

Отчет по лабораторной работе №1

**Установка и конфигурация операционной системы на виртуальную
машину**

Бурдина Ксения Павловна

2022 Sep 10th

Содержание

1	Цель работы	5
2	Ход выполнения лабораторной работы	6
3	Выводы	25
4	Список литературы	26
5	Ответы на контрольные вопросы	27

List of Figures

2.1	рис 1. Создание новой машины	6
2.2	рис 2. Задание имени и типа ОС	7
2.3	рис 3. Размер памяти ВМ	7
2.4	рис 4. Конфигурация жесткого диска	8
2.5	рис 5. Тип жесткого диска	8
2.6	рис 6. Формат хранения	9
2.7	рис 7. Имя и размер файла	9
2.8	рис 8. Созданная виртуальная машина	10
2.9	рис 9. Добавление образа ОС	11
2.10	рис 10. Запуск машины	12
2.11	рис 11. Выбор языка интерфейса	13
2.12	рис 12. Настройки ОС	14
2.13	рис 13. Выбор программ	15
2.14	рис 14. Отключение KDUMP	15
2.15	рис 15. Место установки ОС	16
2.16	рис 16. Имя узла	17
2.17	рис 17. Пароль для root	17
2.18	рис 18. Пароль для администратора	18
2.19	рис 19. Завершение установки ОС	19
2.20	рис 20. Установка ОС	20
2.21	рис 21. Версия ядра Linux	21
2.22	рис 22. Частота процессора	22
2.23	рис 23. Модель процессора	22
2.24	рис 24. Доступная оперативная память	22
2.25	рис 25. Файловые системы	23
2.26	рис 26. Тип ФС корневого каталога	23
2.27	рис 27. Последовательность монтирования ФС	24

List of Tables

1 Цель работы

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

2 Ход выполнения лабораторной работы

1. Запустим VirtualBox и нажмем на создание новой виртуальной машины:

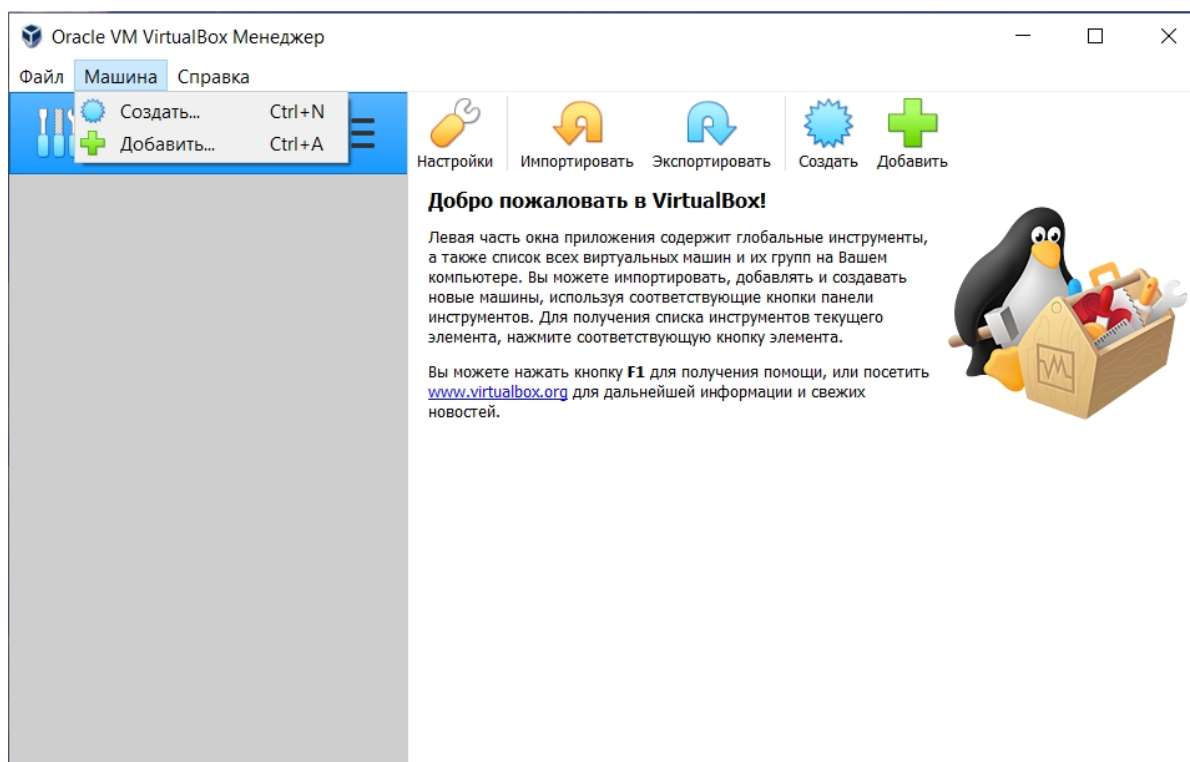


Figure 2.1: рис 1. Создание новой машины

2. Укажем имя виртуальной машины и тип операционной системы:

← 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:




Figure 2.2: рис 2. Задание имени и типа ОС

3. Укажем размер основной памяти виртуальной машины, равный 2048 МБ:

← Создать виртуальную машину

Укажите объём памяти

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

Рекомендуемый объём равен **1024** МБ.

2048 МБ

4 МБ 8192 МБ

Figure 2.3: рис 3. Размер памяти VM

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

← Создать виртуальную машину

Жесткий диск

При желании к новой виртуальной машине можно подключить виртуальный жёсткий диск. Вы можете создать новый или выбрать из уже имеющихся.

Если Вам необходима более сложная конфигурация Вы можете пропустить этот шаг и внести изменения в настройки машины после её создания.

Рекомендуемый объём нового виртуального жёсткого диска равен **8,00 ГБ**.

- ☐ Не подключать виртуальный жёсткий диск
- ☒ Создать новый виртуальный жёсткий диск
- ☐ Использовать существующий виртуальный жёсткий диск

Пусто  

Figure 2.4: рис 4. Конфигурация жесткого диска

← Создать виртуальный жёсткий диск

Укажите тип

Пожалуйста, укажите тип файла, определяющий формат, который Вы хотите использовать при создании нового жёсткого диска. Если у Вас нет необходимости использовать диск с другими продуктами программной виртуализации, Вы можете оставить данный параметр без изменений.

- ☒ VDI (VirtualBox Disk Image)
- ☐ VHD (Virtual Hard Disk)
- ☐ VMDK (Virtual Machine Disk)

Figure 2.5: рис 5. Тип жесткого диска

← Создать виртуальный жёсткий диск

Укажите формат хранения

Пожалуйста уточните, должен ли новый виртуальный жёсткий диск подстраивать свой размер под размер своего содержимого или быть точно заданного размера.

Файл **динамического** жёсткого диска будет занимать необходимое место на Вашем физическом носителе информации лишь по мере заполнения, однако не сможет уменьшиться в размере если место, занятое его содержимым, освободится.

Файл **фиксированного** жёсткого диска может потребовать больше времени при создании на некоторых файловых системах, однако, обычно, быстрее в использовании.

- ☒ Динамический виртуальный жёсткий диск
☐ Фиксированный виртуальный жёсткий диск


Figure 2.6: рис 6. Формат хранения

5. Зададим размер диска 40 ГБ и его расположение:

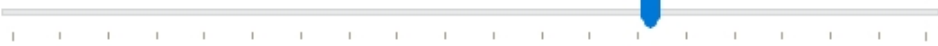
← Создать виртуальный жёсткий диск

Укажите имя и размер файла

Пожалуйста укажите имя нового виртуального жёсткого диска в поле снизу или используйте кнопку с иконкой папки справа от него.

D:\kpburdina\kpburdina.vdi 

Укажите размер виртуального жёсткого диска в мегабайтах. Эта величина ограничивает размер файловых данных, которые виртуальная машина сможет хранить на этом диске.

 40,00 ГБ

4,00 МБ 2,00 ТБ

Figure 2.7: рис 7. Имя и размер файла

В итоге у нас создается новая виртуальная машина:

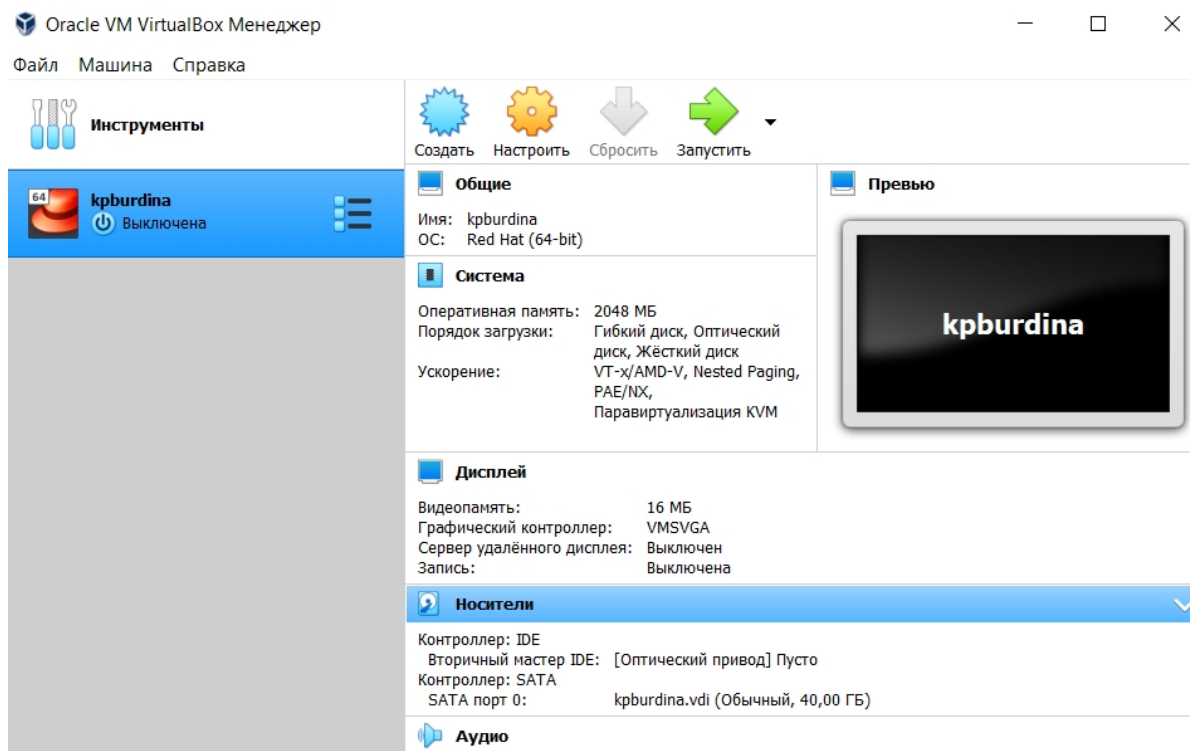


Figure 2.8: рис 8. Созданная виртуальная машина

6. Завершим ее настройку. Выберем в настройках носители и добавим новый привод оптических дисков, выбрав образ операционной системы Rocky:

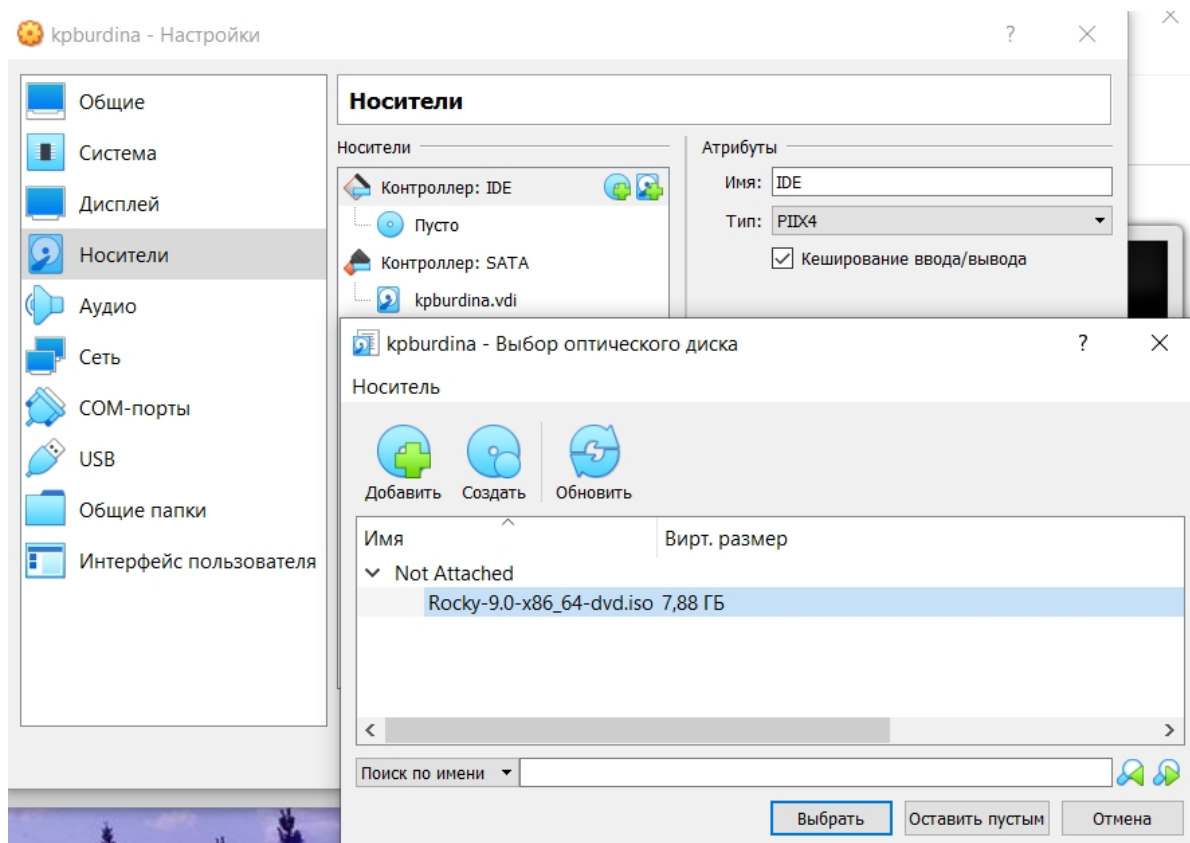


Figure 2.9: рис 9. Добавление образа ОС

7. Запустим созданную виртуальную машину:

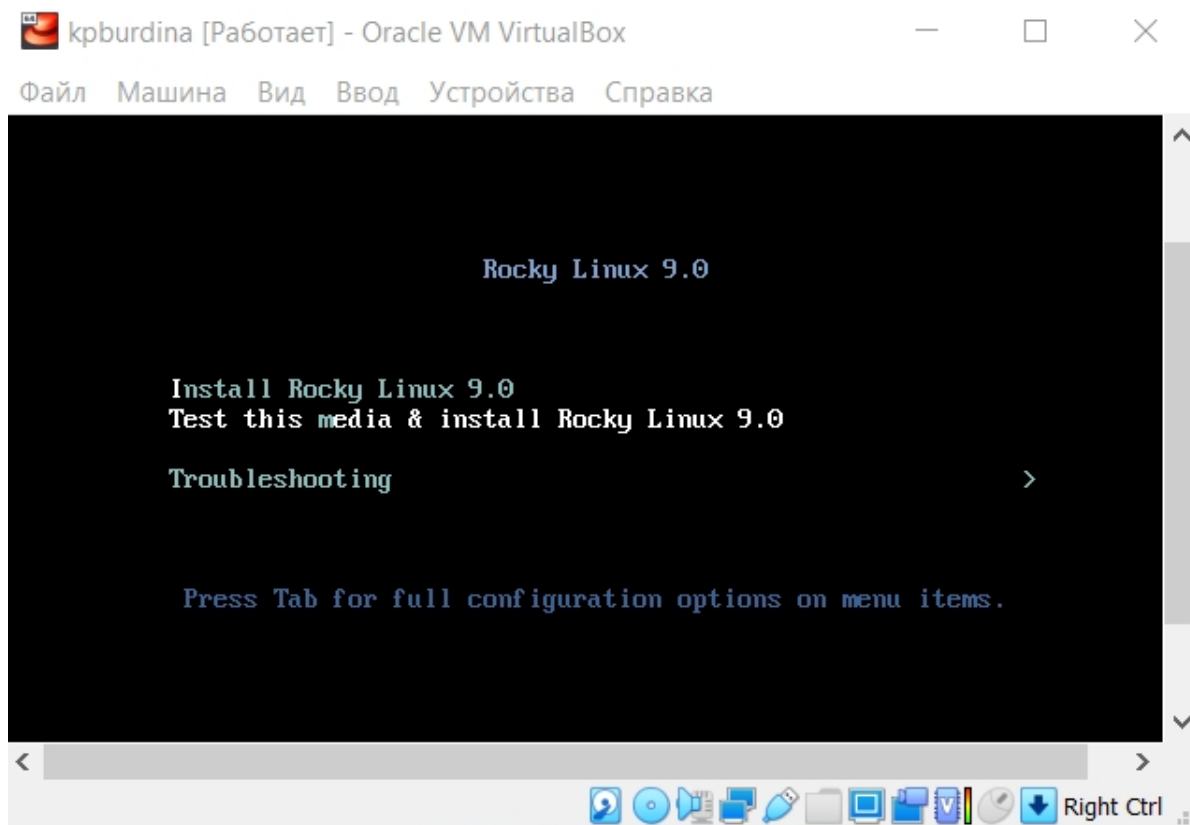


Figure 2.10: рис 10. Запуск машины

8. Выберем английский язык в качестве языка интерфейса:

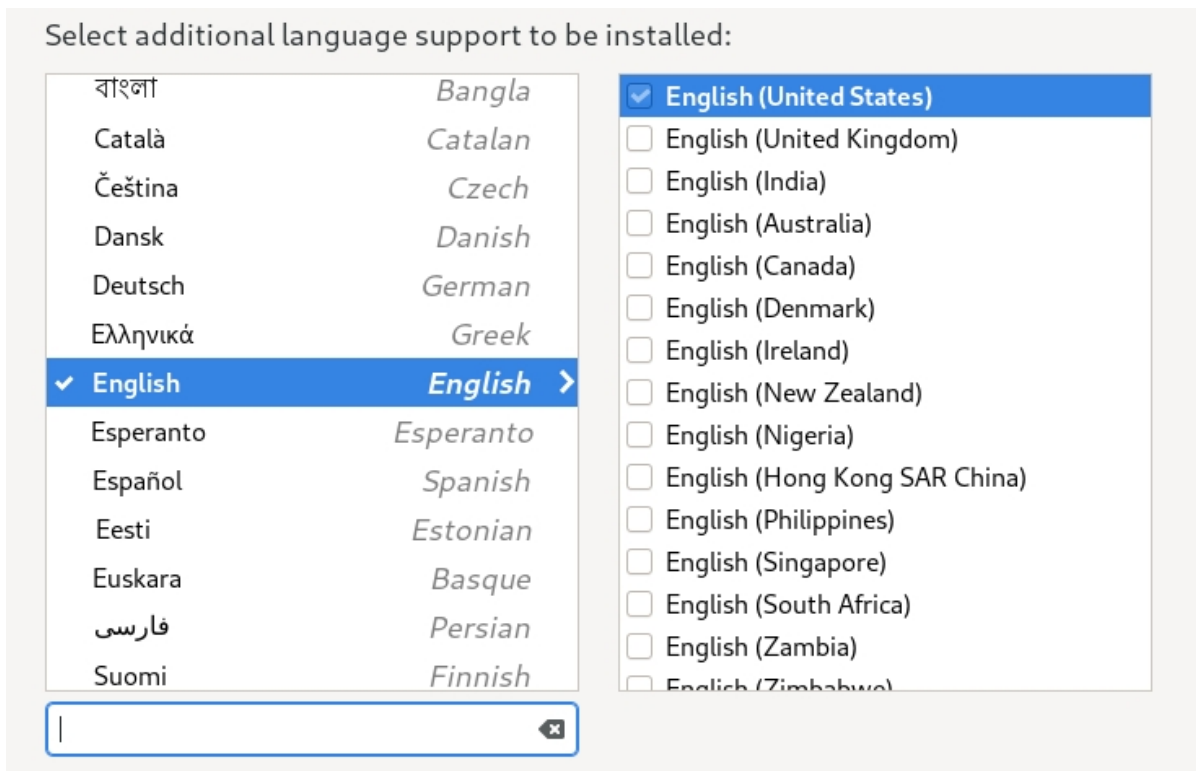


Figure 2.11: рис 11. Выбор языка интерфейса

9. Перейдем к настройкам установки операционной системы:

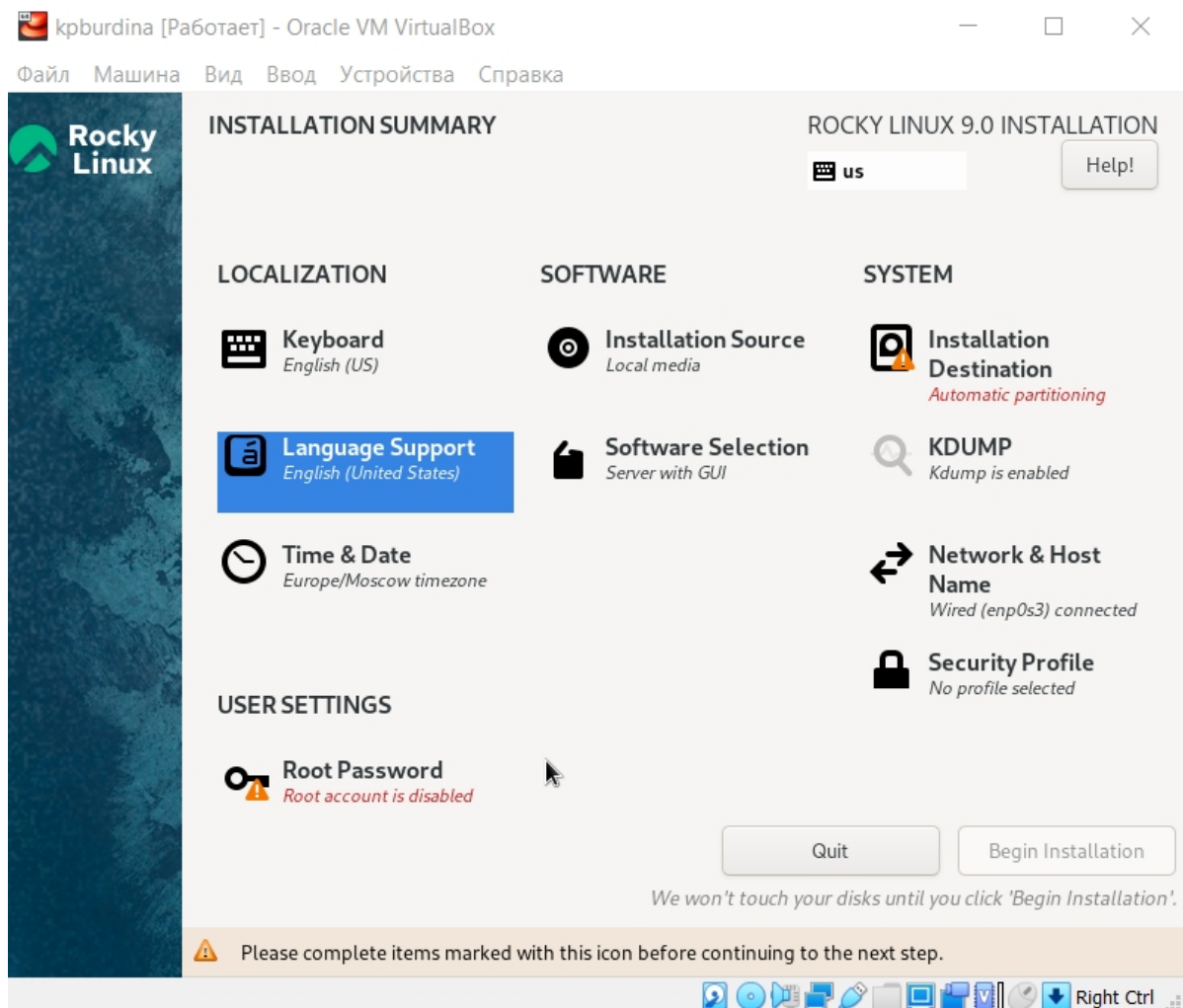


Figure 2.12: рис 12. Настройки ОС

10. В разделе выбора программ укажем в качестве базового окружения Server with GUI, а в качестве дополнения — Development Tools:

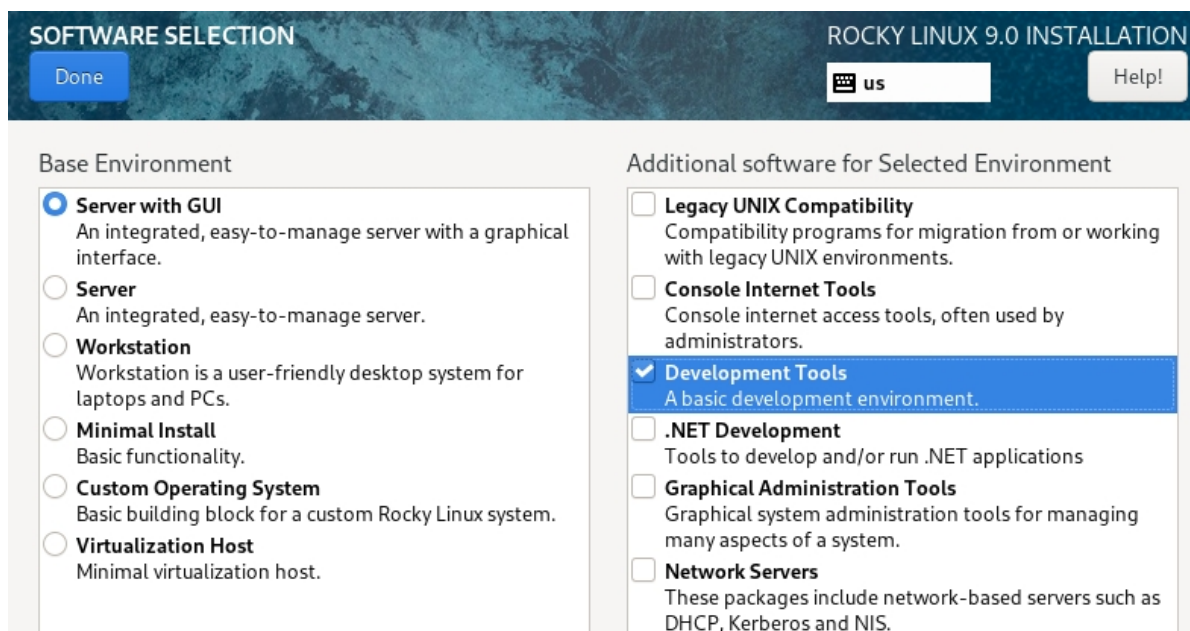


Figure 2.13: рис 13. Выбор программ

11. Отключим KDUMP:

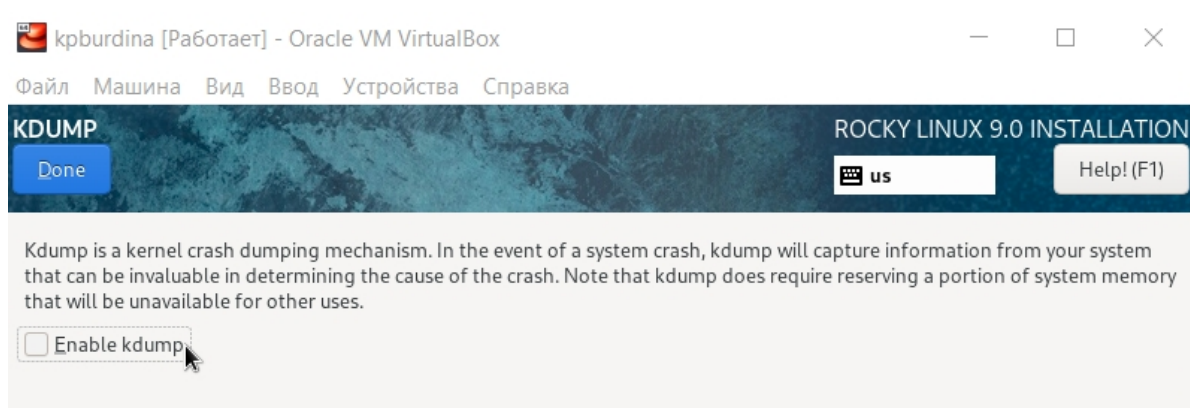


Figure 2.14: рис 14. Отключение KDUMP

12. Место установки операционной системы оставим без изменений по умолчанию:

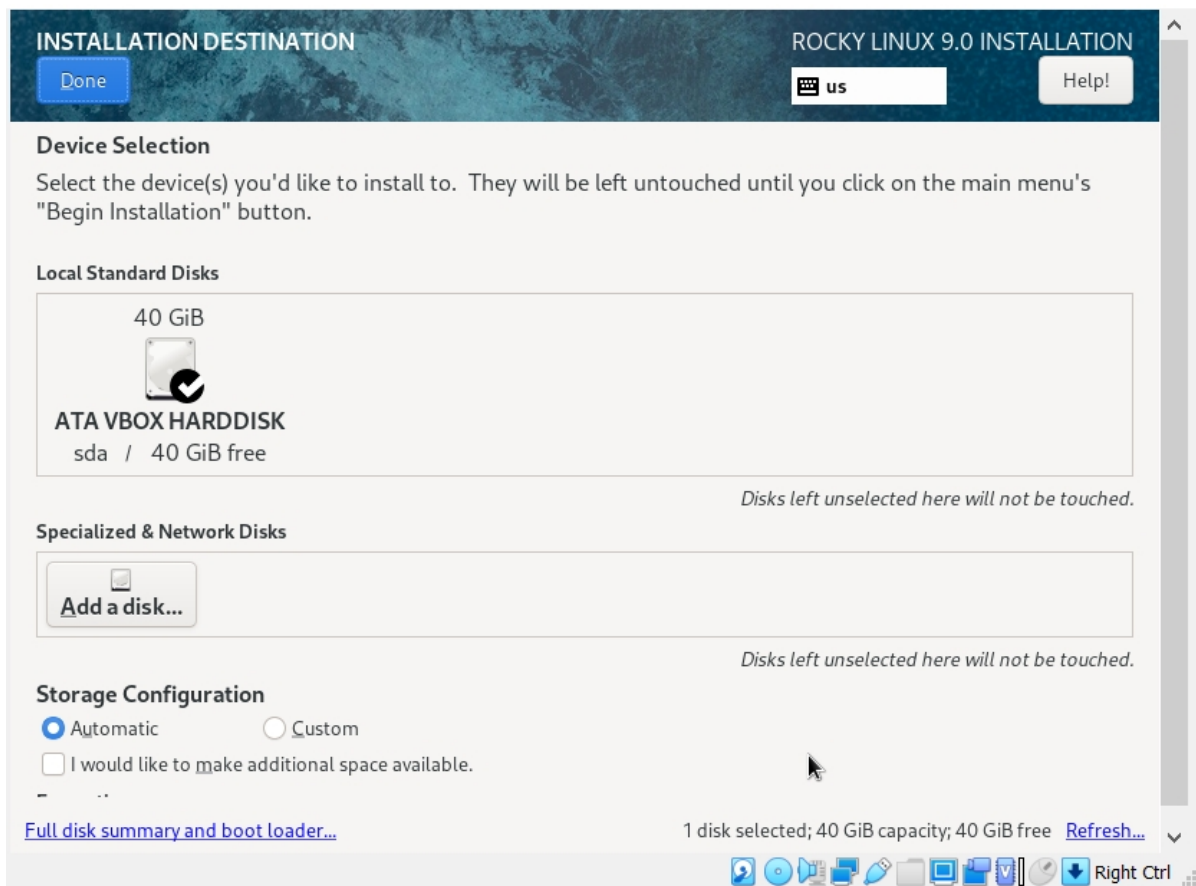


Figure 2.15: рис 15. Место установки ОС

13. Включим сетевое соединение и в качестве имени узла укажем kpburdina.localdomain:

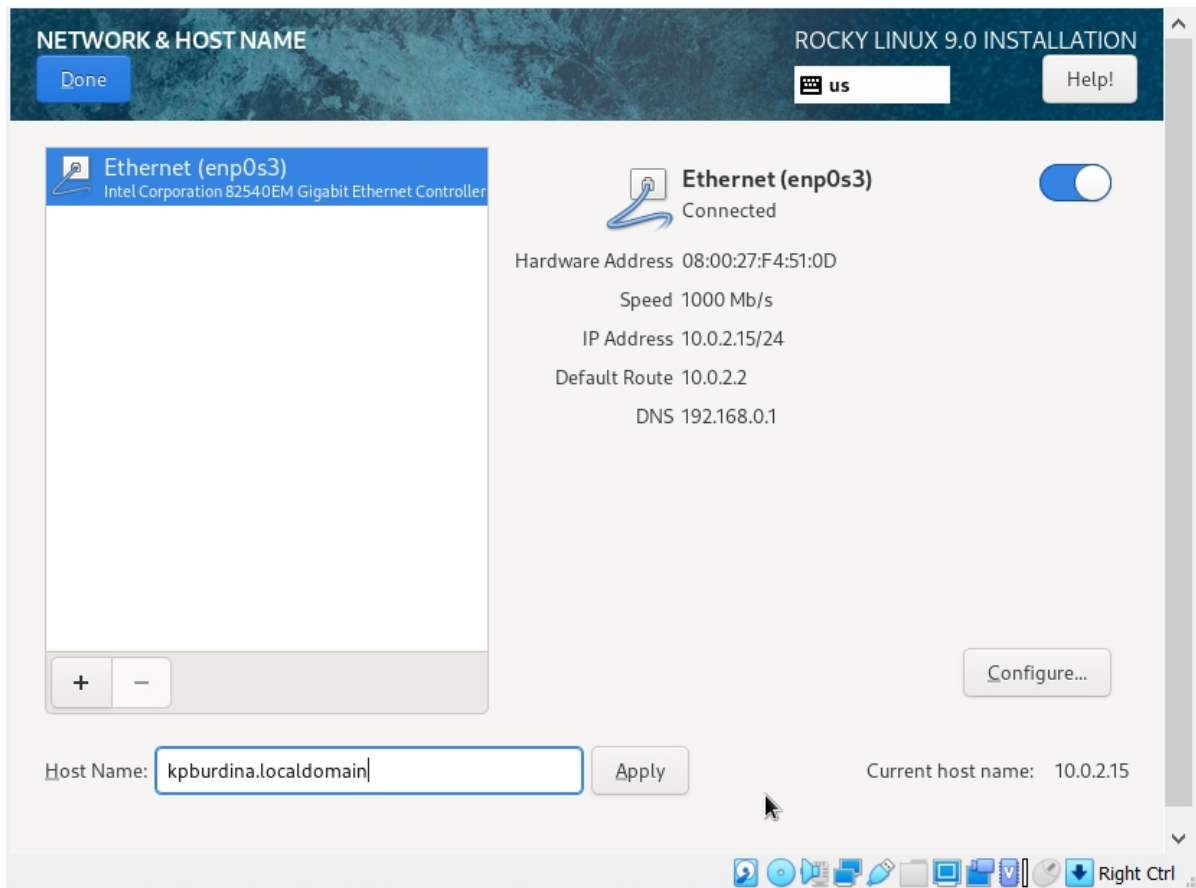


Figure 2.16: рис 16. Имя узла

14. Установим пароль для root, а затем для пользователя с правами администратора:

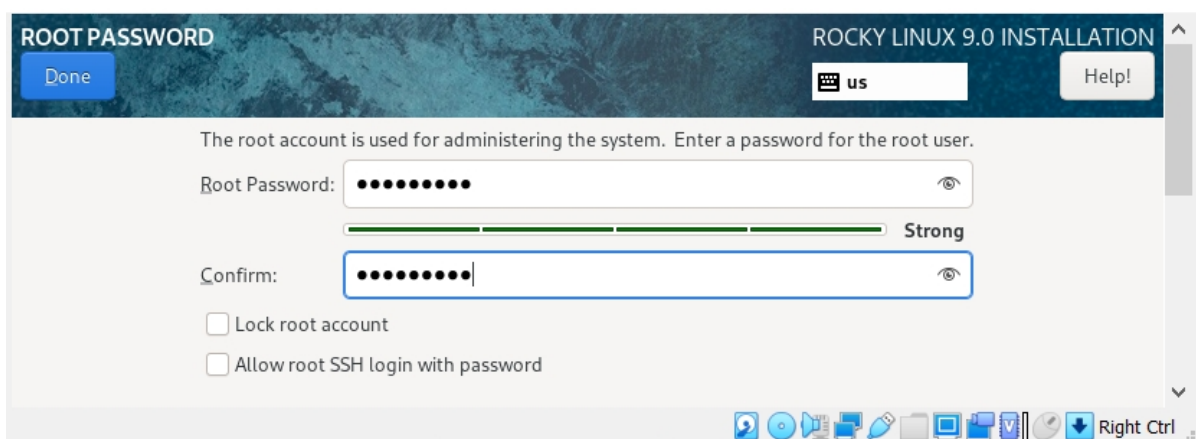
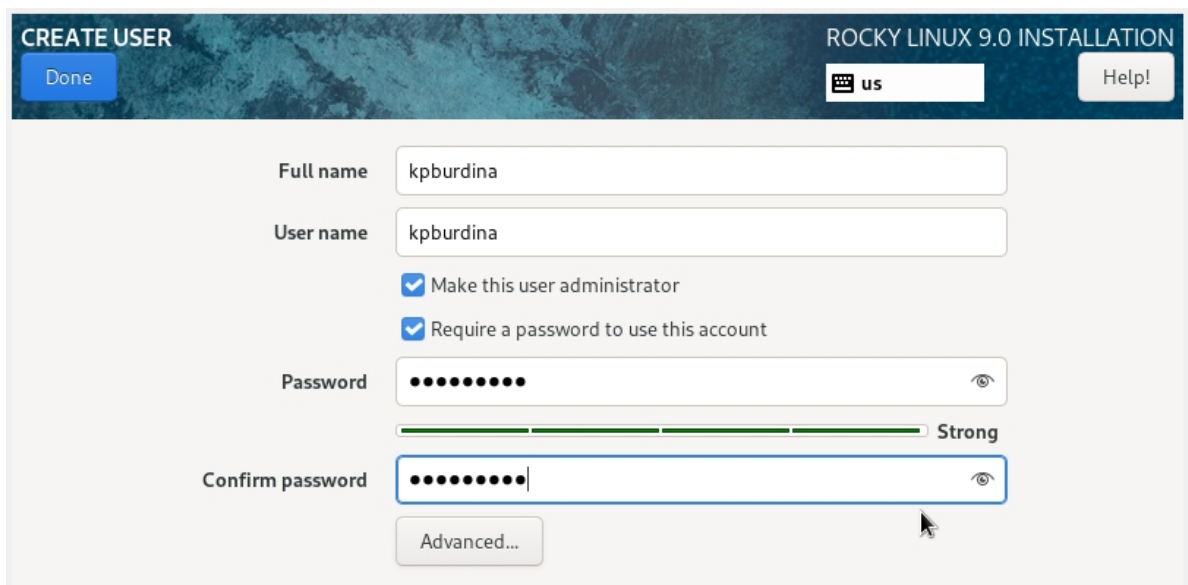


Figure 2.17: рис 17. Пароль для root



The image shows the 'CREATE USER' screen in the Rocky Linux 9.0 installation environment. The header bar is dark blue with 'CREATE USER' on the left, 'ROCKY LINUX 9.0 INSTALLATION' on the right, and a 'Done' button. Below the header, there are input fields for 'Full name' and 'User name', both containing 'kpburdina'. There are two checked checkboxes: 'Make this user administrator' and 'Require a password to use this account'. The 'Password' field is filled with dots, and a strength indicator below it shows a green bar and the word 'Strong'. The 'Confirm password' field is also filled with dots. A mouse cursor is pointing at the 'Advanced...' button at the bottom.

CREATE USER

ROCKY LINUX 9.0 INSTALLATION

Done

us

Help!

Full name kpburdina

User name kpburdina

☒ Make this user administrator

☒ Require a password to use this account

Password ●●●●●●●●

Strong

Confirm password ●●●●●●●●

Advanced...

Figure 2.18: рис 18. Пароль для администратора

15. Дождемся завершения установки операционной системы:

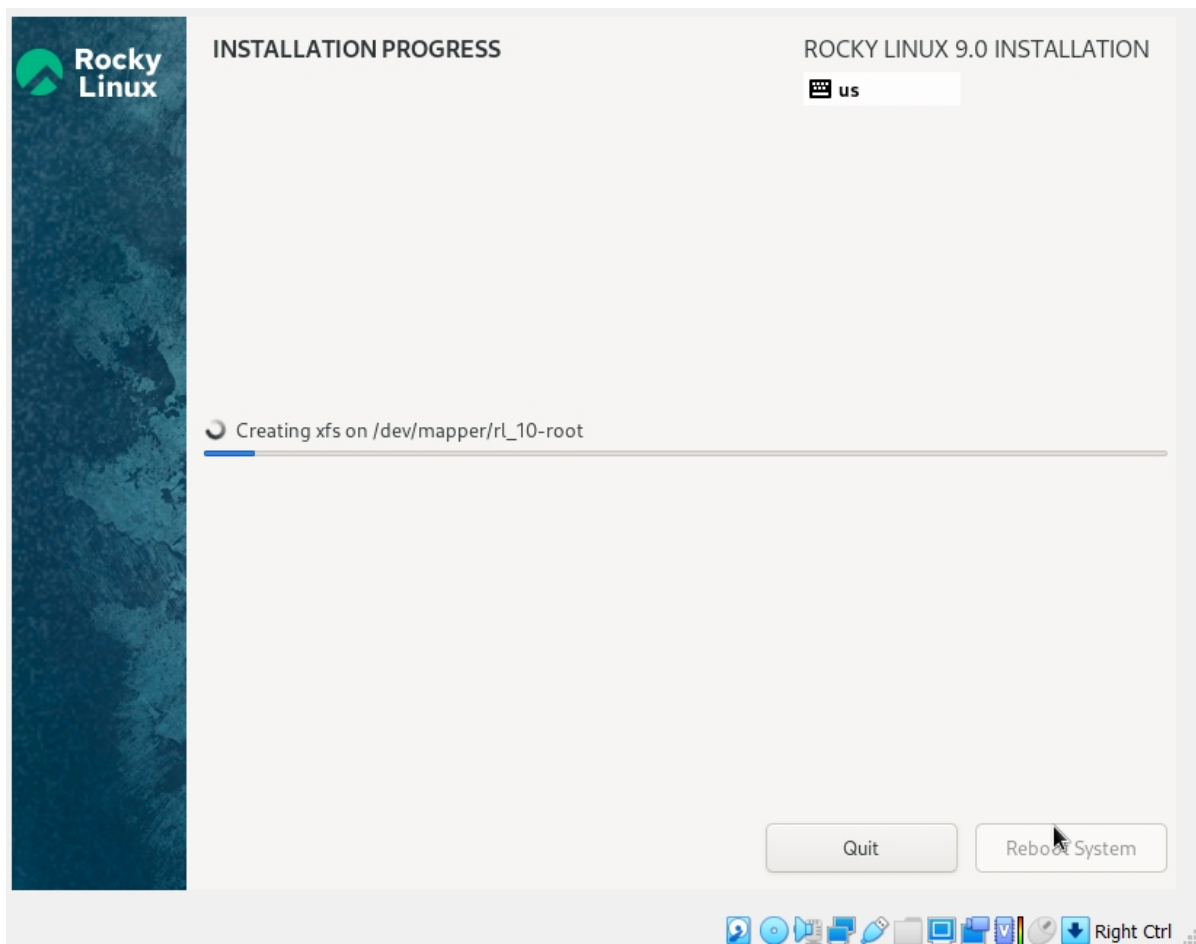


Figure 2.19: рис 19. Завершение установки ОС

16. К сожалению, установить Rocky корректно не получилось, поэтому образ был сменен на CentOS. Аналогично была создана виртуальная машина с тем же именем с данным образом, после чего завершена установка операционной системы:

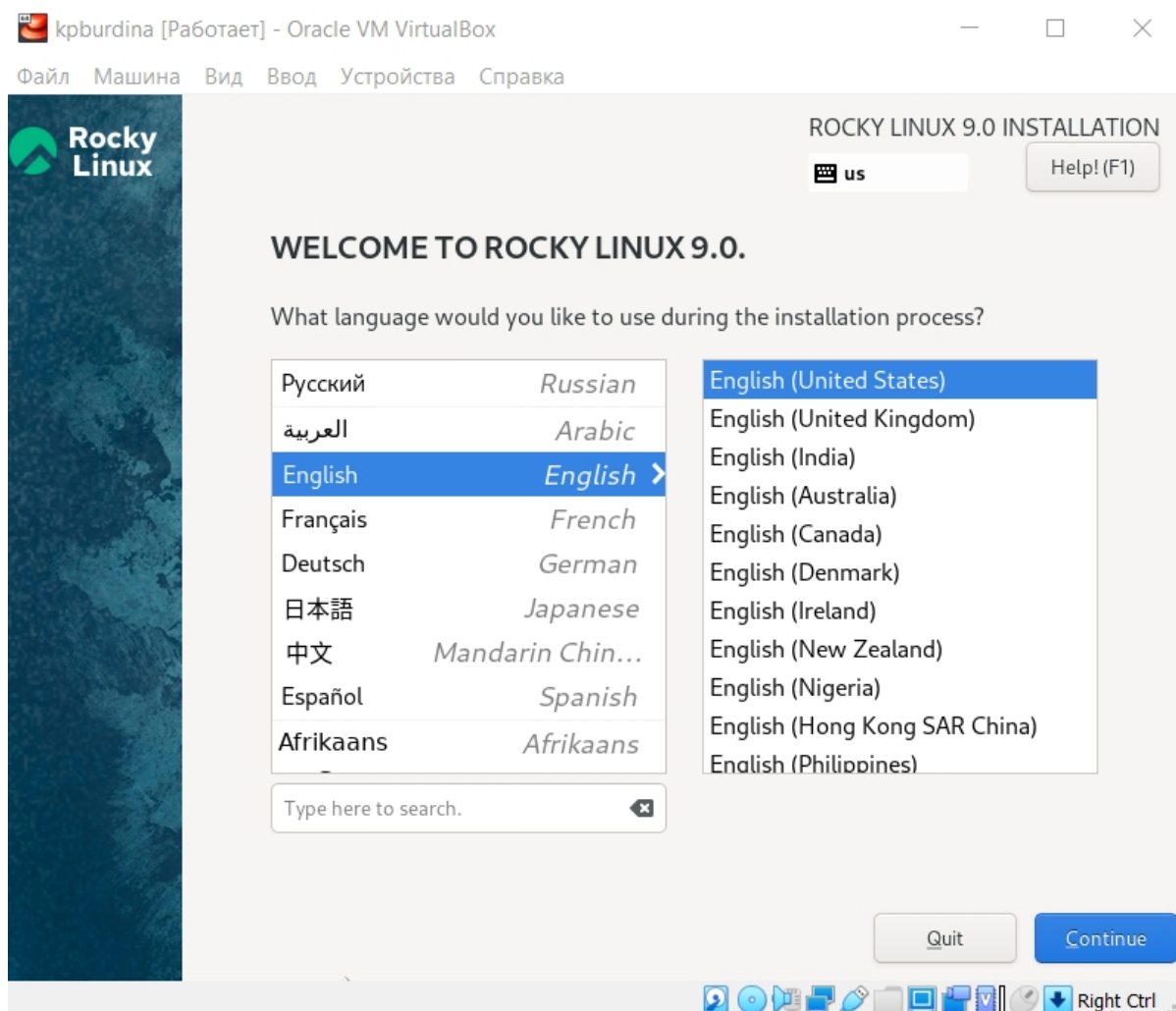


Figure 2.20: рис 20. Установка ОС

17. Так как виртуальная машина на устройстве работает с некоторыми неполадками (в том числе вылетает), было решено переместиться выполнять работу на удаленный рабочий стол. Мы создали виртуальную машину с прописанными ранее параметрами на ином устройстве, после чего продолжилди выполнение лабораторной работы.

Необходимо было подключить образ диска дополнений гостевой ОС, однако на устройстве, где выполнялась работа, имеется UTM, которая не требует дополнений. Поэтому перейдем к работе терминала.

18. После открытия терминала мы выполнили команду `dmesg | less` и посмотр-

рели ее вывод. С помощью данной команды можно найти информацию о данных системы. Для этого необходимо прописать в командной строке текст: `dmesg | grep -i "то, что ищем"`.

Для начала получим информацию о версии ядра Linux. Для этого введем команду `dmesg | grep -i "Linux version"`:

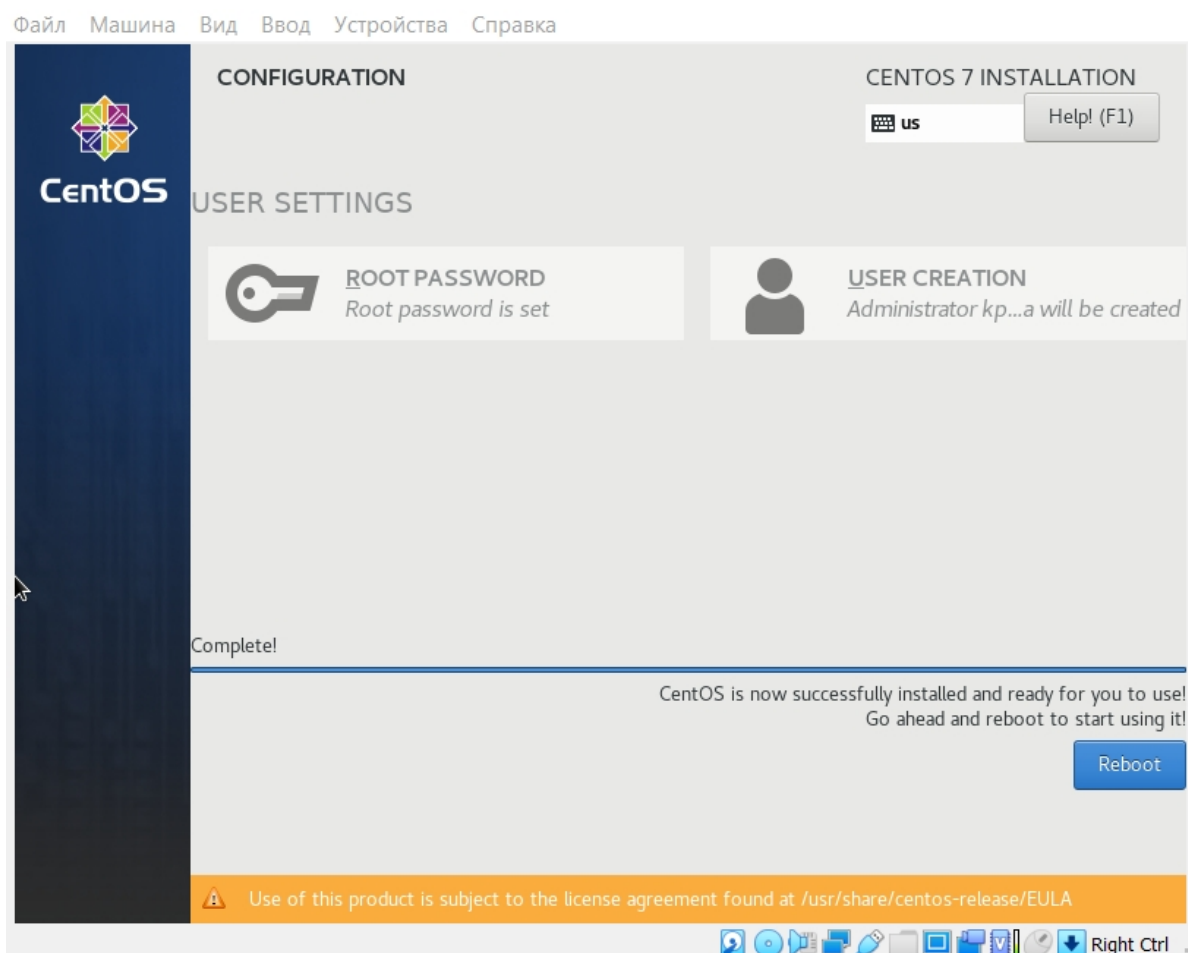


Figure 2.21: рис 21. Версия ядра Linux

Можем увидеть, что версия ядра - 5.14.

19. Далее посмотрим частоту процессора. Для этого введем в терминале команду `dmesg | grep -i "Mhz"`:

```
[kpburdina@kpburdina ~]$ dmesg | less
[kpburdina@kpburdina ~]$ dmesg | grep -i "linux version"
[    0.000000] Linux version 5.14.0-70.13.1.el9.aarch64 (mockbuild@pb-e78792d4-b
a0f-4fe7-aff1-59f42fdd3d2a-b-aarch64) (gcc (GCC) 11.2.1 20220127 (Red Hat 11.2.1
-9), GNU ld version 2.35.2-17.el9) #1 SMP Fri May 27 02:01:33 UTC 2022
[kpburdina@kpburdina ~]$
```

Figure 2.22: рис 22. Частота процессора

Можем увидеть, что частота процессора - 24Mhz.

20. Теперь необходимо узнать модель процессора (CPU0), однако на макбуке линукс не поддерживает данный процессор, поэтому он не отображается:

```
[kpburdina@kpburdina ~]$ dmesg | grep -i "detected mhz procesor"
[kpburdina@kpburdina ~]$ dmesg | grep -i "dmp"
[kpburdina@kpburdina ~]$ dmesg | grep -i "mhz"
[    0.000000] arch_timer: cp15 timer(s) running at 24.00MHz (virt).
[    0.000000] sched_clock: 56 bits at 24MHz, resolution 41ns, wraps every 43980
46511097ns
[kpburdina@kpburdina ~]$
```

Figure 2.23: рис 23. Модель процессора

21. Объем доступной оперативной памяти можем понять из следующего:

```
[kpburdina@kpburdina ~]$ dmesg | grep -i "cpu"
[    0.000000] Booting Linux on physical CPU 0x0000000000 [0x00000000]
[    0.000000] percpu: Embedded 30 pages/cpu s85784 r8192 d28904 u122880
[    0.000000] pcpu-alloc: s85784 r8192 d28904 u122880 alloc=30*4096
[    0.000000] pcpu-alloc: [0] 0 [0] 1 [0] 2 [0] 3
[    0.000000] Detected PIPT I-cache on CPU0
[    0.000000] CPU features: detected: Spectre-v4
[    0.000000] SLUB: HWalign=64, Order=0-3, MinObjects=0, CPUs=4, Nodes=1
[    0.000000] rcu: RCU restricting CPUs from NR_CPUS=4096 to nr_cpu_ids=4.
[    0.000000] rcu: Adjusting geometry for rcu_fanout_leaf=16, nr_cpu_ids=4
[    0.000000] rcu: Offload RCU callbacks from CPUs: (none).
[    0.000672] smp: Bringing up secondary CPUs ...
[    0.000848] Detected PIPT I-cache on CPU1
[    0.000903] CPU1: Booted secondary processor 0x0000000001 [0x00000000]
[    0.001157] Detected PIPT I-cache on CPU2
```

Figure 2.24: рис 24. Доступная оперативная память

Здесь видно, что нам доступно 3937260К памяти.

22. Тип обнаруженного гипервизора определить нельзя, поскольку на данном устройстве гипервизор отсутствует. Однако можно узнать тип файловой системы корневого раздела. С помощью команды `findmnt` посмотрим все файловые системы на устройстве:

```
[kpburdina@kpburdina ~]$ dmesg | grep -i "memory"
[ 0.000000] Early memory node ranges
[ 0.000000] Memory: 3937260K/4194304K available (11904K kernel code, 5486K rw
data, 9148K rodata, 4480K init, 11066K bss, 257044K reserved, 0K cma-reserved)
[ 0.492205] Freeing initrd memory: 51524K
[ 0.573410] Freeing unused kernel memory: 4480K
[kpburdina@kpburdina ~]$
```

Figure 2.25: рис 25. Файловые системы

Корневой раздел - это папка `home`. видим, что ее тип - `xfs`:

```
[kpburdina@kpburdina ~]$ findmnt
TARGET                                SOURCE                                FSTYPE  OPTIONS
/                                     /dev/mapper/rl-root                  xfs     rw,relatime,seclabel,attr2,inod
-/proc                               proc                                 proc     rw,nosuid,nodev,noexec,relatime
  -/proc/sys/fs/binfmt_misc          systemd-1                            autofs   rw,relatime,fd=31,pgrp=1,timeou
-/sys                                 sysfs                                sysfs     rw,nosuid,nodev,noexec,relatime
  -/sys/kernel/security              securityfs                            securit  rw,nosuid,nodev,noexec,relatime
  -/sys/fs/cgroup                    cgroup2                              cgroup2  rw,nosuid,nodev,noexec,relatime
  -/sys/fs/pstore                    pstore                              pstore   rw,nosuid,nodev,noexec,relatime
  -/sys/firmware/efi/efivars         efivarfs                             efivarf  rw,nosuid,nodev,noexec,relatime
  -/sys/fs/bpf                       none                                 bpf       rw,nosuid,nodev,noexec,relatime
  -/sys/fs/selinux                   selinuxfs                            selinux  rw,nosuid,noexec,relatime
  -/sys/kernel/debug                 debugfs                              debugfs   rw,nosuid,nodev,noexec,relatime
  -/sys/kernel/tracing               tracefs                              tracefs   rw,nosuid,nodev,noexec,relatime
  -/sys/fs/fuse/connections          fusectl                             fusectl   rw,nosuid,nodev,noexec,relatime
  -/sys/kernel/config               configfs                             configf   rw,nosuid,nodev,noexec,relatime
-/dev                                devtmpfs                             devtmpf   rw,nosuid,seclabel,size=1968628
  -/dev/shm                         tmpfs                                tmpfs     rw,nosuid,nodev,seclabel,inode6
  -/dev/pts                         devpts                               devpts    rw,nosuid,noexec,relatime,secla
```

Figure 2.26: рис 26. Тип ФС корневого каталога

23. Последовательность монтирования файловых систем также можем увидеть из выполнения данной команды. Они идут в очереди `boot`, `home`, `run`, `dev` и так далее:

```

|/home                               /dev/mapper/rl-home
|                                xfs      rw,relatime,seclabel,attr2,inod
|/boot                              /dev/vda2  xfs      rw,relatime,seclabel,attr2,inod

```

Figure 2.27: рис 27. Последовательность монтирования ФС

3 Выводы

В ходе работы мы приобрели практические навыки установки операционной системы на виртуальную машину и настроили минимально необходимые для дальнейшей работы сервисы. Также мы выполнили действия в командной строке и с помощью команды `dmesg` нашли необходимую нам информацию.

4 Список литературы

1. Методические материалы курса (<https://esystem.rudn.ru/mod/folder/view.php?id=892073>)

5 Ответы на контрольные вопросы

1. Учётная запись содержит сведения, необходимые для опознания пользователя при подключении к системе, сведения для авторизации и учёта. Это идентификатор пользователя (login) и его пароль. Пароль или его аналог, как правило, хранится в зашифрованном или хэшированном виде для обеспечения его безопасности.

2. Команды терминала с примерами.

- для получения справки по команде:

`man bash` - руководство по терминалу

- для перемещения по файловой системе:

`cd home` - перейти в домашнюю папку

- для просмотра содержимого каталога:

`ls -l` - посмотреть файлы каталога с поясняющей информацией

- для определения объёма каталога:

`df -h` - представляет данные о размере в удобном для восприятия формате

- для создания / удаления каталогов / файлов:

`touch lab` - создать каталог для лабораторных работ

`rm lab1.pdf` - удалить файл с отчетом по лабораторной работе

- для задания определённых прав на файл / каталог:

`chmod rw-` - разрешение на чтение и изменение

- для просмотра истории команд:

`history` | `tail` - показ последних 10 выполненных команд

3. Файловая система - это часть операционной системы, которая обеспечивает выполнение операций над файлами. Она позволяет создавать, переименовывать, удалять, переносить и копировать файлы с одного носителя на другой; искать файлы, хранящиеся на разных носителях, запускать программы на выполнение.

Файловые системы Windows: FAT, NTFS, ReFS.

Файловые системы macOS: HFS+, Apple Xsan.

Файловые системы Linux: Ext4, JFS, XFS, Btrfs, ReiserFS.

4. Отображение информации о смонтированной файловой системе в ОС Linux:

`findmnt -mtab`

5. Команда `ps` предназначена для вывода списка активных процессов в системе и информации о них. Команда `grep` запускается одновременно с `ps` и будет выполнять поиск по результатам команды `ps`. Вывести список всех процессов можно, выполнив в командной строке: `ps aux`.

Когда известен PID процесса, мы можем убить его командой `kill`. Команда `kill` принимает в качестве параметра PID процесса. Например, убьем процесс с номером 25609: `kill 25609`.

Команда `killall` в Linux предназначена для «убийства» всех процессов, имеющих одно и то же имя. Это удобно, так как нам не нужно знать PID процесса. Например, мы хотим закрыть все процессы с именем `gcalctool`. Выполните в терминале: `killall gcalctool`.