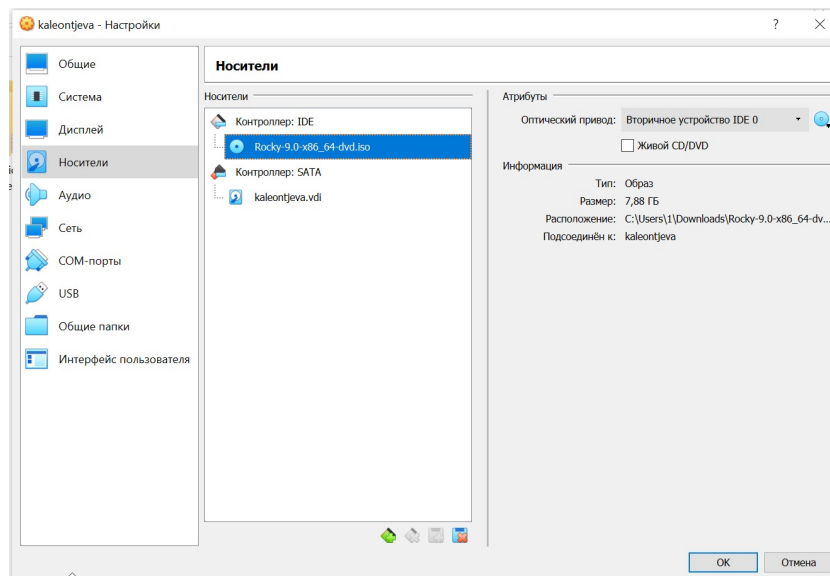


Рис. 2.7: Выбор оптического диска

Теперь запускаем виртуальную машину ((рис. 2.8, 2.9).

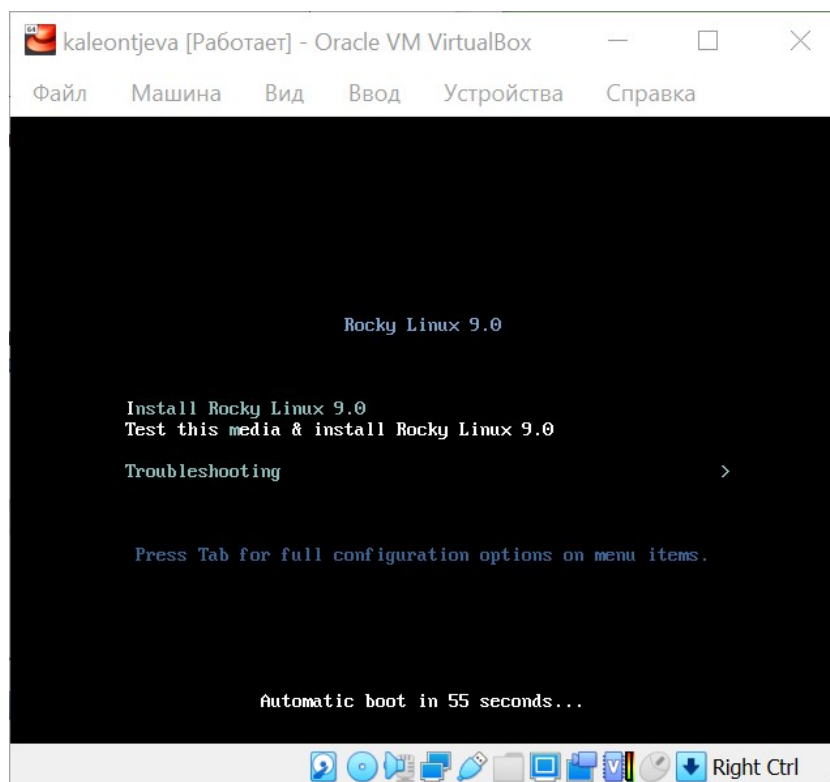


Рис. 2.8: Запуск машины

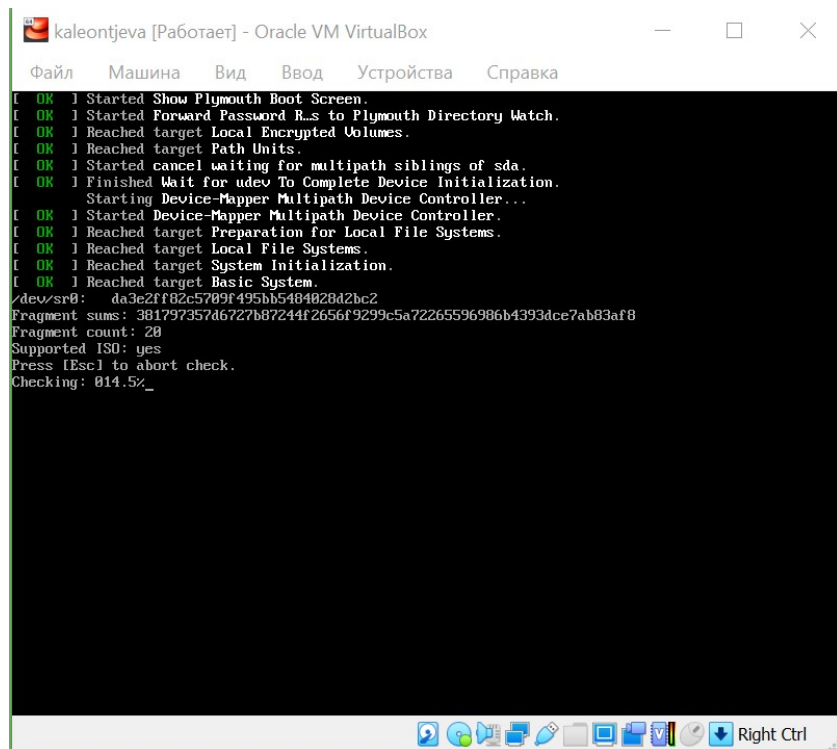


Рис. 2.9: Запуск машины

Переходим к настройке машины. Выбираем английский язык (рис. 2.10).

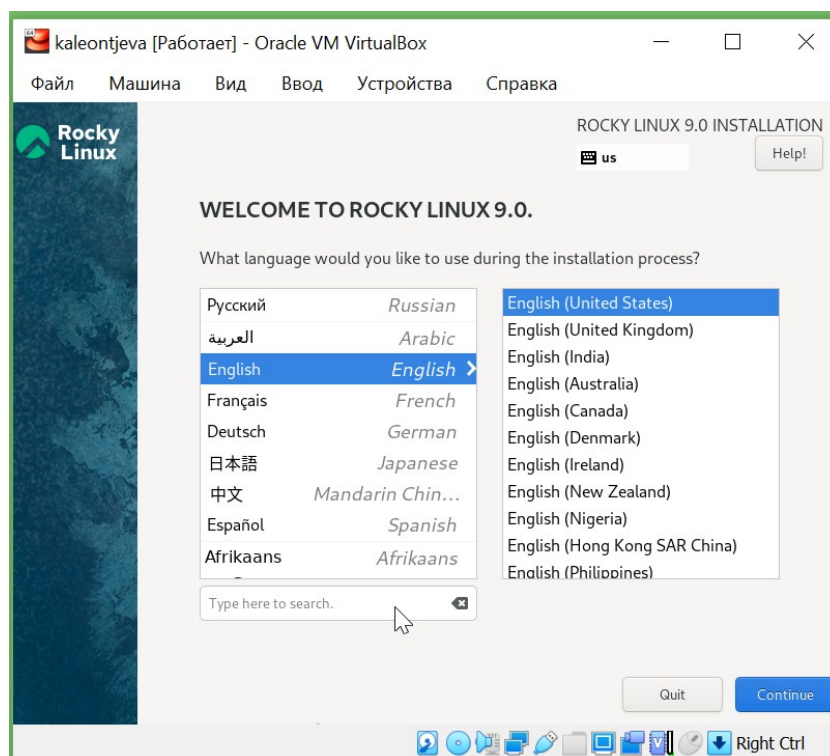


Рис. 2.10: Выбор языка

Выбираем языки раскладки и комбинацию клавиш для переключения между ними(рис. 2.11).

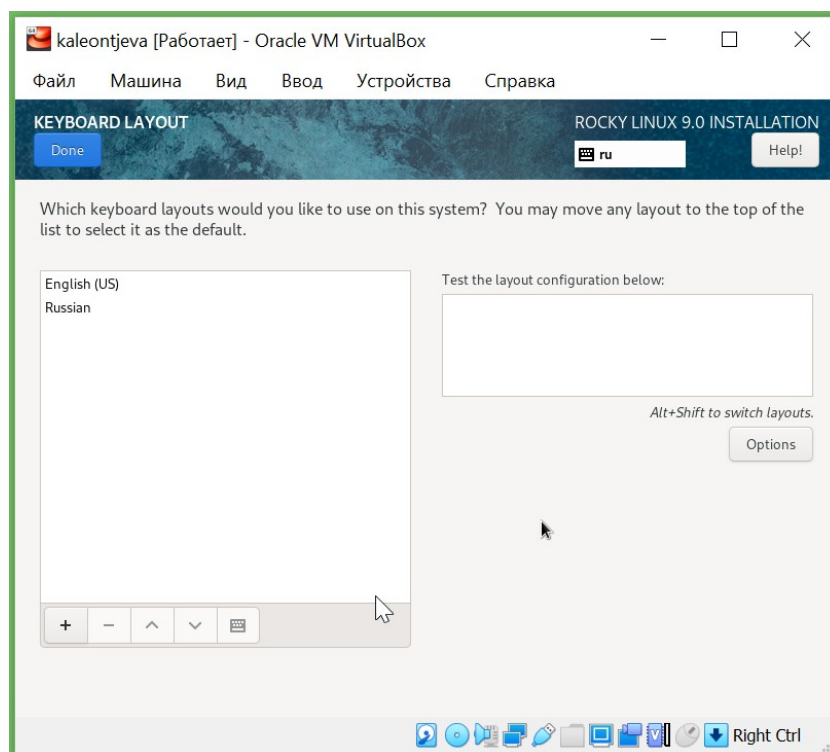


Рис. 2.11: Выбор языков раскладки

Выбираем дополнительный язык - русский (рис. 2.12).

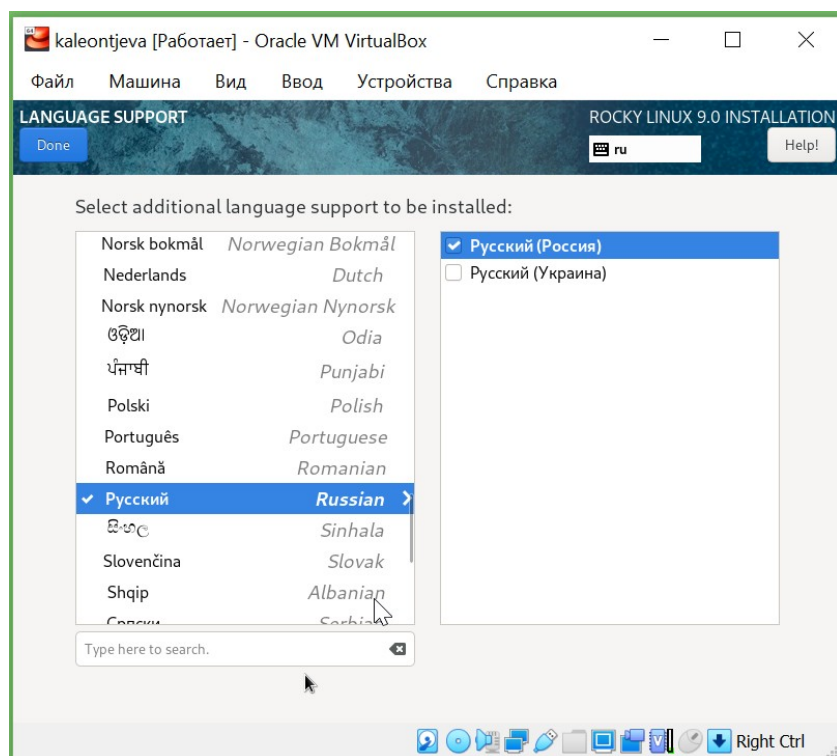


Рис. 2.12: Выбор дополнительного языка

Выбираем программы: базовое окружение Server with GUI и дополнение Development Tools (рис. 2.13).

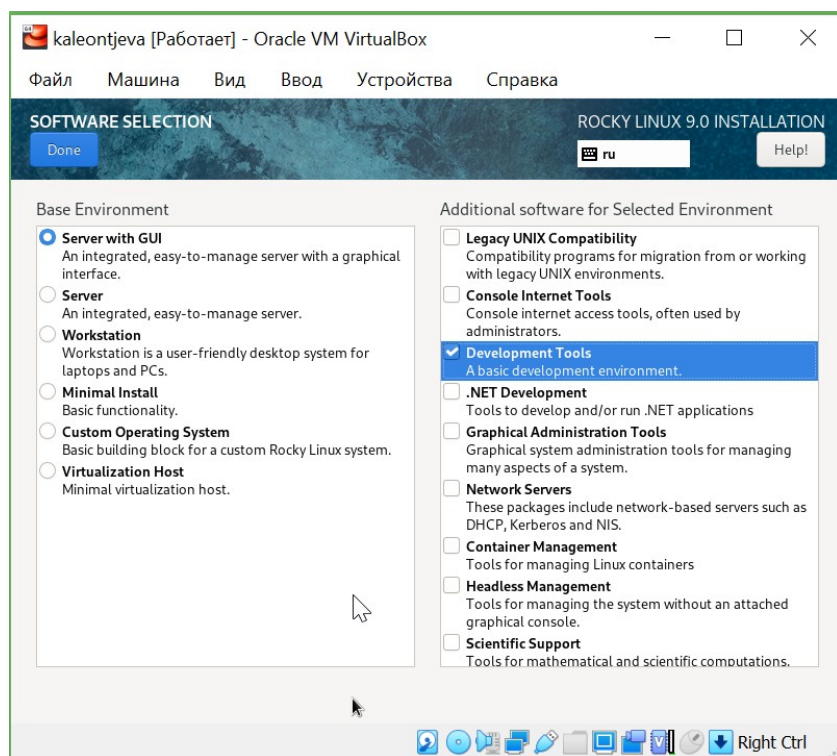


Рис. 2.13: Выбор программ

Отключаем KDUMP (рис. 2.14).

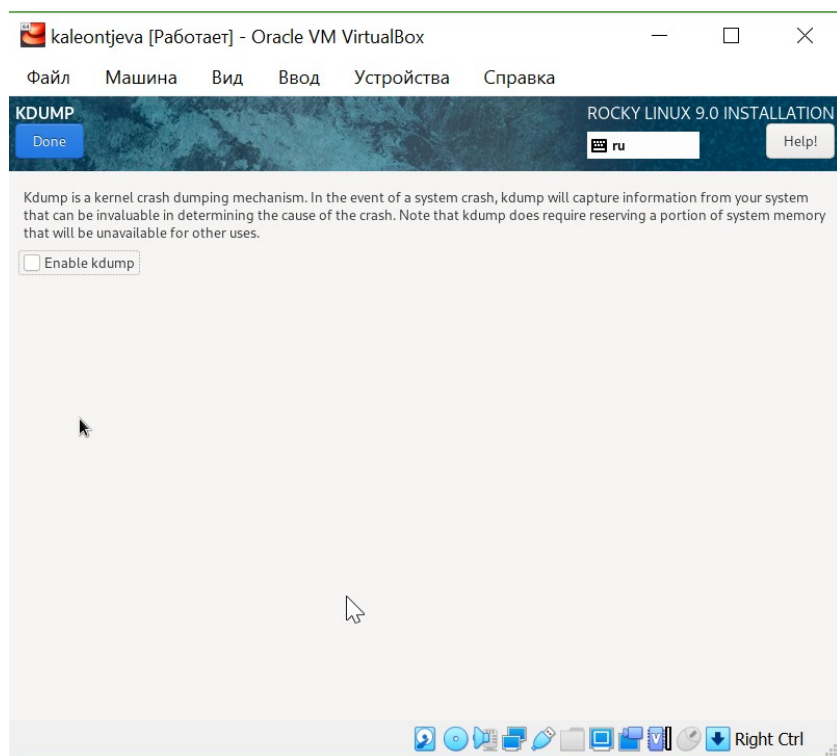


Рис. 2.14: Отключение KDUMP

Место установки ОС оставляем без изменения (рис. 2.15).

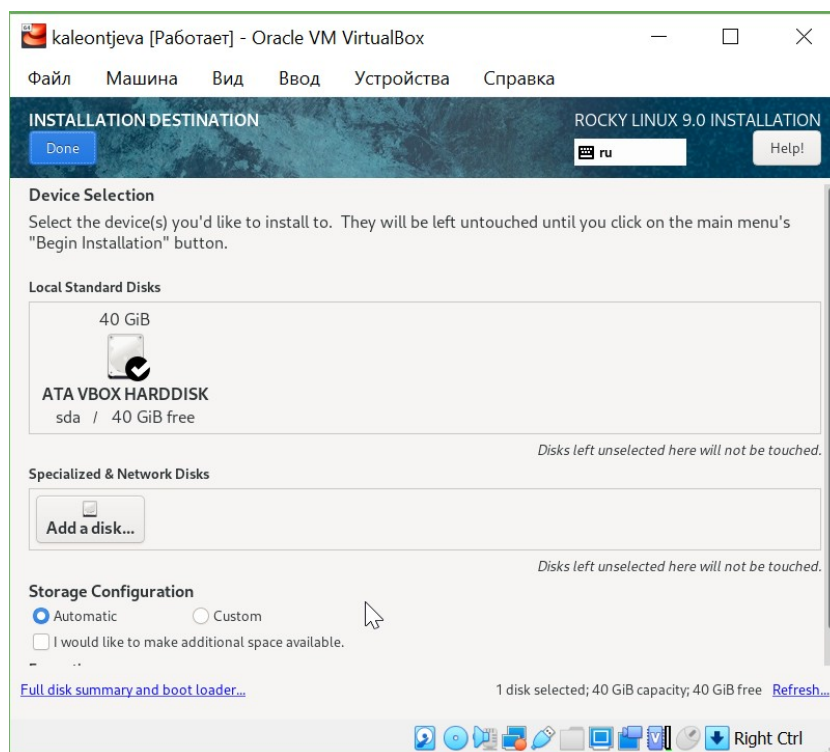


Рис. 2.15: Место установки ОС

Включаем сетевое соединение и в качестве имени узла указываем `kaleontjeva.localdomain` (рис. 2.16).

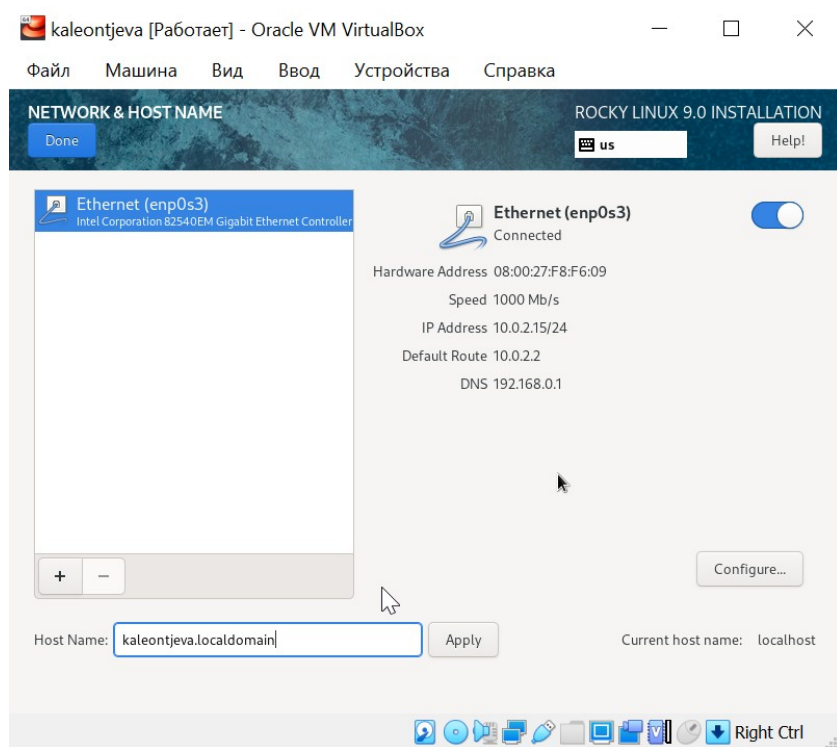


Рис. 2.16: Сеть и имя узла

Устанавливаем пароль для root (рис. 2.17).

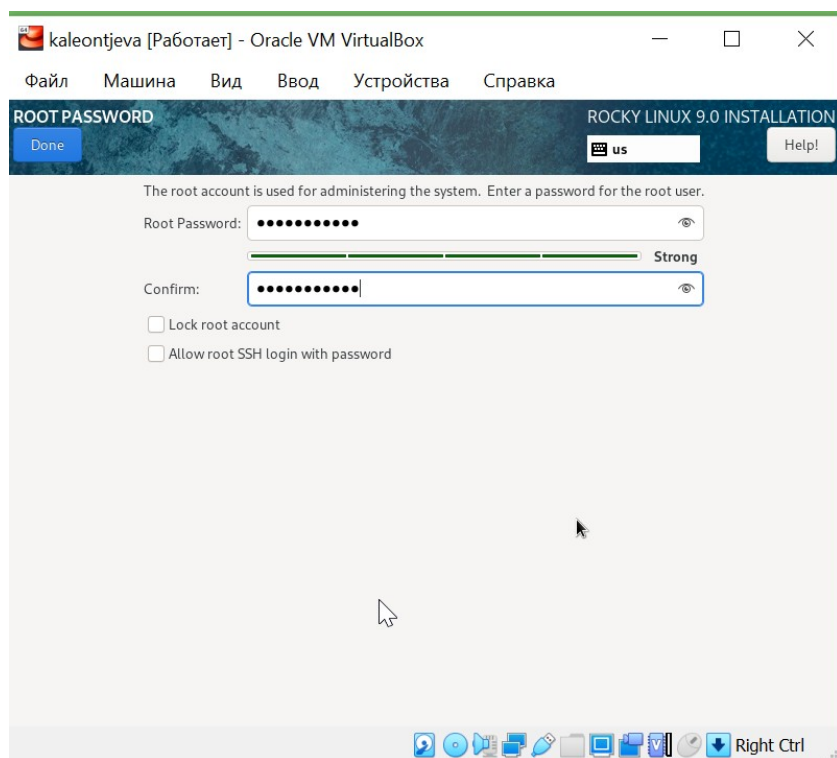


Рис. 2.17: Пароль для root

Создаем пользователя с правами администратора (рис. 2.18).

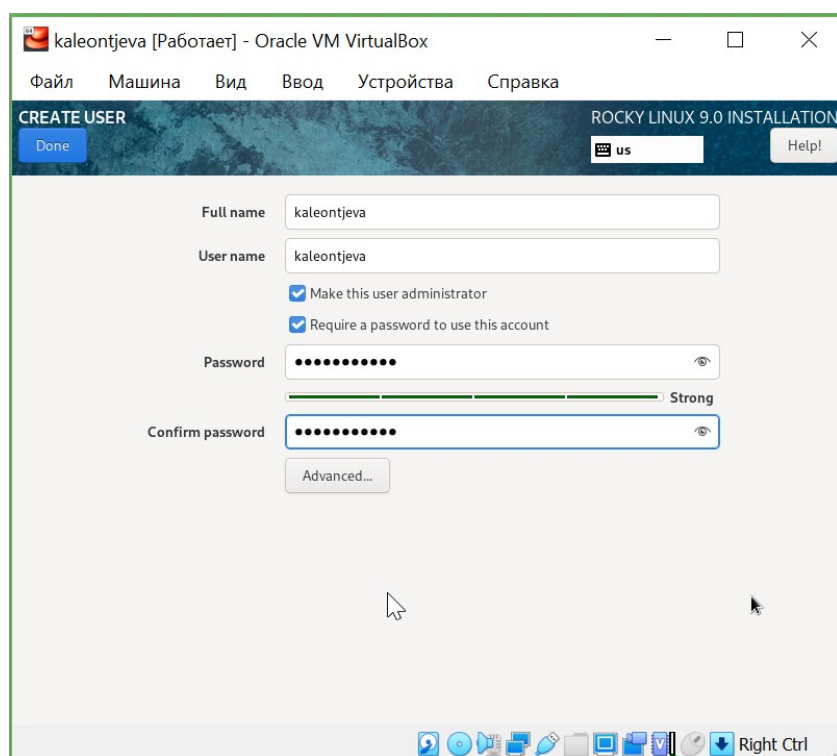


Рис. 2.18: Создание пользователя

Завершаем установку операционной системы, корректно перезагружаем виртуальную машину (рис. 2.19).

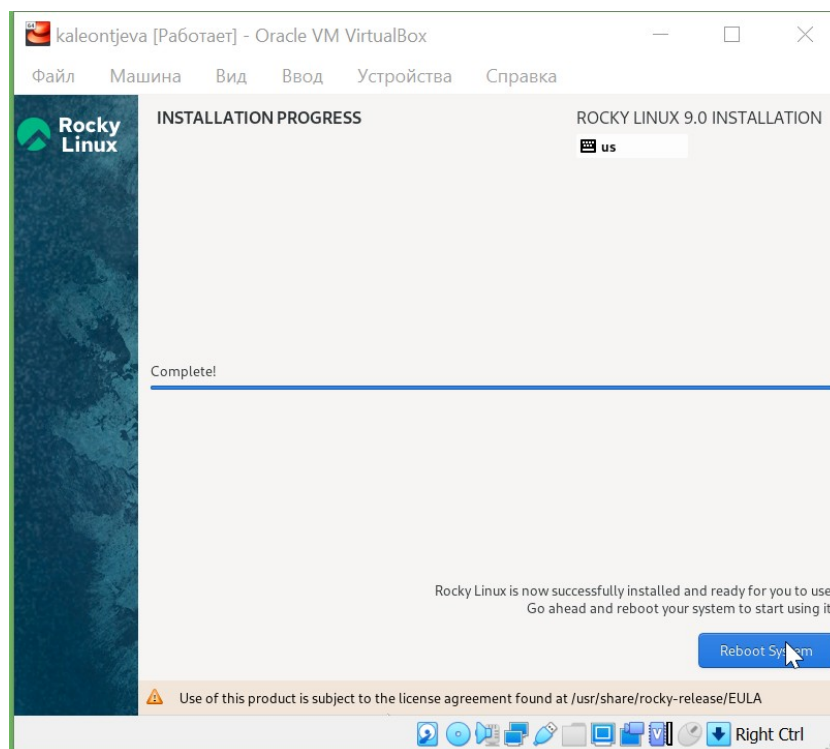


Рис. 2.19: Завершение установки

Теперь можно войти в систему, введя пароль (рис. 2.20).

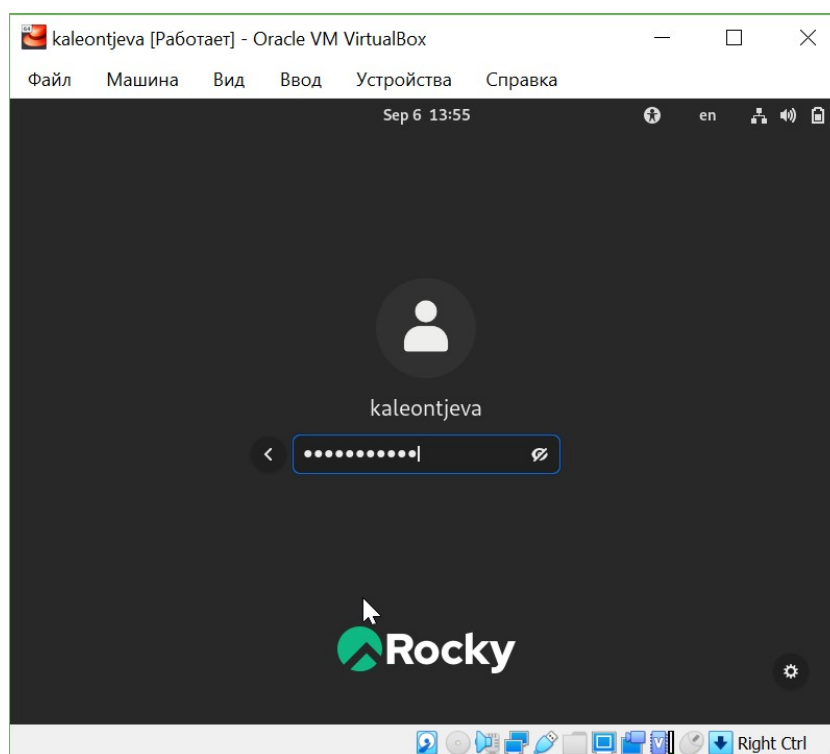


Рис. 2.20: Вход в систему

Чтобы подключить образ диска дополнений гостевой ОС, сначала удаляем устройство в разделе “Носители” и оставляем диск пустым (рис. 2.21).

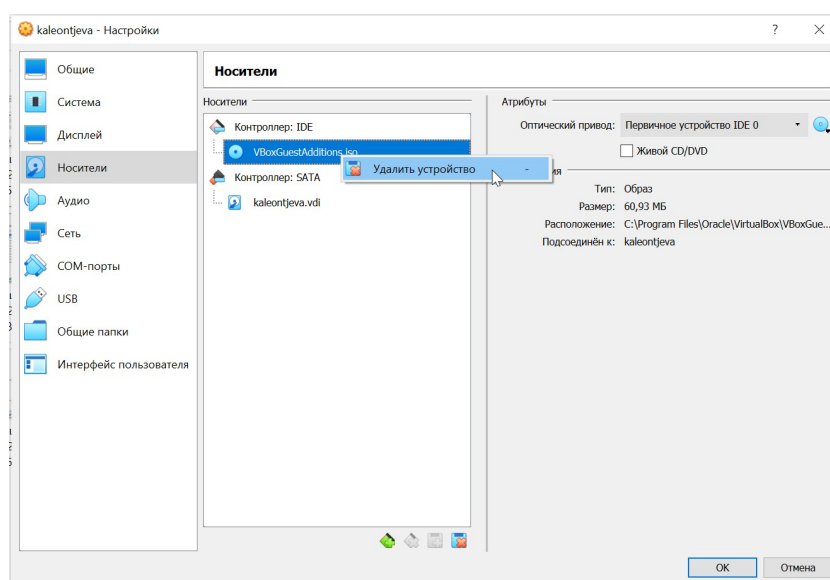


Рис. 2.21: Удаление устройства

Затем в разделе “Устройства” выбираем “Подключить образ диска дополнений гостевой ОС” (рис. 2.22).

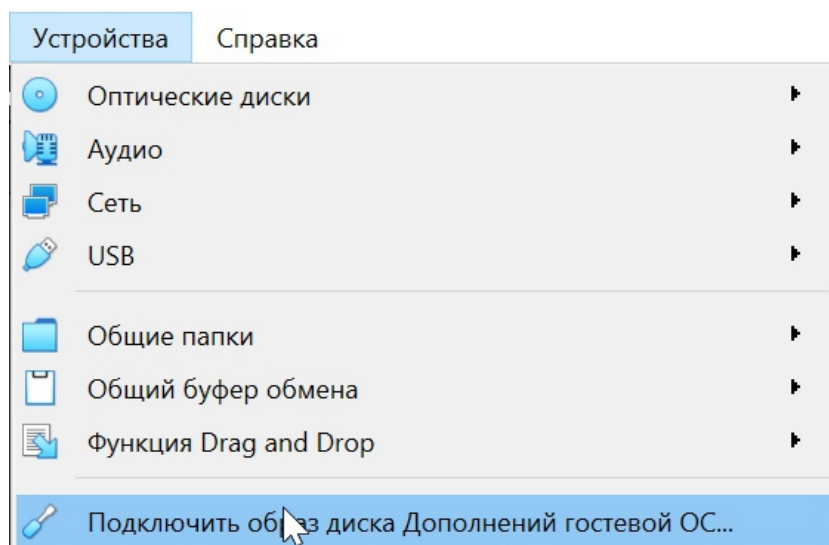


Рис. 2.22: Подключение образа диска дополнений гостевой ОС

Запускаем образ диска дополнений гостевой ОС (рис. 2.23).

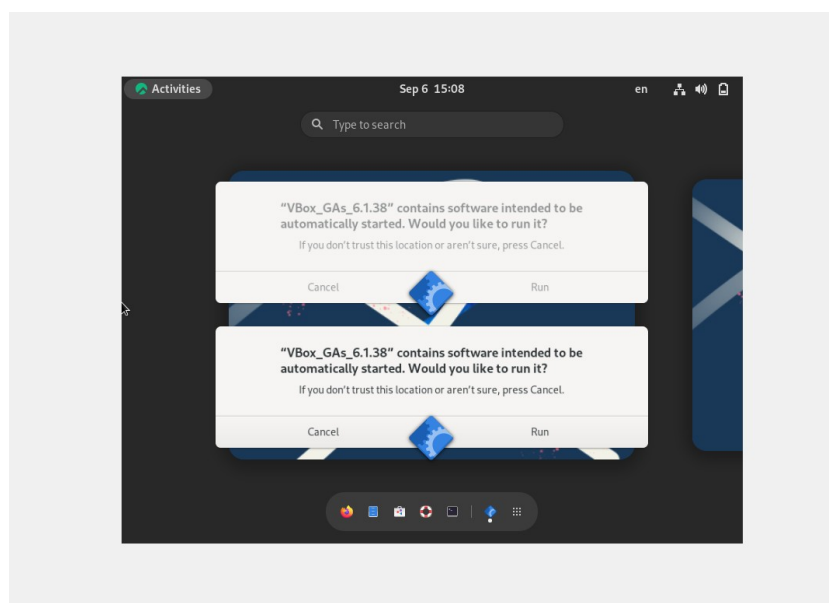


Рис. 2.23: Запуск образ диска дополнений гостевой ОС

После загрузки дополнений нажимаем Enter и корректно перезапускаем виртуальную машину.

Таким образом, установили операционную систему Linux с дистрибутивом Rocky, а также дополнения гостевой ОС, благодаря которым теперь не нужно нажимать хост-клавишу для переключения мышки между двумя ОС и можно настроить разрешение экрана.

3 Домашнее задание

Загружаем графическое окружение и открываем консоль. Анализируем последовательность загрузки системы, используя команду “`sudo dmesg`” и введя пароль (рис. 3.1).

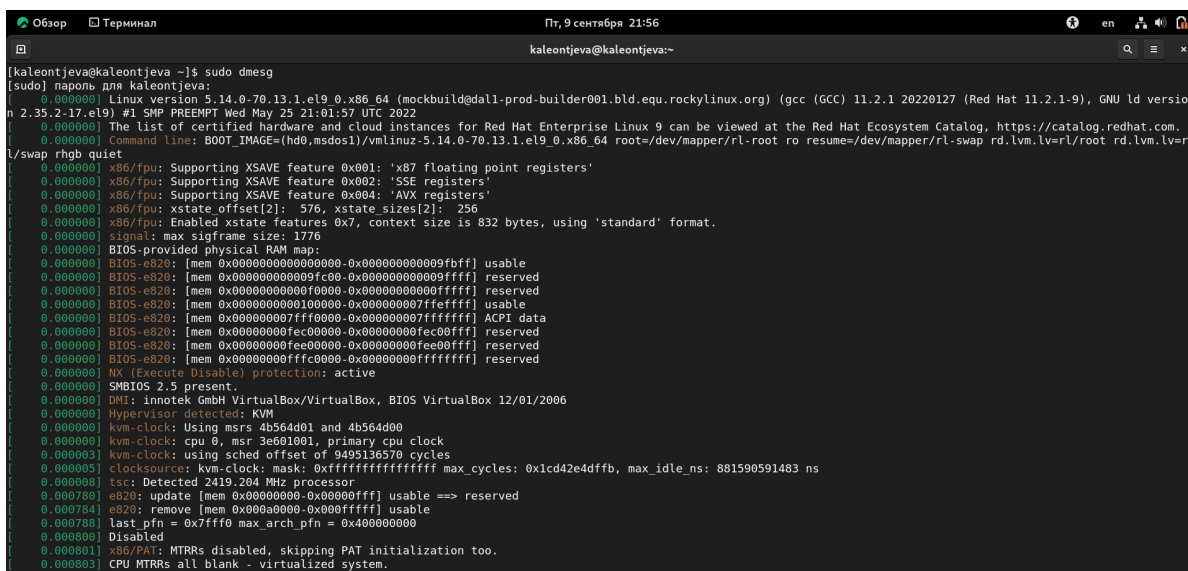


Рис. 3.1: Команда `sudo dmesg`

Смотрим вывод этой команды, выполнив “`sudo dmesg | less`” (рис. 3.2, 3.3). В данном случае после каждого нажатия клавиши “Enter” в консоли отображается только одна команда.



Рис. 3.2: Команда `sudo dmesg | less`

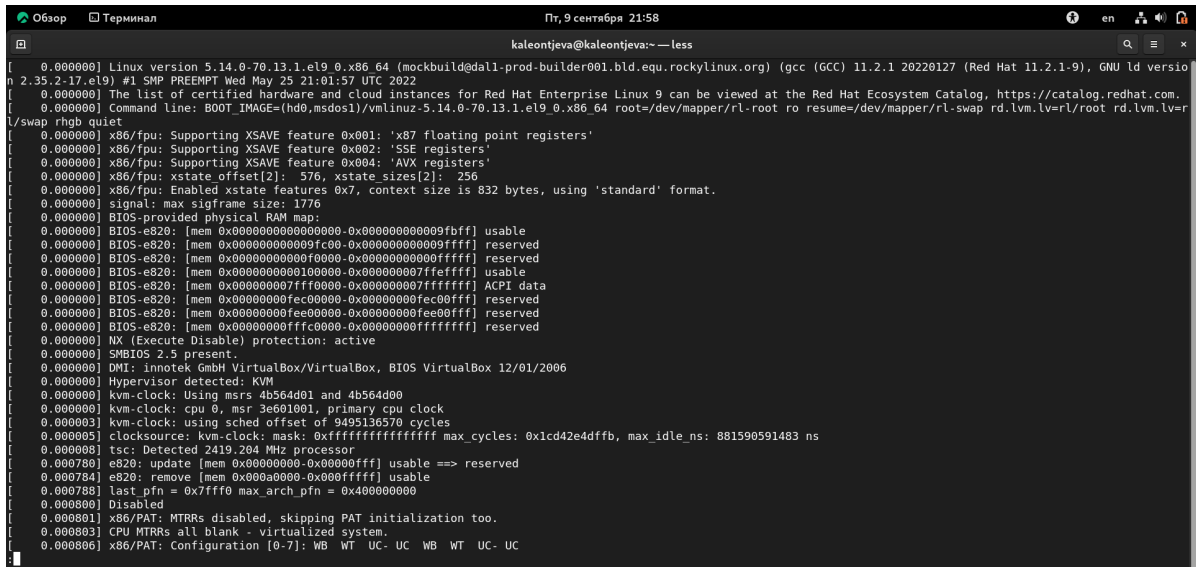
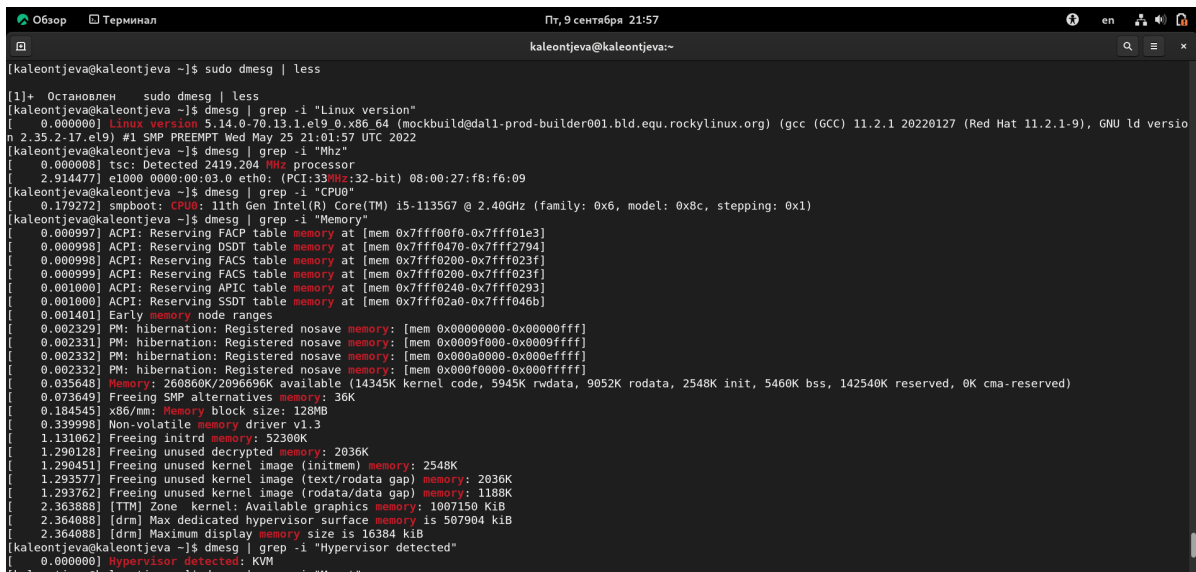


Рис. 3.3: Команда `sudo dmesg | less`

Далее получаем следующую информацию (рис. 3.4, 3.5).

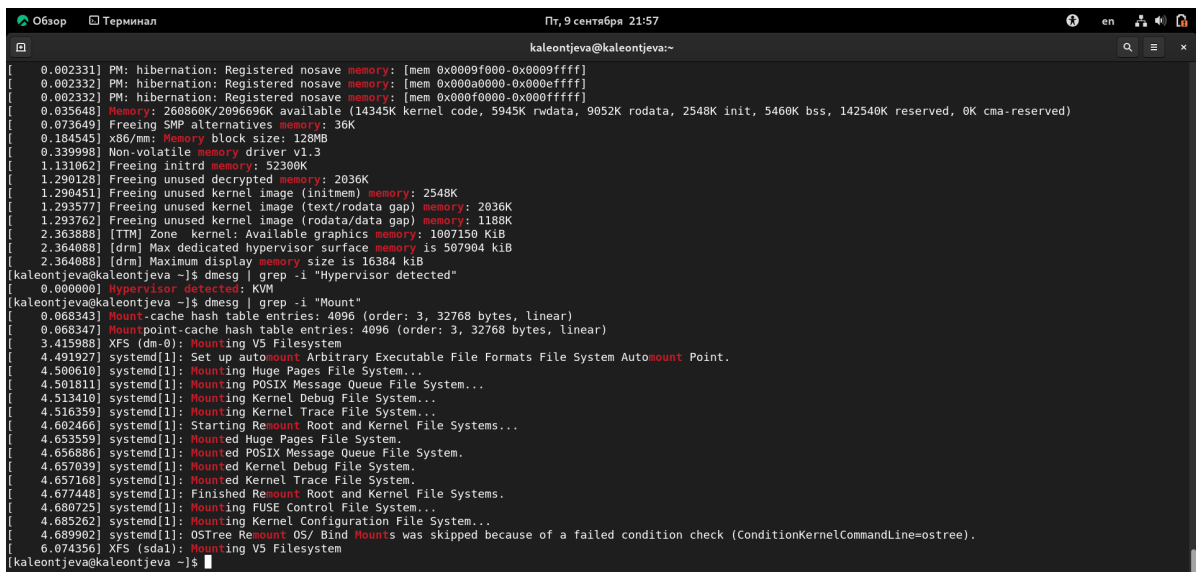
1. Версия ядра Linux: `dmesg | grep -i "Linux version"`. Ответ: 5.14.0-70.13.1.el9_0.x86_64
2. Частота процессора: `dmesg | grep -i "Mhz"`. Ответ: 2419.204 MHz
3. Модель процессора: `dmesg | grep -i "CPU0"`. Ответ: 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz
4. Объём доступной оперативной памяти: `dmesg | grep -i "Memory"`. Ответ: 2096696K
5. Тип обнаруженного гипервизора: `dmesg | grep -i "Hypervisor detected"`. Ответ: KVM

6. Тип файловой системы корневого раздела и последовательность монтирования файловых систем: `dmesg | grep -i "Mount"`. Ответ: XFS



```
kaleontjeva@kaleontjeva ~]$ sudo dmesg | less
[1]+  Остановлен  sudo dmesg | less
kaleontjeva@kaleontjeva ~]$ dmesg | grep -i "Linux version"
[ 0.000000] Linux version 5.14.0-70.13.1.el9_0.x86_64 (mockbuild@dal1-prod-builder001.bld.equ.rockylinux.org) (gcc (GCC) 11.2.1 20220127 (Red Hat 11.2.1-9), GNU ld versio
n 2.35.2-17.el9) #1 SMP PREEMPT Wed May 25 21:01:57 UTC 2022
kaleontjeva@kaleontjeva ~]$ dmesg | grep -i "Mhz"
[ 0.000008] tsc: Detected 2419.204 Mhz processor
[ 2.914477] cpuid 0000:00:03.0 eth0: (PCI:33MHz:32-bit) 08:00:27:f8:f6:09
kaleontjeva@kaleontjeva ~]$ dmesg | grep -i "CPU0"
[ 0.179272] smpboot: CPU0: 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz (family: 0x6, model: 0x8c, stepping: 0x1)
kaleontjeva@kaleontjeva ~]$ dmesg | grep -i "Memory"
[ 0.000997] ACPI: Reserving FACP table memory at [mem 0x7fff00f0-0x7fff01e3]
[ 0.000998] ACPI: Reserving DSDT table memory at [mem 0x7fff0470-0x7fff0794]
[ 0.000999] ACPI: Reserving FACS table memory at [mem 0x7fff0200-0x7fff023f]
[ 0.000999] ACPI: Reserving FACS table memory at [mem 0x7fff0200-0x7fff023f]
[ 0.001000] ACPI: Reserving APIC table memory at [mem 0x7fff0240-0x7fff0293]
[ 0.001000] ACPI: Reserving SSDT table memory at [mem 0x7fff02a0-0x7fff046b]
[ 0.001401] Early memory node ranges
[ 0.002329] PM: hibernation: Registered nosave memory: [mem 0x00000000-0x00000fff]
[ 0.002331] PM: hibernation: Registered nosave memory: [mem 0x0009f000-0x0009ffff]
[ 0.002332] PM: hibernation: Registered nosave memory: [mem 0x000a0000-0x000effff]
[ 0.002332] PM: hibernation: Registered nosave memory: [mem 0x000f0000-0x000fffff]
[ 0.035648] Memory: 260860K/209669K available (14345K kernel code, 5945K rwdats, 9052K rodata, 2548K init, 5460K bss, 142540K reserved, 0K cma-reserved)
[ 0.073649] Freeing SMP alternatives memory: 36K
[ 0.184545] x86/mm: Memory block size: 128MB
[ 0.339998] Non-volatile memory driver v1.3
[ 1.131062] Freeing initrd memory: 52300K
[ 1.290128] Freeing unused decrypted memory: 2036K
[ 1.290451] Freeing unused kernel image (initmem) memory: 2548K
[ 1.293577] Freeing unused kernel image (text/rodata gap) memory: 2036K
[ 1.293762] Freeing unused kernel image (rodata/data gap) memory: 1188K
[ 2.363888] [TTM] Zone kernel: Available graphics memory: 1007150 KiB
[ 2.364088] [drm] Max dedicated hypervisor surface memory is 507904 kiB
[ 2.364088] [drm] Maximum display memory size is 16384 kiB
kaleontjeva@kaleontjeva ~]$ dmesg | grep -i "Hypervisor detected"
[ 0.000000] Hypervisor detected: KVM
kaleontjeva@kaleontjeva ~]$
```

Рис. 3.4: Поиск информации с помощью grep



```
kaleontjeva@kaleontjeva ~]$ dmesg | grep -i "Mount"
[ 0.002331] PM: hibernation: Registered nosave memory: [mem 0x0009f000-0x0009ffff]
[ 0.002332] PM: hibernation: Registered nosave memory: [mem 0x000a0000-0x000effff]
[ 0.002332] PM: hibernation: Registered nosave memory: [mem 0x000f0000-0x000fffff]
[ 0.035648] Memory: 260860K/209669K available (14345K kernel code, 5945K rwdats, 9052K rodata, 2548K init, 5460K bss, 142540K reserved, 0K cma-reserved)
[ 0.073649] Freeing SMP alternatives memory: 36K
[ 0.184545] x86/mm: Memory block size: 128MB
[ 0.339998] Non-volatile memory driver v1.3
[ 1.131062] Freeing initrd memory: 52300K
[ 1.290128] Freeing unused decrypted memory: 2036K
[ 1.290451] Freeing unused kernel image (initmem) memory: 2548K
[ 1.293577] Freeing unused kernel image (text/rodata gap) memory: 2036K
[ 1.293762] Freeing unused kernel image (rodata/data gap) memory: 1188K
[ 2.363888] [TTM] Zone kernel: Available graphics memory: 1007150 KiB
[ 2.364088] [drm] Max dedicated hypervisor surface memory is 507904 kiB
[ 2.364088] [drm] Maximum display memory size is 16384 kiB
kaleontjeva@kaleontjeva ~]$ dmesg | grep -i "Hypervisor detected"
[ 0.000000] Hypervisor detected: KVM
kaleontjeva@kaleontjeva ~]$ dmesg | grep -i "Mount"
[ 0.068343] Mount-cache hash table entries: 4096 (order: 3, 32768 bytes, linear)
[ 0.068347] Mountpoint-cache hash table entries: 4096 (order: 3, 32768 bytes, linear)
[ 3.415988] XFS (dm-0): Mounting V5 Filesystem
[ 4.491927] systemd[1]: Set up automount Arbitrary Executable File Formats File System Automount Point.
[ 4.506101] systemd[1]: Mounting Huge Pages File System...
[ 4.501011] systemd[1]: Mounting POSIX Message Queue File System...
[ 4.513410] systemd[1]: Mounting Kernel Debug File System...
[ 4.516359] systemd[1]: Mounting Kernel Trace File System...
[ 4.602466] systemd[1]: Starting Remount Root and Kernel File Systems...
[ 4.653559] systemd[1]: Mounted Huge Pages File System.
[ 4.656886] systemd[1]: Mounted POSIX Message Queue File System.
[ 4.657039] systemd[1]: Mounted Kernel Debug File System.
[ 4.657168] systemd[1]: Mounted Kernel Trace File System.
[ 4.677448] systemd[1]: Finished Remount Root and Kernel File Systems.
[ 4.680725] systemd[1]: Mounting FUSE Control File System...
[ 4.685262] systemd[1]: Mounting Kernel Configuration File System...
[ 4.689902] systemd[1]: OSTree Remount OS/ Bind Mounts was skipped because of a failed condition check (ConditionKernelCommandLine=ostree).
[ 6.074356] XFS (sdal): Mounting V5 Filesystem
kaleontjeva@kaleontjeva ~]$
```

Рис. 3.5: Поиск информации с помощью grep

4 Контрольные вопросы

1. Учетная запись пользователя - это необходимая для системы информация о пользователе, которая хранится в специальных файлах. Вся информация о пользователе обычно хранится в файлах `/etc/passwd` и `/etc/group`. Учетная запись пользователя содержит: имя пользователя (user name), идентификационный номер группы (GID), идентификационный номер пользователя (UID), пароль (password), полное имя (full name), домашний каталог (home directory), начальную оболочку (login shell).
2. Команды терминала:
 1. Для получения справки по команде: `man` команда. Например, команда `"man ls"` выведет справку о команде `"ls"`.
 2. Для перемещения по файловой системе: `cd` путь. Например, команда `"cd newdir"` осуществляет переход в каталог `newdir`.
 3. Для просмотра содержимого каталога: `ls` опции путь. Например, команда `"ls -a ~/newdir"` отобразит имена скрытых файлов в каталоге `newdir`.
 4. Для определения объёма каталога: `du` опция путь. Например, команда `"du -k ~/newdir"` выведет размер каталога `newdir` в килобайтах.
 5. 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`.

6. 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`.
 7. 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 жёстком диске, с другой - это организация каталогов в соответствии с идеологией Linux. Файловая система - это архитектура хранения данных в системе, хранение данных в оперативной памяти и доступа к конфигурации ядра. 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. Примеры файловых систем:
1. XFS рассчитана на файлы большого размера, поддерживает диски до 2 терабайт. Преимущества: высокая скорость работы с большими файлами, отложенное выделение места, увеличение разделов на лету, незначительный размер служебной информации. Недостатки: невозможность уменьшения размера, сложность восстановления данных и риск потери файлов при аварийном отключении питания.
 2. Ext2, Ext3, Ext4 или Extended Filesystem - стандартная файловая система, первоначально разработанная еще для Minix. Содержит максимальное количество функций и является наиболее стабильной в связи с редкими изменениями кодовой базы. Начиная с ext3 в системе используется функция журналирования. Сегодня версия ext4 присутствует во всех дистрибутивах Linux.
 3. JFS или Journaled File System разработана в IBM в качестве альтернативы для файловых систем ext. Сейчас используется там, где необходима высокая стабильность и минимальное потребление ресурсов (в первую очередь в многопроцессорных компьютерах). В журнале хранятся толь-

ко метаданные, что позволяет восстанавливать старые версии файлов после сбоев.

4. Команда “findmnt” или “findmnt –all” будет отображать все подмонтированные файловые системы или искать файловую систему.
5. Команда “kill -сигнал pid_процесса” позволяет удалить зависший процесс, где PID - уникальный идентификатор процесса. Сигналы могут быть следующие:
 1. 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
 2. SIGQUIT - 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+/
 3. SIGHUP - сообщает процессу, что соединение с управляющим терминалом разорвано, отправляется, в основном, системой при разрыве соединения с интернетом.
 4. SIGTERM - немедленно завершает процесс, но обрабатывается программой, поэтому позволяет ей завершить дочерние процессы и освободить все ресурсы
 5. 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

5 Выводы

В ходе выполнения данной лабораторной работы я приобрела практические навыки установки операционной системы на виртуальную машину и настройки минимально необходимых для дальнейшей работы сервисов.