

Лабораторная работа №1. Установка и конфигурация операционной системы на виртуальную машину

Выполнила: Лебедева Ольга Андреевна

Преподаватель Кулябов Дмитрий Сергеевич д.ф.-м.н., профессор кафедры прикладной информатики и кибербезопасности

2024

Российский университет дружбы народов, Москва, Россия

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

Программа VirtualBox предоставляет широкий спектр возможностей для работы с виртуальными машинами. Это решение подходит для тестирования новых операционных систем, запуска старых приложений или изоляции потенциально опасного программного обеспечения. Благодаря интуитивно понятному интерфейсу и богатому функционалу, VirtualBox стал выбором многих пользователей по всему миру[1].

Запускаем виртуальную машину, нажимаем кнопку “создать” и выбираем скаченный образ ISO: См. рис. 1

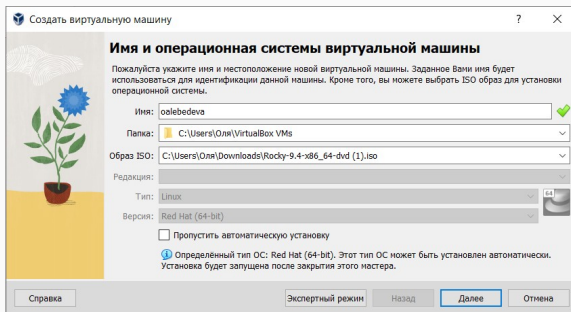


Рис. 1: Создание виртуальной машины

Задаём настройки гостевой ОС: См. рис. 2

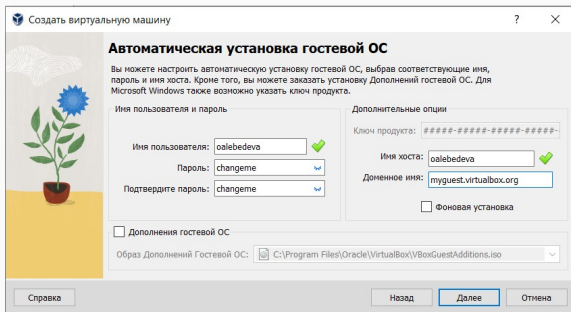


Рис. 2: Настройки гостевой ОС

Настраиваем оборудование VM, изменяя размер ОЗУ: См. рис. 3

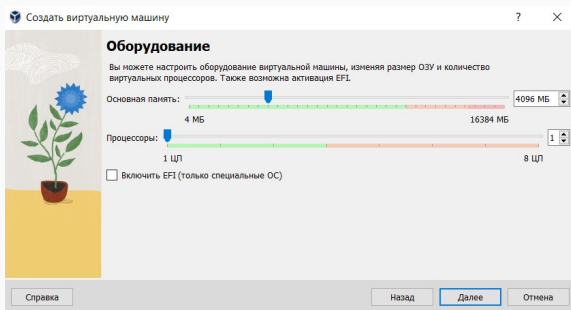


Рис. 3: Оборудование VB

Создаём новый виртуальный жёсткий диск размером 40 Гб: См. рис. 4

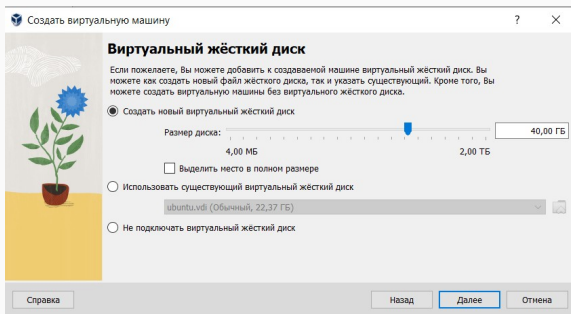


Рис. 4: Размер памяти

Проверяю итоговую конфигурацию для виртуальной машины: См. рис. 5

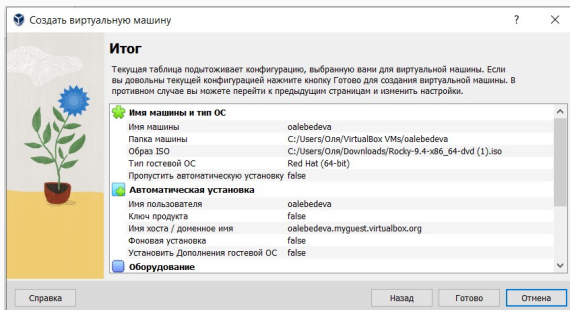


Рис. 5: Итоговые настройки

Меняем контроллер на скаченный образ Rocky: См. рис. 6

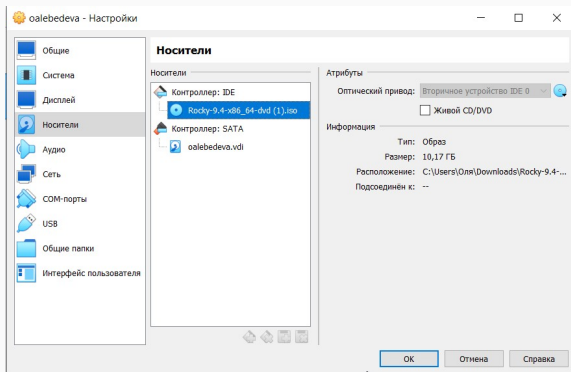


Рис. 6: Носители

Попадаем в стартовое меню установки, выбираем английский язык: См. рис. 7

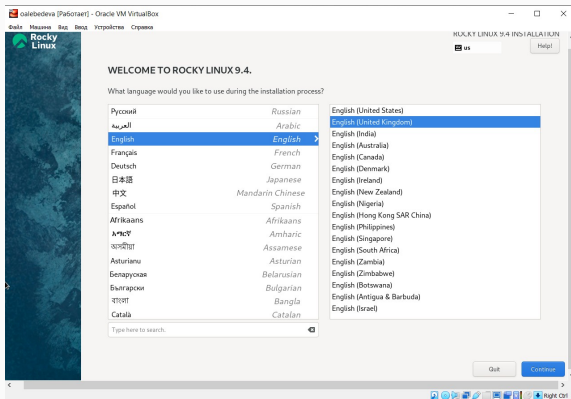


Рис. 7: Стартовое меню установки

В Installation Destination выбираем диск: См. рис. 8

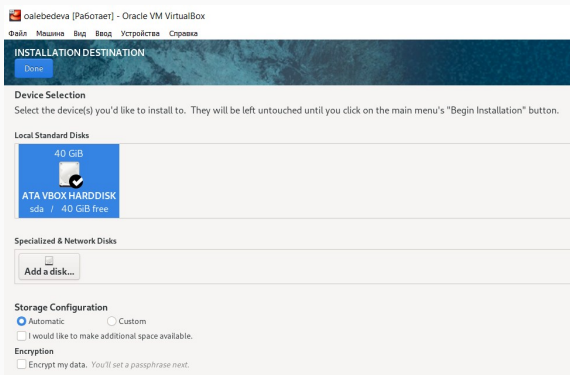


Рис. 8: Выбор диска

В Softwar Selection выбираем Server with GUI. В дополнительном ПО отмечаем Development Tools: См. рис. 9

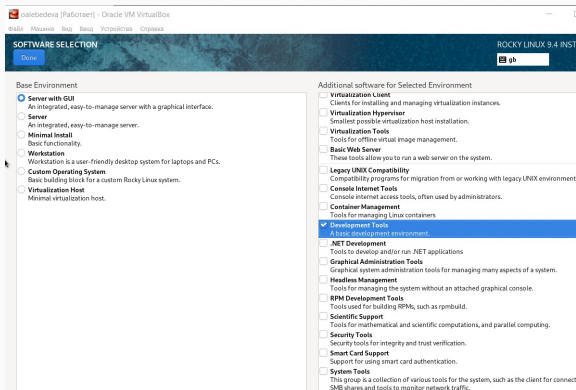


Рис. 9: Server with GUI

Заходим в KDUMP и отключаем его: См. рис. 10

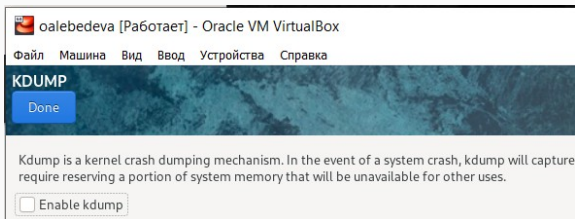


Рис. 10: Отключение KDUMP

Выполнение лабораторной работы

Заходим в Network&Host Name и прописываем host name: См. рис. 11

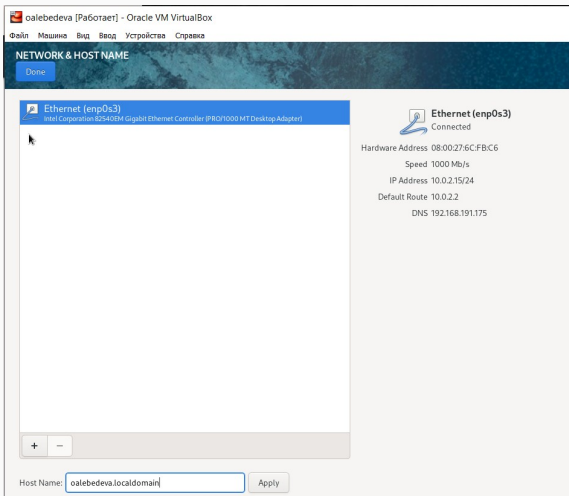




Рис. 11: Имя хоста

В разделе Root Password задаём пароль: См. рис. 12

The root account is used for administering the system. Enter a password for the root user.

Root Password: 

Strong

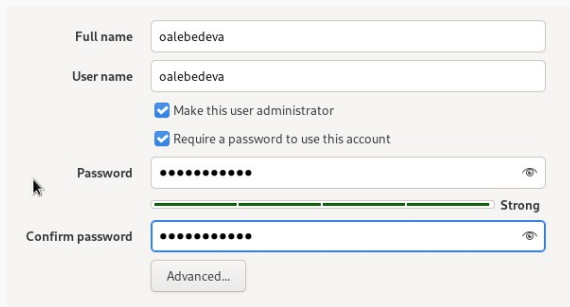
Confirm: 

☐ Lock root account

☐ Allow root SSH login with password

Рис. 12: Root password

Завершаем настройки во вкладке Create User: См. рис. 13



The screenshot shows a 'Create User' form with the following fields and options:

- Full name:** oalebedeva
- User name:** oalebedeva
- ☒ Make this user administrator
- ☒ Require a password to use this account
- Password:** A text field with 10 dots, a toggle eye icon, and a strength indicator bar below it. The bar is green and labeled 'Strong'.
- Confirm password:** A text field with 10 dots and a toggle eye icon.
- Advanced...** A button at the bottom.

Рис. 13: Create User

Запускаем установку и ждем перезагрузки системы: См. рис. 14

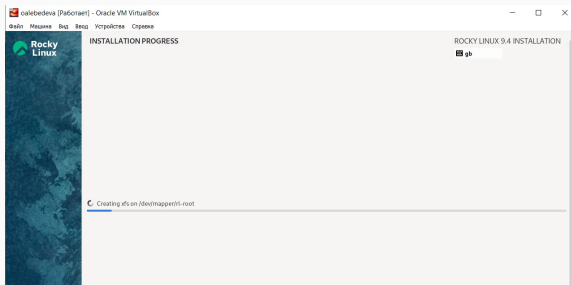


Рис. 14: Завершение установки

Заходим в созданный аккаунт: См. рис. 15

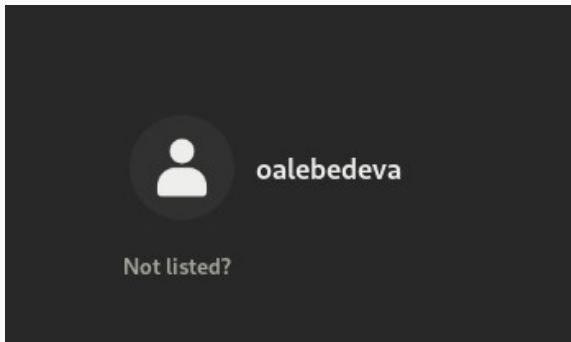


Рис. 15: Вход в аккаунт

Запускаем образ диска дополнений гостевой ОС: См. рис. 16

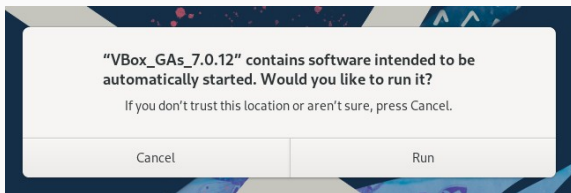
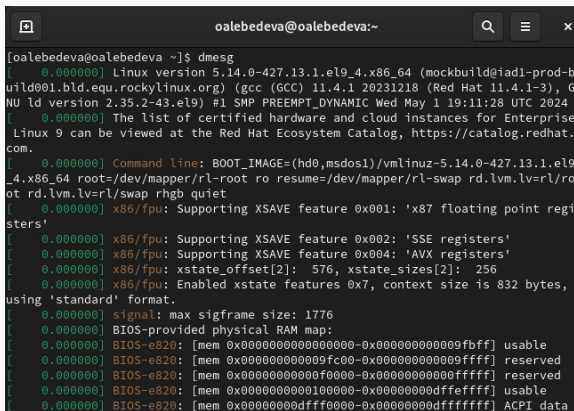


Рис. 16: Подключение гостевых настроек

Посмотрим последовательность загрузки системы, выполнив команду `dmesg`: См. рис. 17

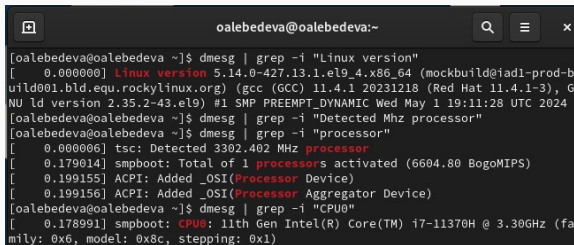


```
oalebedeva@oalebedeva:~$ dmesg
[ 0.000000] Linux version 5.14.0-427.13.1.el9_4.x86_64 (mockbuild@iad1-prod-b
uild001.bld.equ.rockylinux.org) (gcc (GCC) 11.4.1 20231218 (Red Hat 11.4.1-3), G
NU ld version 2.35.2-43.el9) #1 SMP PREEMPT_DYNAMIC Wed May 1 19:11:28 UTC 2024
[ 0.000000] The list of certified hardware and cloud instances for Enterprise
Linux 9 can be viewed at the Red Hat Ecosystem Catalog, https://catalog.redhat.com.
[ 0.000000] Command line: BOOT_IMAGE=(hd0,msdos1)/vmlinuz-5.14.0-427.13.1.el9
_4.x86_64 root=/dev/mapper/rl-root ro resume=/dev/mapper/rl-swap rd.lvm.lv=rl/ro
ot rd.lvm.lv=rl/swap rhgb quiet
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point regi
sters'
[ 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] signal: max sigframe size: 1776
[ 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 0x00000000001000000-0x0000000000dffff] usable
[ 0.000000] BIOS-e820: [mem 0x00000000dffff0000-0x00000000dfffffff] ACPI data
```

Рис. 17: Последовательность загрузки системы

Получим следующую информацию: См. рис. 18, См. рис. 19, См. рис. 20, См. рис. 21

1. Версия ядра Linux (Linux version).
2. Частота процессора (Detected Mhz processor).
3. Модель процессора (CPU0).
4. Объем доступной оперативной памяти (Memory available).
5. Тип обнаруженного гипервизора (Hypervisor detected).
6. Тип файловой системы корневого раздела.
7. Последовательность монтирования файловых систем



```
oalebedeva@oalebedeva:~  
[oalebedeva@oalebedeva ~]$ dmesg | grep -i "Linux version"  
[ 0.000000] Linux version 5.14.0-427.13.1.el9_4.x86_64 (mockbuild@iad1-prod-b  
uild001.bld.equ.rockylinux.org) (gcc (GCC) 11.4.1 20231218 (Red Hat 11.4.1-3), G  
NU ld version 2.35.2-43.el9) #1 SMP PREEMPT_DYNAMIC Wed May 1 19:11:28 UTC 2024  
[oalebedeva@oalebedeva ~]$ dmesg | grep -i "Detected Mhz processor"  
[oalebedeva@oalebedeva ~]$ dmesg | grep -i "processor"  
[ 0.000006] tsc: Detected 3302.402 MHz processor  
[ 0.179014] smpboot: Total of 1 processors activated (6604.80 BogoMIPS)  
[ 0.199155] ACPI: Added _OSI(Processor Device)  
[ 0.199156] ACPI: Added _OSI(Processor Aggregator Device)  
[oalebedeva@oalebedeva ~]$ dmesg | grep -i "CPU0"  
[ 0.178991] smpboot: CPU0: 11th Gen Intel(R) Core(TM) i7-11370H @ 3.30GHz (fa  
mily: 0x6, model: 0x8c, stepping: 0x1)
```

Рис. 18: Версия ядра Linux, частота процессора, модель процессора

```
oalebedeva@oalebedeva ~]$ dmesg | grep -i "memory"
0.001012] ACPI: Reserving FACP table memory at [mem 0xdfff00f0-0xdfff01e3]
0.001013] ACPI: Reserving DSDT table memory at [mem 0xdfff0610-0xdfff2962]
0.001013] ACPI: Reserving FACS table memory at [mem 0xdfff0200-0xdfff023f]
0.001014] ACPI: Reserving FACS table memory at [mem 0xdfff0200-0xdfff023f]
0.001014] ACPI: Reserving APIC table memory at [mem 0xdfff0240-0xdfff0293]
0.001015] ACPI: Reserving SSDT table memory at [mem 0xdfff02a0-0xdfff060b]
0.001239] Early memory node ranges
0.011695] PM: hibernation: Registered nosave memory: [mem 0x00000000-0x0000
ffff]
```

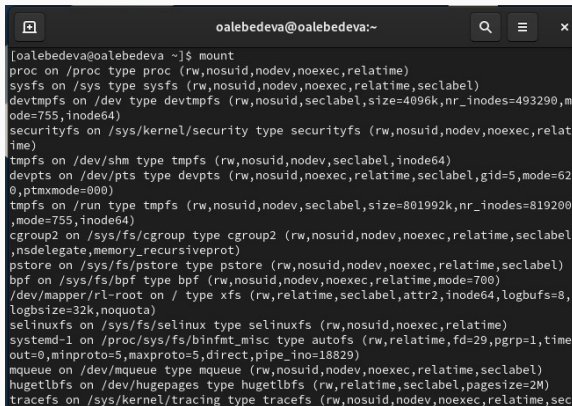
Рис. 19: Объем доступной оперативной памяти

```
[oalebedeva@oalebedeva ~]$ dmesg | grep -i "hypervisor"
[ 0.000000] Hypervisor detected: KVM
[ 0.065535] GDS: Unknown: Dependent on hypervisor status
[ 1.691648] vmwgfx 0000:00:02.0: [drm] *ERROR* vmwgfx seems to be running on
an unsupported hypervisor.
[oalebedeva@oalebedeva ~]$ df -T
```

Filesystem	Type	1K-blocks	Used	Available	Use%	Mounted on
devtmpfs	devtmpfs	4096	0	4096	0%	/dev
tmpfs	tmpfs	2004976	0	2004976	0%	/dev/shm
tmpfs	tmpfs	801992	9344	792648	2%	/run
/dev/mapper/rl-root	xfs	36683776	6261720	30422056	18%	/
/dev/sda1	xfs	983040	277640	705400	29%	/boot
tmpfs	tmpfs	400992	120	400872	1%	/run/user/1000
/dev/sr0	iso9660	52196	52196	0	100%	/run/media/oalebedeva/VBox_GAs_7.0.12

Рис. 20: Тип обнаруженного гипервизора, тип файловой системы корневого раздела

Выполнение лабораторной работы



```
oalebedeva@oalebedeva:~  
[oalebedeva@oalebedeva ~]$ mount  
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)  
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime,seclabel)  
devtmpfs on /dev type devtmpfs (rw,nosuid,seclabel,size=4096k,nr_inodes=493290,mode=755,inode64)  
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)  
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,seclabel,inode64)  
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,seclabel,gid=5,mode=620,ptmxmode=000)  
tmpfs on /run type tmpfs (rw,nosuid,nodev,seclabel,size=801992k,nr_inodes=819200,mode=755,inode64)  
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,seclabel,nsdelegate,memory_recursiveprot)  
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime,seclabel)  
bpf on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)  
/dev/mapper/rl-root on / type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquota)  
selinuxfs on /sys/fs/selinux type selinuxfs (rw,nosuid,noexec,relatime)  
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=29,prgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=18829)  
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime,seclabel)  
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,seclabel,pagesize=2M)  
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime,sec
```

Рис. 21: Последовательность монтирования файловых систем

Приобрели практические навыки установки операционной системы на виртуальную машину, настроили минимально необходимые для дальнейшей работы сервисы.

[1] Документация по VirtualBox: <https://www.virtualbox.org/>