

Защита лабораторной работы №1. Установка и конфигурация операционной системы на виртуальную машину

Бурдина Ксения Павловна

2022 Sep 10th

RUDN University, Moscow, Russian Federation

Результат выполнения лабораторной работы №1

Цель выполнения лабораторной
работы

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

Результат выполнения лабораторной работы

Создание новой виртуальной машины:

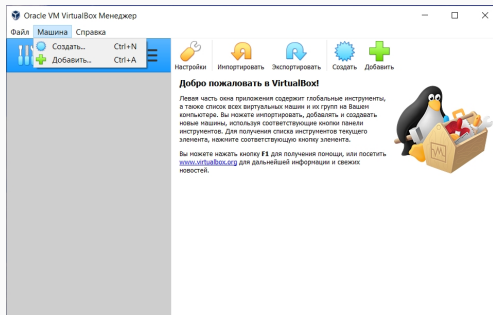


Figure 1: рис 1. Создание новой машины

Результат выполнения лабораторной работы


Настройка виртуальной машины:


← 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: рис 2. Задание имени и типа ОС

Результат выполнения лабораторной работы

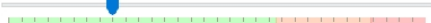
Настройка виртуальной машины:

← 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

The image shows a screenshot of a virtual machine creation wizard. It features a slider for memory allocation ranging from 4 MB to 8192 MB. A blue handle is positioned at 2048 MB, which is also displayed in a text box on the right. The slider bar has a green segment from 4 MB to 1024 MB and an orange segment from 1024 MB to 8192 MB. The recommended value of 1024 MB is indicated by text above the slider.

Figure 3: рис 3. Размер памяти ВМ

Результат выполнения лабораторной работы

Настройка виртуальной машины:

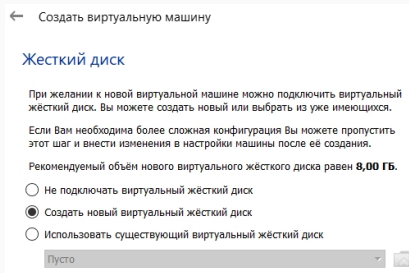


Figure 4: рис 4. Конфигурация жесткого диска

Результат выполнения лабораторной работы

Настройка виртуальной машины:

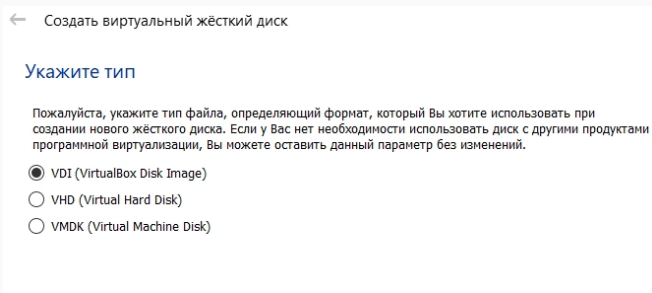


Figure 5: рис 5. Тип жесткого диска

Результат выполнения лабораторной работы

Настройка виртуальной машины:

← 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 6: рис 6. Формат хранения


Результат выполнения лабораторной работы

Настройка виртуальной машины:


← 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.

 4,00 МБ 2,00 ТБ

40,00 ГБ

Figure 7: рис 7. Имя и размер файла

Результат выполнения лабораторной работы

Добавление образа:

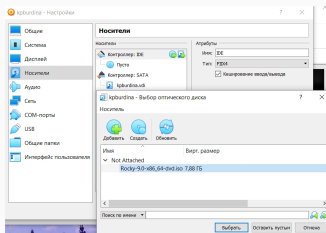


Figure 8: рис 8. Добавление образа ОС

Результат выполнения лабораторной работы

Настройка операционной системы:

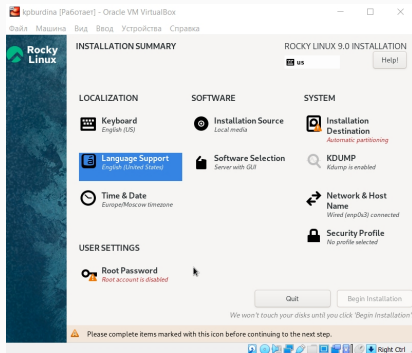


Figure 9: рис 9. Настройки ОС

Результат выполнения лабораторной работы

Завершение установки операционной системы:

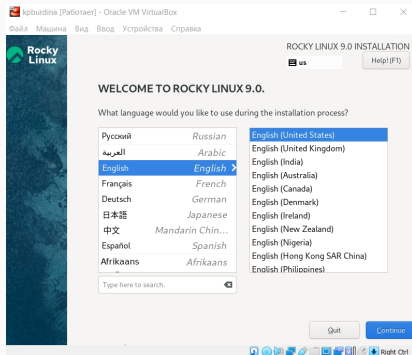


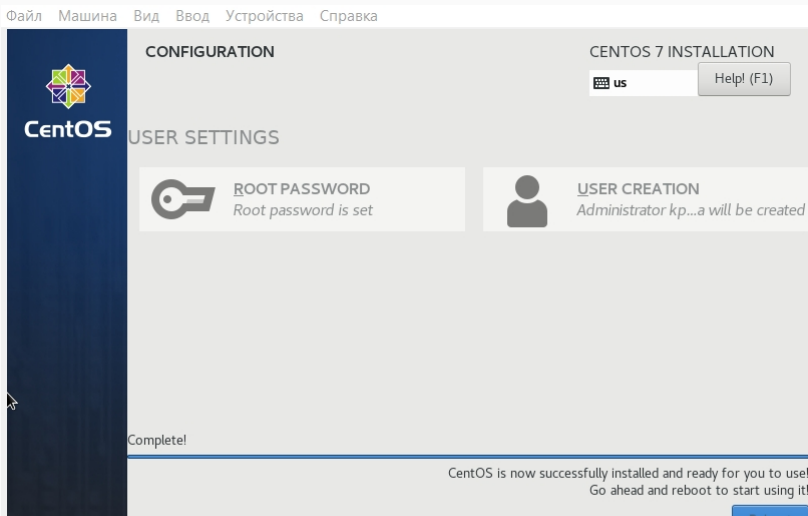
Figure 10: рис 10. Установка ОС

Результат выполнения лабораторной работы

Результат выполнения лабораторной работы

Работа с командой dmesg в консоли.

Версия ядра Linux:



Результат выполнения лабораторной работы

Работа с командой dmesg в консоли.

Частота процессора:

```
[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 12: рис 12. Частота процессора

Результат выполнения лабораторной работы

Работа с командой dmesg в консоли.

Модель процессора:

```
[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 13: рис 13. Модель процессора

Результат выполнения лабораторной работы

Результат выполнения лабораторной работы

Работа с командой dmesg в консоли.

Объем доступной оперативной памяти:

```
[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 0x00000000001 [0x00000000]
[ 0.001157] Detected PIPT I-cache on CPU2
```

Figure 14: рис 14. Доступная оперативная память

Результат выполнения лабораторной работы

Результат выполнения лабораторной работы

Работа с командой dmesg в консоли.

Тип файловой системы корневого раздела:

```
[kpburdina@kpburdina ~]$ findmnt
TARGET          SOURCE          FSTYPE  OPTIONS
/               /dev/mapper/rl-root
|
├─/proc          proc            xfs      rw,relatime,seclabel,attr2,inod
├─┬─/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     bpf             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         devtmpfs  rw,nosuid,seclabel,size=1968628
├─┬─/dev/shm        tmpfs            tmpfs     rw,nosuid,nodev,seclabel,inode6
├─┬─/dev/pts         devpts           devpts    rw,nosuid,noexec,relatime,secla
```

Figure 15: рис 15. Тип ФС корневого каталога

Результат выполнения лабораторной работы

Работа с командой dmesg в консоли.

Последовательность монтирования файловых систем:

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

Figure 16: рис 16. Последовательность монтирования ФС

Выводы

1. Приобрели практические навыки установки операционной системы на виртуальную машину.
2. Настроили минимально необходимые для дальнейшей работы сервисы.
3. Выполнили действия в командной строке и с помощью команды `dmesg` нашли необходимую нам информацию.