

Лабораторная работа №1

Архитектура компьютера и операционные системы

Румянцев Артём Олегович

Содержание

1	Цель работы	6
2	Задание	7
3	Выполнение лабораторной работы	8
3.1	Создание виртуальной машины	8
3.2	Установка операционной системы	11
3.3	Работа с операционной системой после установки	18
3.4	Установка программного обеспечения для создания документации	24
4	Выводы	26
5	Ответы на контрольные вопросы	27
5.1	Выполнение домашнего задания	28
	Список литературы	31

Список иллюстраций

3.1	Рис 1	8
3.2	Рис 2	9
3.3	Рис 3	9
3.4	Рис 4	10
3.5	Рис 5	10
3.6	Рис 6	11
3.7	Рис 7	11
3.8	Рис 8	12
3.9	Рис 9	12
3.10	Рис 10	13
3.11	Рис 11	14
3.12	Рис 12	15
3.13	Рис 13	16
3.14	Рис 14	17
3.15	Рис 15	17
3.16	Рис 16	18
3.17	Рис 17	19
3.18	Рис 18	19
3.19	Рис 19	19
3.20	Рис 20	20
3.21	Рис 21	20
3.22	Рис 22	20
3.23	Рис 23	21
3.24	Рис 23	21
3.25	Рис 24	21
3.26	Рис 25	22
3.27	Рис 26	22
3.28	Рис 27	23
3.29	Рис 28	23
3.30	Рис 29	23
3.31	Рис 30	23
3.32	Рис 31	24
3.33	Рис 32	24
3.34	Рис 34	25
3.35	Рис 35	25
3.36	Рис 35	25

5.1	Рис 36	28
5.2	Рис 37	29
5.3	Рис 38	29
5.4	Рис 39	29
5.5	Рис 40	30
5.6	Рис 41	30
5.7	Рис 42	30

Список таблиц

1 Цель работы

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

2 Задание

1.Создание виртуальной машины 2.Установка операционной системы 3.Работа с операционной системой после установки 4.Установка программного обеспечения для создания документации 5.Дополнительные задания

3 Выполнение лабораторной работы

3.1 Создание виртуальной машины

VirtualBox я устанавливал и настраивал при выполнении лабораторной работы на курсе “Архитектура компьютера”, поэтому сразу открываю окно приложения (рис. 1).

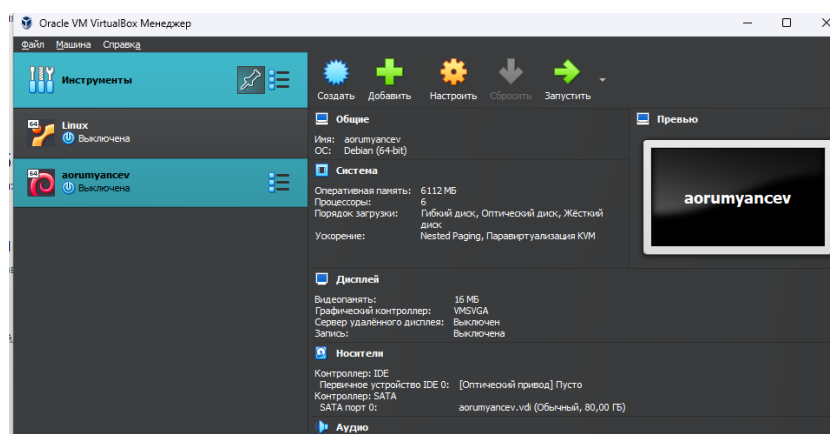


Рис. 3.1: Рис 1

Нажимая “создать”, создаю новую виртуальную машину, указываю её имя, путь к папке машины по умолчанию меня устраивает, выбираю тип ОС и версию (рис. 2).

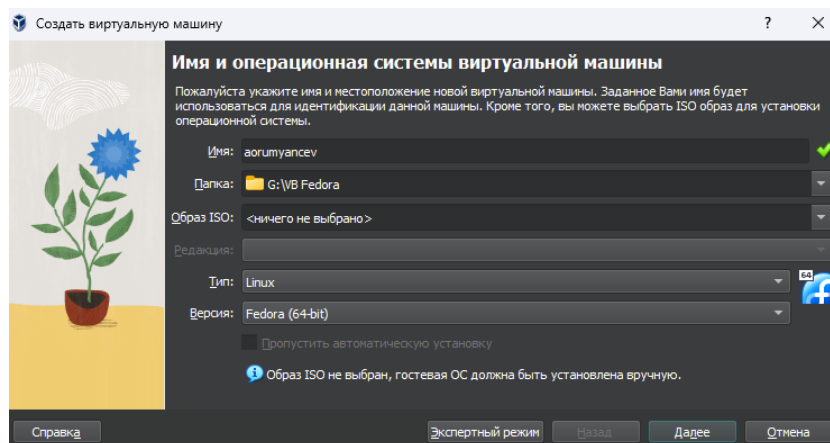


Рис. 3.2: Рис 2

Указываю объем оперативной памяти виртуальной машины размером 6112 МБ и отдаю 6 виртуальных процессора(рис. 3).

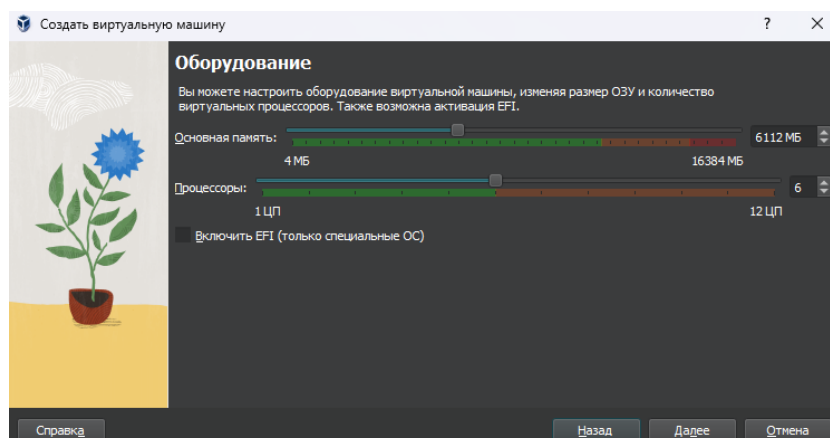


Рис. 3.3: Рис 3

Выбираю создание нового виртуального жесткого диска и выделяю ему память 80 гб(рис. 4).

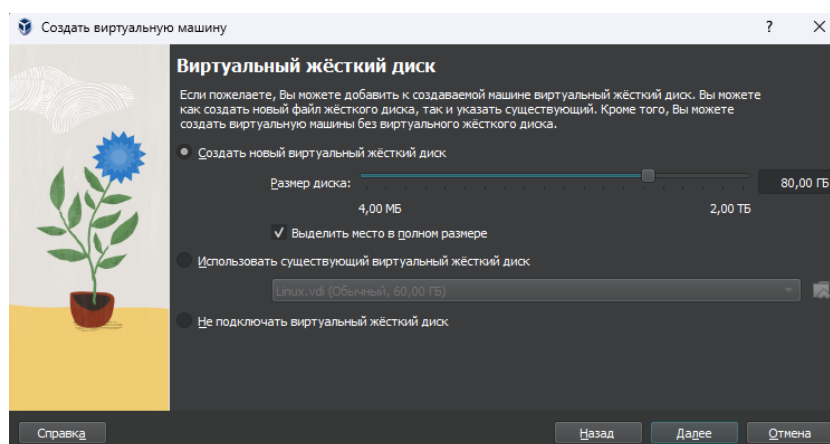
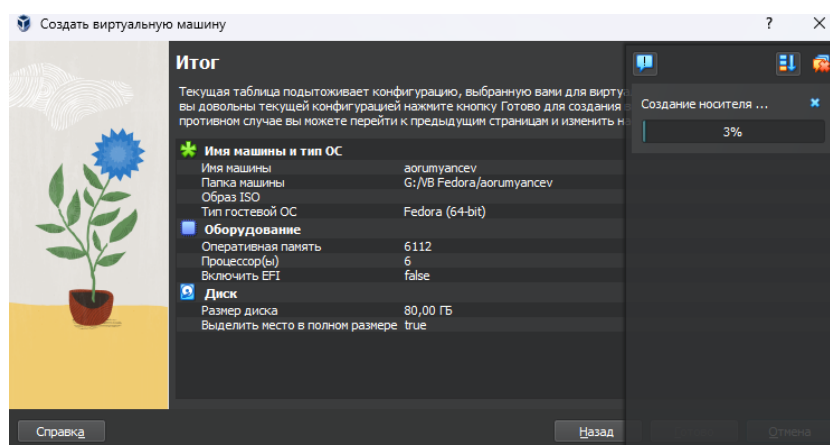


Рис. 3.4: Рис 4

Создание носителя (рис. 5).



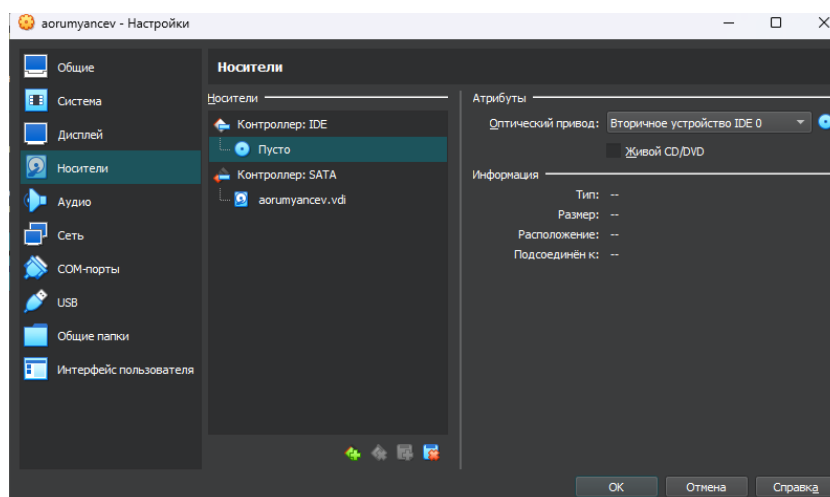


Рис. 3.6: Рис 6

Скачанный образ ОС был успешно выбран (рис. 7).

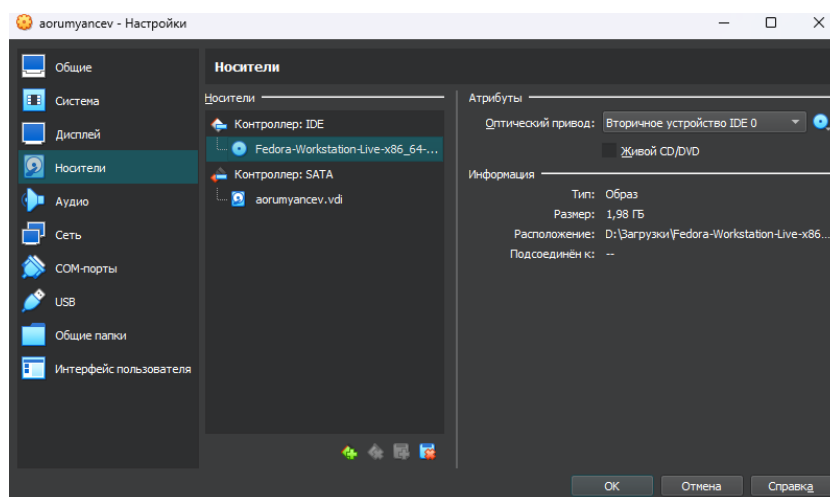


Рис. 3.7: Рис 7

3.2 Установка операционной системы

Запускаю созданную виртуальную машину для установки (рис. 8).

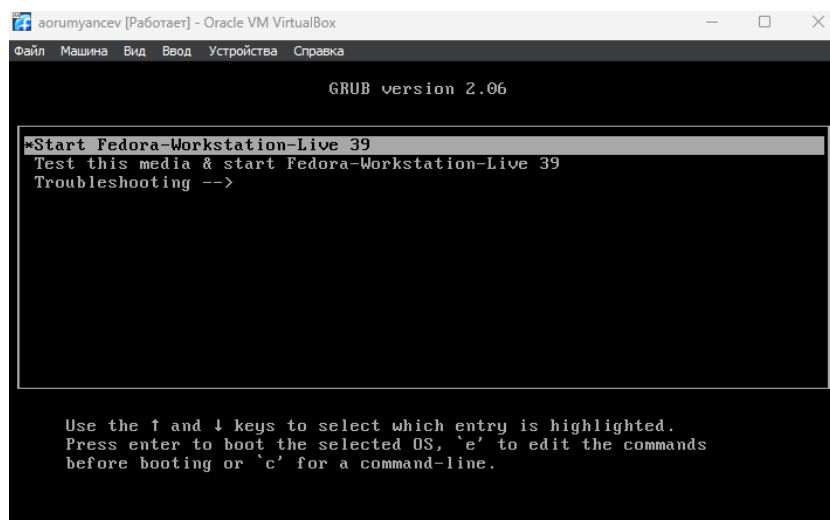


Рис. 3.8: Рис 8

Открываю терминал и ввожу команду liveinst (рис. 9).

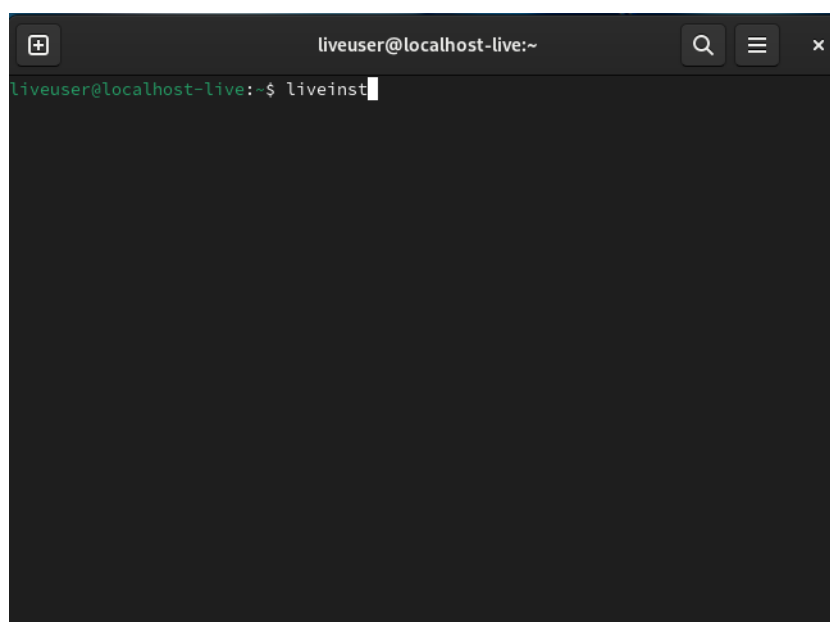


Рис. 3.9: Рис 9

Выбираю язык для использования в процессе установки русский (рис. 10).

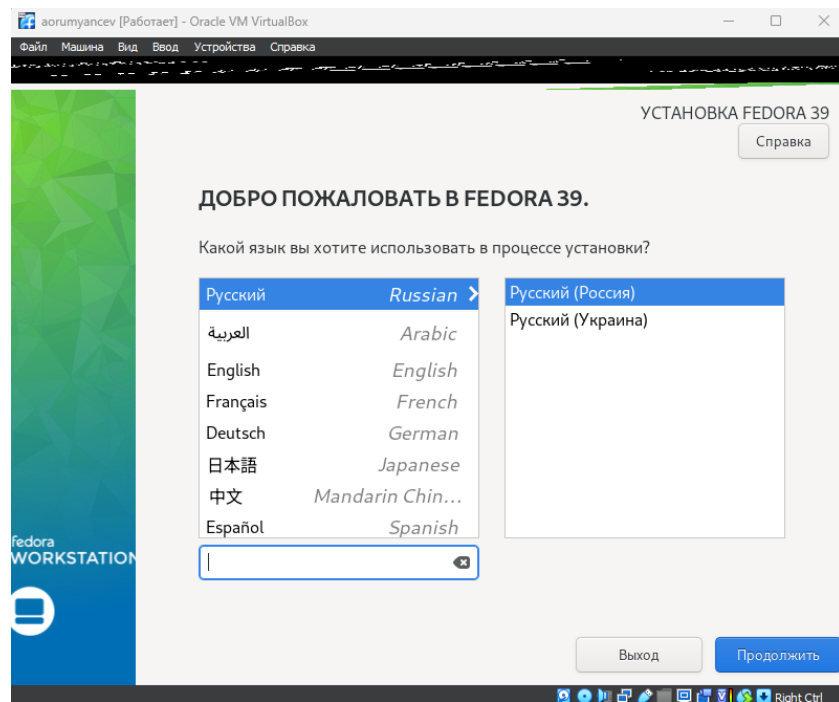


Рис. 3.10: Рис 10

Проверяю место установки значение по умолчанию (рис. 11).

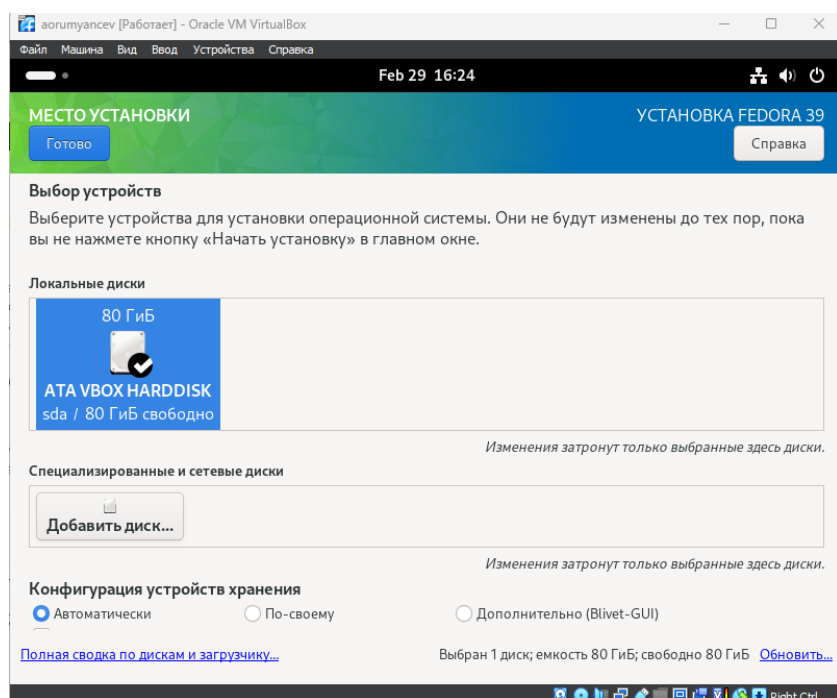


Рис. 3.11: Рис 11

Выбираю раскладку клавиатуры русскую и английскую (рис. 12).

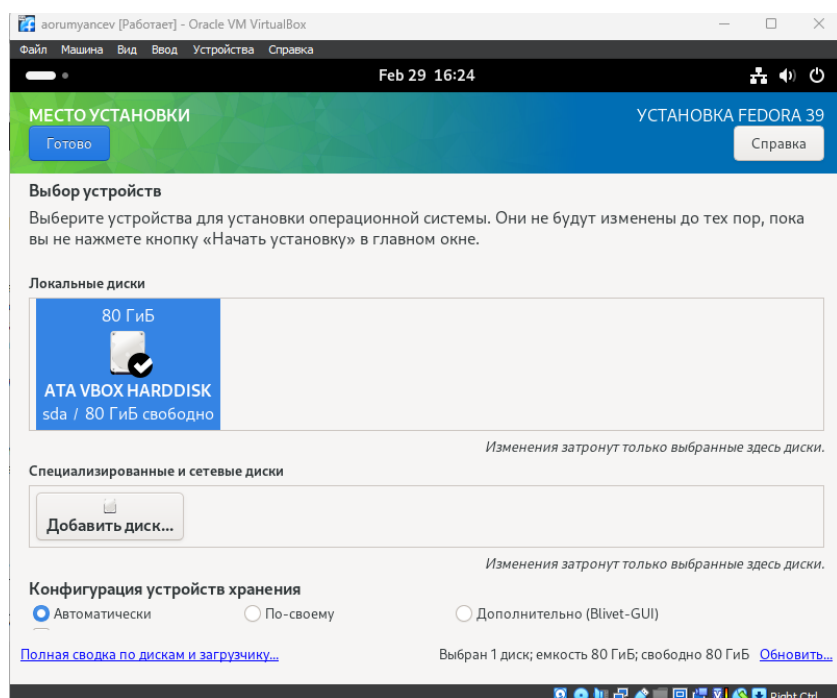


Рис. 3.12: Рис 12

Корректирую часовой пояс, чтобы время совпадало с моим регионом (рис. 13).

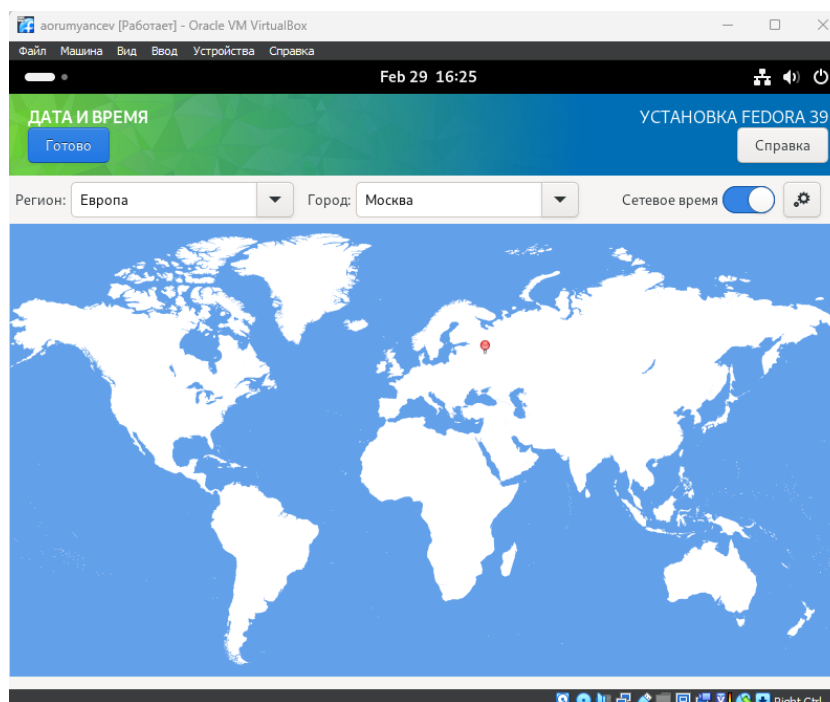


Рис. 3.13: Рис 13

Далее операционная система устанавливается. После установки “завершить установку”(рис. 14).

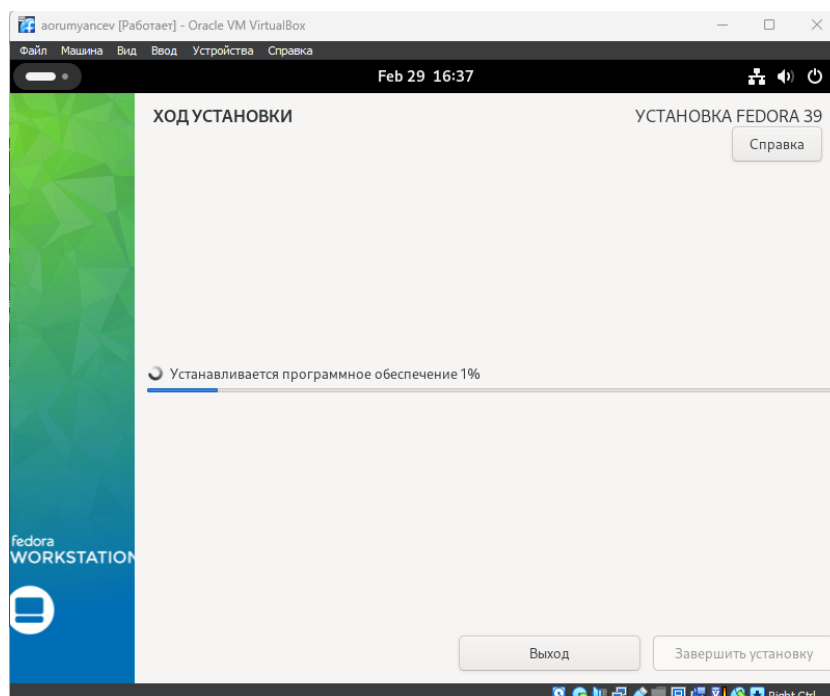


Рис. 3.14: Рис 14

Диск не отключился автоматически, поэтому отключаю носитель информации с образом (рис. 15).

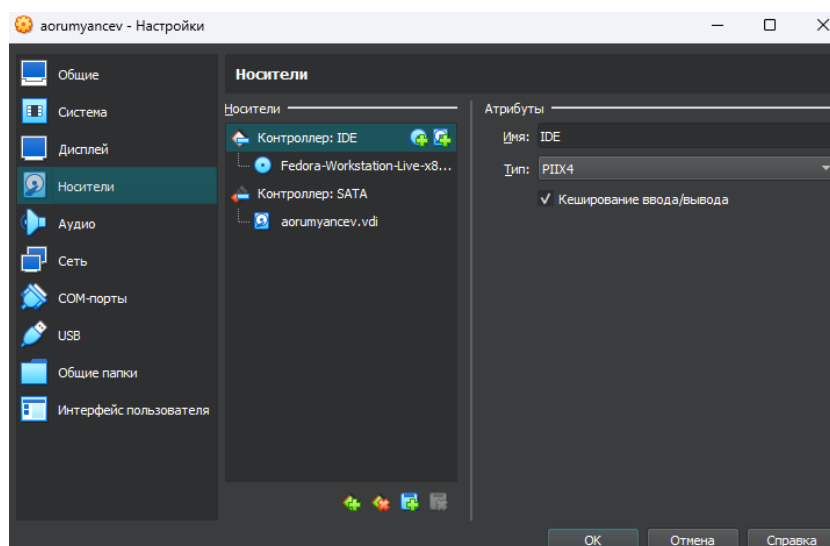


Рис. 3.15: Рис 15

Носитель информации с образом отключен (рис. 16).

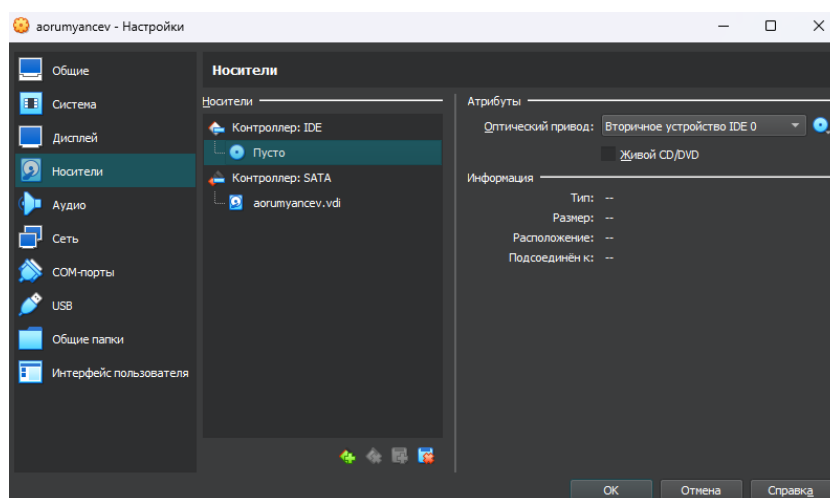


Рис. 3.16: Рис 16

3.3 Работа с операционной системой после установки

Запускаю виртуальную машину. Ввожу свои инициалы и фамилию (рис. 17).

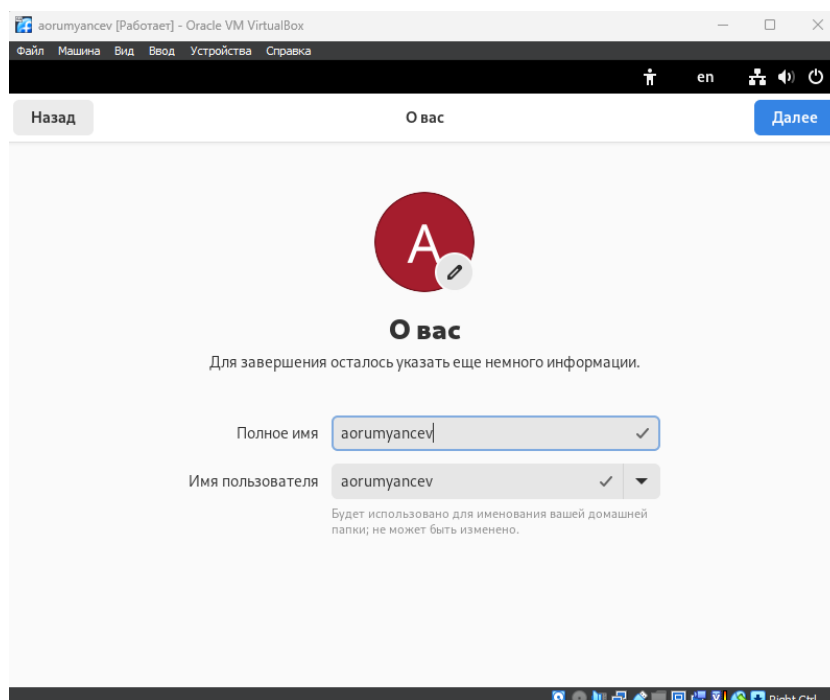


Рис. 3.17: Рис 17

Запускаю терминал и переключаюсь на роль супер-пользователя (рис. 18).



Рис. 3.18: Рис 18

Обновляю все пакеты (рис. 19).



Рис. 3.19: Рис 19

Устанавливаю программы для удобства работы в консоли: `tmux` для открытия нескольких “вкладок” в одном терминале, `mc` в качестве файлового менеджера в терминале (рис. 20).

```
[root@10 ~]# dnf -y install tmux mc
Последняя проверка окончания срока действия метаданных: 0:07:05 назад, Пт 01 мар 2024 00:51:57.
Пакет tmux-3.3a-7.20230918gitb202a2f.fc39.x86_64 уже установлен.
Зависимости разрешены.
```

Пакет	Архитектура	Версия	Репозиторий	Размер
Установка:				
mc	x86_64	1:4.8.30-1.fc39	fedora	1.9 M
Установка зависимостей:				
gpm-libs	x86_64	1.20.7-44.fc39	fedora	20 k
slang	x86_64	2.3.3-4.fc39	updates	433 k

Рис. 3.20: Рис 20

Устанавливаю программы для автоматического обновления (рис. 21).

```
[root@10 ~]# dnf install dnf-automatic
Последняя проверка окончания срока действия метаданных: 0:08:12 назад, Пт 01 мар 2024 00:51:57.
Зависимости разрешены.
```

Пакет	Архитектура	Версия	Репозиторий	Размер
Установка:				
dnf-automatic	noarch	4.19.0-1.fc39	updates	46 k

Рис. 3.21: Рис 21

Запускаю таймер (рис. 22).

```
[root@10 ~]# systemctl enable --now dnf-automatic.timer
Created symlink /etc/systemd/system/timers.target.wants/dnf-automatic.timer → /usr/lib/systemd/system/dnf-automatic.timer.
[root@10 ~]#
```

Рис. 3.22: Рис 22

Перемещаюсь в директорию /etc/selinux,открываю md,ищу нужный файл и редактирую его, заменяю SELINUX=enforcing на SELinux=permissive и перезапускаю машину командой reboot(рис. 23).

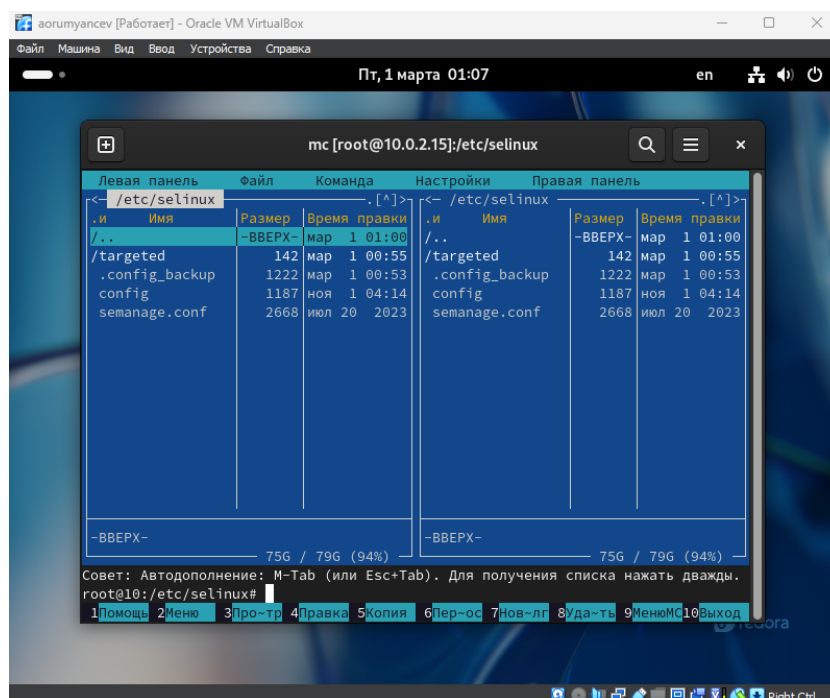


Рис. 3.23: Рис 23



Рис. 3.24: Рис 23

Снова вхожу в ОС, снова запускаю терминал, запускаю терминальный мульт-плексор tmux (рис. 24).



Рис. 3.25: Рис 24

Переключаюсь на роль суперпользователя (рис. 25).

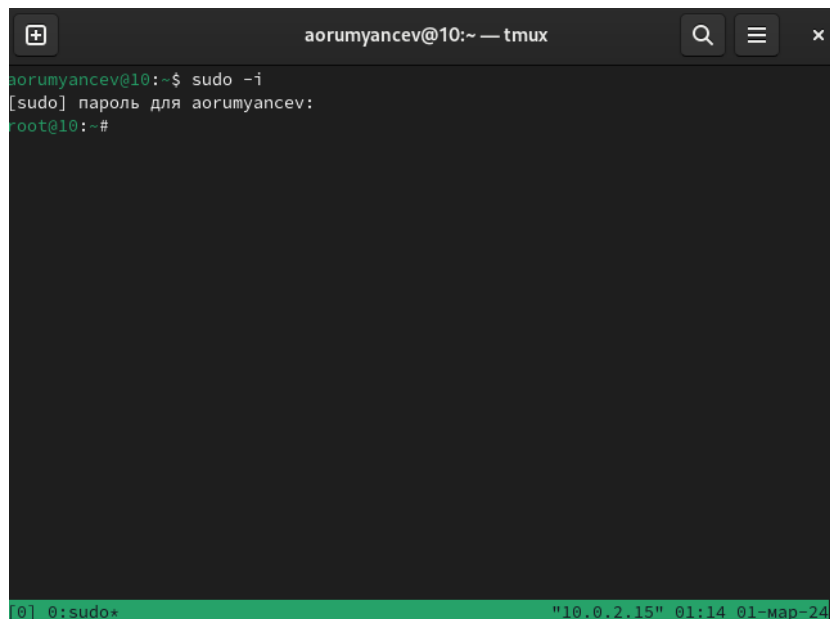


Рис. 3.26: Рис 25

Устанавливаю средства разарботки (рис. 26).

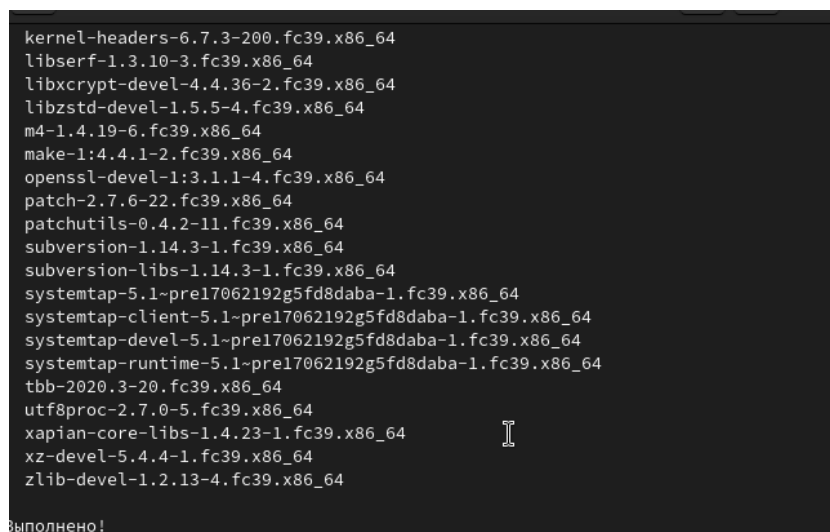


Рис. 3.27: Рис 26

Устанавливаю пакет DKMS (рис. 27).

```
root@10:~# dnf -y install dkms
```

Рис. 3.28: Рис 27

В меню виртуальной машины подключаю образ диска гостевой ОС и примонтирую диск с помощью утилиты mount (рис. 28).

```
root@10:~# mount /dev/sr0 /media
mount: /media: WARNING: source write-protected, mounted read-only.
```

Рис. 3.29: Рис 28

Перезагружаю виртуальную машину (рис. 29).

```
root@10:~# reboot
```

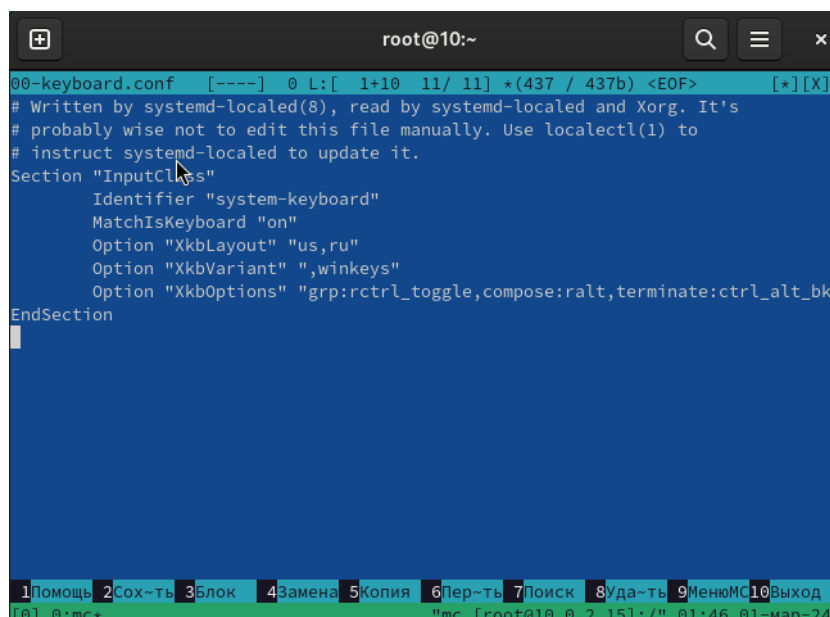
Рис. 3.30: Рис 29

Перехожу в директорию /etc/X11/xorg.conf.d, открываю mc для удобства, открываю файл 00-keyboard.conf (рис. 30).

```
root@10:~# cd /etc/X11/xorg.conf.d/
root@10:/etc/X11/xorg.conf.d# mc
```

Рис. 3.31: Рис 30

Редактирую конфигурационный файл (рис. 31).



```
00-keyboard.conf [-----] 0 L:[ 1+10 11/ 11] *(437 / 437b) <EOF> [*][X]
# Written by systemd-localed(8), read by systemd-localed and Xorg. It's
# probably wise not to edit this file manually. Use localectl(1) to
# instruct systemd-localed to update it.
Section "InputClass"
    Identifier "system-keyboard"
    MatchIsKeyboard "on"
    Option "XkbLayout" "us,ru"
    Option "XkbVariant" ",winkeys"
    Option "XkbOptions" "grp:rctrl_toggle,compose:ralt,terminate:ctrl_alt_bk
EndSection
```

Рис. 3.32: Рис 31

Перезагружаю ОС (рис. 32).



```
root@10:/etc/X11/xorg.conf.d# reboot
```

Рис. 3.33: Рис 32

3.4 Установка программного обеспечения для создания документации

Запускаю терминал. Запускаю терминальный мультиплексор tmux, переключаюсь на роль супер-пользователя и устанавливаю pandoc (рис. 33).



```
root@10:~# dnf -y install pandoc
Последняя проверка окончания срока действия метаданных:
2024 00:51:57.
Зависимости разрешены.
=====
Пакет                Архитектура  Версия
=====
Установка:
pandoc                x86_64       3.1.3-25.fc39
Установка зависимостей:
pandoc-common          noarch       3.1.3-25.fc39
```


Устанавливаю необходимые расширения для pandoc (рис. 34).

```
root@10:~# pip install pandoc-fignos pandoc-egnos pandoc-tablenos pandoc-secnos  
--user
```

Рис. 3.34: Рис 34

Устанавливаю дистрибутив texlive (рис. 35).

```
root@10:~# dnf -y install texlive-scheme-full
```

Рис. 3.35: Рис 35

```
texlive-zitite-11:svn60676-69.fc39.noarch  
texlive-zlmtt-11:svn64076-69.fc39.noarch  
texlive-zootaxa-bst-11:svn50619-69.fc39.noarch  
texlive-zref-11:svn62977-69.fc39.noarch  
texlive-zref-check-11:svn63845-69.fc39.noarch  
texlive-zref-clever-11:svn66021-69.fc39.noarch  
texlive-zref-vario-11:svn65453-69.fc39.noarch  
texlive-zwget-date-11:svn15878.0-69.fc39.noarch  
texlive-zwpage-layout-11:svn63074-69.fc39.noarch  
texlive-zx-calculus-11:svn60838-69.fc39.noarch  
texlive-zxjafont-11:svn28539.0.2-69.fc39.noarch  
texlive-zxjafont-11:svn62864-69.fc39.noarch  
texlive-zxjatype-11:svn53500-69.fc39.noarch  
texlive-zztex-11:svn55862-69.fc39.noarch  
tk-1:8.6.12-5.fc39.x86_64  
tre-0.8.0-41.20140228gitc2f5d13.fc39.x86_64  
tre-common-0.8.0-41.20140228gitc2f5d13.fc39.noarch  
urw-base35-fonts-legacy-20200910-18.fc39.noarch  
xpdf-libs-1:4.04-10.fc39.x86_64  
zziplib-0.13.72-5.fc39.x86_64  
Выполнено!
```

Рис. 3.36: Рис 35

4 Выводы

При выполнении данной лабораторной работы, я приобрел практические навыки установки операционной системы на виртуальную машину, а так же сделал настройки минимально необходимых для дальнейшей работы сервисов.

5 Ответы на контрольные вопросы

1. Учетная запись содержит необходимые для идентификации пользователя при подключении к системе данные, а так же информацию для авторизации и учета: системного имени (user name) (оно может содержать только латинские буквы и знак нижнее подчеркивание, еще оно должно быть уникальным), идентификатор пользователя (UID) (уникальный идентификатор пользователя в системе, целое положительное число), идентификатор группы (GID) (группа, к которой относится пользователь. Она, как минимум, одна, по умолчанию - одна), полное имя (full name) (Могут быть ФИО), домашний каталог (home directory) (каталог, в который попадает пользователь после входа в систему и в котором хранятся его данные), начальная оболочка (login shell) (командная оболочка, которая запускается при входе в систему).

2. Для получения справки по команде: `man`; для перемещения по файловой системе - `cd`; для просмотра содержимого каталога - `ls`; для определения объема каталога - `du`; для создания / удаления каталогов - `mkdir/rmdir`; для создания / удаления файлов - `touch/rm`; для задания определенных прав на файл / каталог - `chmod`; для просмотра истории команд - `history`

3. Файловая система - это порядок, определяющий способ организации и хранения и именования данных на различных носителях информации. Примеры: FAT32 представляет собой пространство, разделенное на три части: одна область для служебных структур, форма указателей в виде таблиц и зона для хранения самих файлов. ext3/ext4 - журналируемая файловая система, используемая в основном в ОС с ядром Linux.

4.С помощью команды `df`, введя ее в терминале. Это утилита, которая показывает список всех файловых систем по именам устройств, сообщает их размер и данные о памяти. Также посмотреть подмонтированные файловые системы можно с помощью утилиты `mount`.

5.Чтобы удалить зависший процесс, вначале мы должны узнать, какой у него `id`: используем команду `ps`. Далее в терминале вводим команду `kill < id процесса >`. Или можно использовать утилиту `killall`, что “убьет” все процессы, которые есть в данный момент, для этого не нужно знать `id` процесса.

5.1 Выполнение домашнего задания

Ввожу в терминале команду `dmesg`, чтобы проанализировать последовательность загрузки системы (рис. 36).

```
aorumyancev@10:~$ dmesg
dmesg: read kernel buffer failed: Операция не позволена
aorumyancev@10:~$ sudo -i
[sudo] пароль для aorumyancev:
root@10:~# dmesg
[    0.000000] Linux version 6.7.6-200.fc39.x86_64 (mockbuild@1fbae28ea38d40908fb246e7adfe592f) (gcc (GCC) 13.2.1 20231205 (Red Hat 13.2.1-6), GNU ld version 2.40-14.fc39) #1 SMP PREEMPT_DYNAMIC Fri Feb 23 18:27:29 UTC 2024
[    0.000000] Command line: BOOT_IMAGE=(hd0,gpt2)/vmlinuz-6.7.6-200.fc39.x86_64 root=UUID=99a3ba6e-e41d-4b85-937f-8769e2620633 ro rootflags=subvol=root rhgb quiet
[    0.000000] [Firmware Bug]: TSC doesn't count with P0 frequency!
[    0.000000] 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 0x0000000001000000-0x0000000000dfffff] usable
[    0.000000] BIOS-e820: [mem 0x000000000dff0000-0x000000000dfffff] ACPI data
[    0.000000] BIOS-e820: [mem 0x00000000fec00000-0x00000000fec0ffff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000fee00000-0x00000000fee0ffff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000fffc0000-0x00000000ffffff] reserved
[    0.000000] BIOS-e820: [mem 0x0000000010000000-0x0000000019dfffff] usable
[    0.000000] NX (Execute Disable) protection: active
[    0.000000] APIC: Static calls initialized
[    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] kvm-clock: Using msrs 4b564d01 and 4b564d00
[    0.000004] kvm-clock: using sched offset of 4344926521 cycles
[    0.000010] clocksource: kvm-clock: mask: 0xffffffffffffff max_cycles: 0x1cd42e4dffb, max_idle_ns: 881590591483 ns
[    0.000014] tsc: Detected 3600.000 MHz processor
```

Рис. 5.1: Рис 36

С помощью поиска, осуществляемого командой `'dmesg | grep -i'`, ищу версию ядра Linux: 6.1.10-200.fc37.x86_64 (рис. 37)).

```

root@10:~# dmesg | grep -i "Linux version"
[ 0.000000] Linux version 6.7.6-200.fc39.x86_64 (mockbuild@1fbae28a38d40908fb246e7adfe592f) (gcc (GCC) 13.2.1 20231205 (Red Hat 13.2.1-6), GNU ld version 2.40-14.fc39) #1 SMP PREEMPT_DYNAMIC Fri Feb 23 18:27:29 UTC 2024
root@10:~#

```

Рис. 5.2: Рис 37

К сожалению, если вводить “Detected Mhz processor” там, где нужно указывать, что я ищу, то мне ничего не выведется. Это происходит потому, что запрос не предусматривает дополнительные символы внутри него (я проверяла, будет ли работать он с маской - не будет). В таком случае я оставила одно из ключевых слов (могла оставить два: “Mhz processor”) и получила результат: 1992 Mhz (рис. 38).

```

root@10:~# dmesg | grep -i "Detected Mhz processor"
root@10:~# dmesg | grep -i "processor"
[ 0.000014] tsc: Detected 3600.000 MHz processor
[ 0.158721] smpboot: CPU0: AMD Ryzen 5 3600 6-Core Processor (family: 0x17, model: 0x71, stepping: 0x0)
[ 0.168289] smpboot: Total of 6 processors activated (43200.00 BogoMIPS)
[ 0.175321] ACPI: Added _OSI(Processor Device)
[ 0.175324] ACPI: Added _OSI(Processor Aggregator Device)
root@10:~#

```

Рис. 5.3: Рис 38

Аналогично ищу модель процессора (рис. 39).

```

root@10:~# dmesg | grep -i "Memory: "
[ 0.011276] PM: hibernation: Registered nosave memory: [mem 0x00000000-0x00000fff]
[ 0.011278] PM: hibernation: Registered nosave memory: [mem 0x0000f000-0x0000ffff]
[ 0.011279] PM: hibernation: Registered nosave memory: [mem 0x000a0000-0x0000ffff]
[ 0.011279] PM: hibernation: Registered nosave memory: [mem 0x000f0000-0x0000ffff]
[ 0.011280] PM: hibernation: Registered nosave memory: [mem 0xdfff0000-0xdfffffff]
[ 0.011281] PM: hibernation: Registered nosave memory: [mem 0xe0000000-0xfefbffff]
[ 0.011281] PM: hibernation: Registered nosave memory: [mem 0xfec00000-0xfec0ffff]
[ 0.011282] PM: hibernation: Registered nosave memory: [mem 0xfec01000-0xfec0ffff]
[ 0.011282] PM: hibernation: Registered nosave memory: [mem 0xfec00000-0xfec0ffff]
[ 0.011283] PM: hibernation: Registered nosave memory: [mem 0xfec01000-0xfefbffff]
[ 0.011283] PM: hibernation: Registered nosave memory: [mem 0xffff0000-0xffffffff]
[ 0.041061] Memory: 598816K/6258232K available (20480K kernel code, 3276K rdata, 14748K rodata, 4588K init, 4892K bss, 269808K reserved, 0K cma-reserved)
[ 0.060205] Freeing SMP alternatives memory: 48K
[ 0.620847] Freeing initrd memory: 32756K
[ 1.009008] Freeing unused decrypted memory: 2028K
[ 1.009000] Freeing unused kernel image (initmem) memory: 4588K
[ 1.010422] Freeing unused kernel image (rodata/data gap) memory: 1636K
root@10:~#

```

Рис. 5.4: Рис 39

Объем доступной оперативной памяти ищу аналогично поиску частоты процессора, т. к. возникла та же проблема, что и там рис. 40).

```

root@10:~# dmesg | grep -i "Hypervisor detected"
[ 0.000000] Hypervisor detected: KVM
root@10:~#

```

Рис. 5.5: Рис 40

Нахожу тип обнаруженного гипервизора (рис. 41).

```

root@10:~# fdisk -l
Disk /dev/sda: 80 GiB, 85899345920 bytes, 167772160 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: D3827BE9-C964-4E9D-AE15-9B2E6459D954

   Device   Start      End  Sectors Size Type
   -----
/dev/sda1    2048      4095     2048  1M BIOS boot
/dev/sda2    4096   2101247   2097152  1G Linux filesystem
/dev/sda3  2101248 167770111 165668864 79G Linux filesystem

Disk /dev/zram0: 5,75 GiB, 6174015488 bytes, 1507328 sectors
Units: sectors of 1 * 4096 = 4096 bytes
Sector size (logical/physical): 4096 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
root@10:~#

```

Рис. 5.6: Рис 41

Тип файловой системы корневого раздела можно посмотреть с помощью утилиты fdisk (рис. 42).

```

root@10:~# dmesg | grep -i "mount"
[ 0.060205] Mount-cache hash table entries: 16384 (order: 5, 131072 bytes, linear)
[ 0.060205] Mountpoint-cache hash table entries: 16384 (order: 5, 131072 bytes, linear)
[ 2.566403] BTRFS: device label fedora devid 1 transid 373 /dev/sda3 scanned by mount (500)
[ 2.569389] BTRFS info (device sda3): first mount of filesystem 99a3ba6e-e41d-4b85-937f-8769e2620633
[ 4.371481] systemd[1]: Set up automount proc-sys-fs-binfmt_misc.automount - Arbitrary Executable File Formats File System Automount Point.
[ 4.397499] systemd[1]: Mounting dev-hugepages.mount - Huge Pages File System...
[ 4.400265] systemd[1]: Mounting dev-mqueue.mount - POSIX Message Queue File System...
[ 4.401969] systemd[1]: Mounting sys-kernel-debug.mount - Kernel Debug File System...
[ 4.403659] systemd[1]: Mounting sys-kernel-tracing.mount - Kernel Trace File System...
[ 4.431602] systemd[1]: Starting systemd-remount-fs.service - Remount Root and Kernel File Systems...
[ 4.436679] systemd[1]: Mounted dev-hugepages.mount - Huge Pages File System.
[ 4.436970] systemd[1]: Mounted dev-mqueue.mount - POSIX Message Queue File System.
[ 4.437298] systemd[1]: Mounted sys-kernel-debug.mount - Kernel Debug File System.
[ 4.437560] systemd[1]: Mounted sys-kernel-tracing.mount - Kernel Trace File System.
[ 5.260234] EXT4-fs (sda2): mounted filesystem cd5a8fdf-2e12-4c83-84d8-6755b2308c27 r/w with ordered data mode. Quota mode: none.
root@10:~#

```

Рис. 5.7: Рис 42

Список литературы

- 1.Dash P. Getting started with oracle vm virtualbox. Packt Publishing Ltd, 2013.
- 2.Colvin H. Virtualbox: An ultimate guide book on virtualization with virtualbox. CreateSpace Independent Publishing Platform, 2015. . van Vugt S. Red hat rhcsa/rhce 7 cert guide : Red hat enterprise linux 7 (ex200 and ex300). Pearson IT Certification, 2016.
- 3.Робачевский А., Немнюгин С., Стесик О. Операционная система unix. 2-е изд. Санкт-Петербург: БХВ-Петербург, 2010.
- 4.Немет Э. et al. Unix и Linux: руководство системного администратора. 4-е изд. Вильямс, 2014.
- 5.Колисниченко Д.Н. Самоучитель системного администратора Linux. СПб.: БХВ-Петербург, 2011.
- 6.Robbins A. Bash pocket reference. O'Reilly Media, 2016.