

Лабораторная работа №1

Архитектура компьютера и операционные системы

Овезов Мерген

Содержание

1	Цель работы.....	1
2	Задание	1
3	Выполнение лабораторной работы.....	1
3.1	Создание виртуальной машины	1
3.2	Установка операционной системы.....	4
3.3	Работа с операционной системой после установки.....	7
3.4	Установка программного обеспечения для создания документации.....	13
4	Выводы	14
5	Ответы на контрольные вопросы	14
5.1	Выполнение домашнего задания.....	15
	Список литературы	18

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).

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

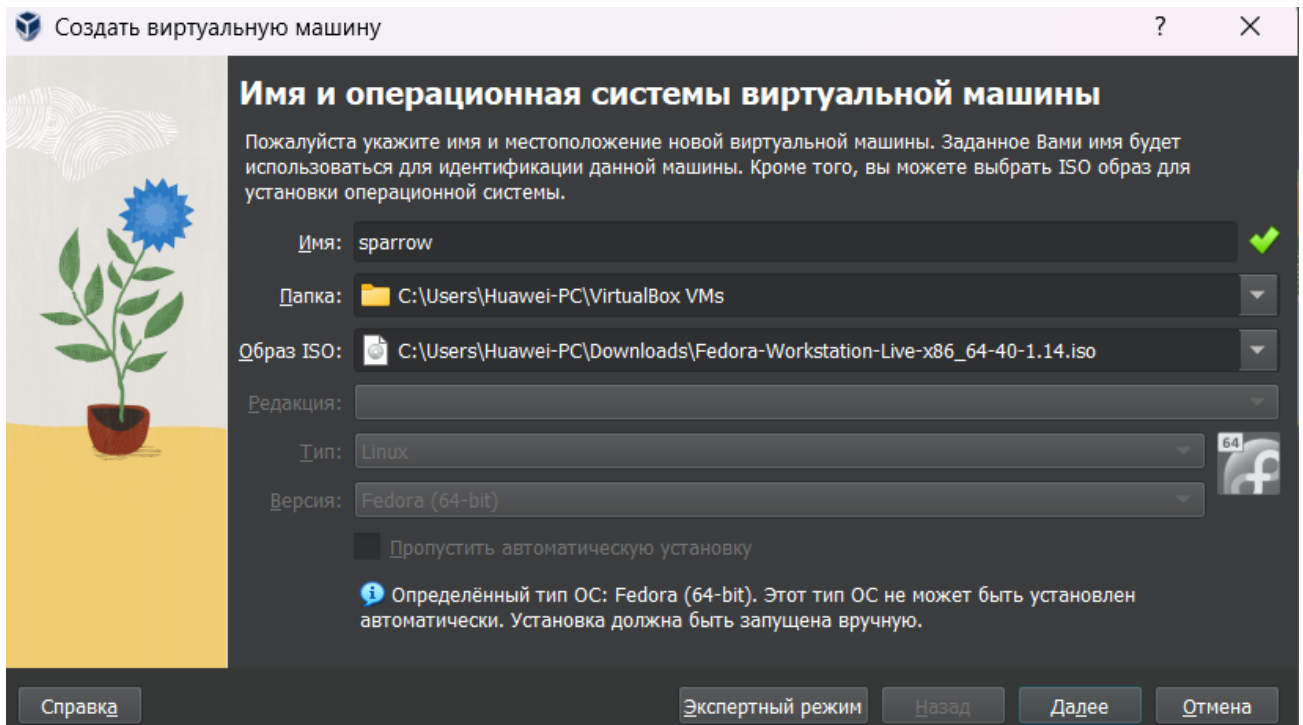


Рис 2

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

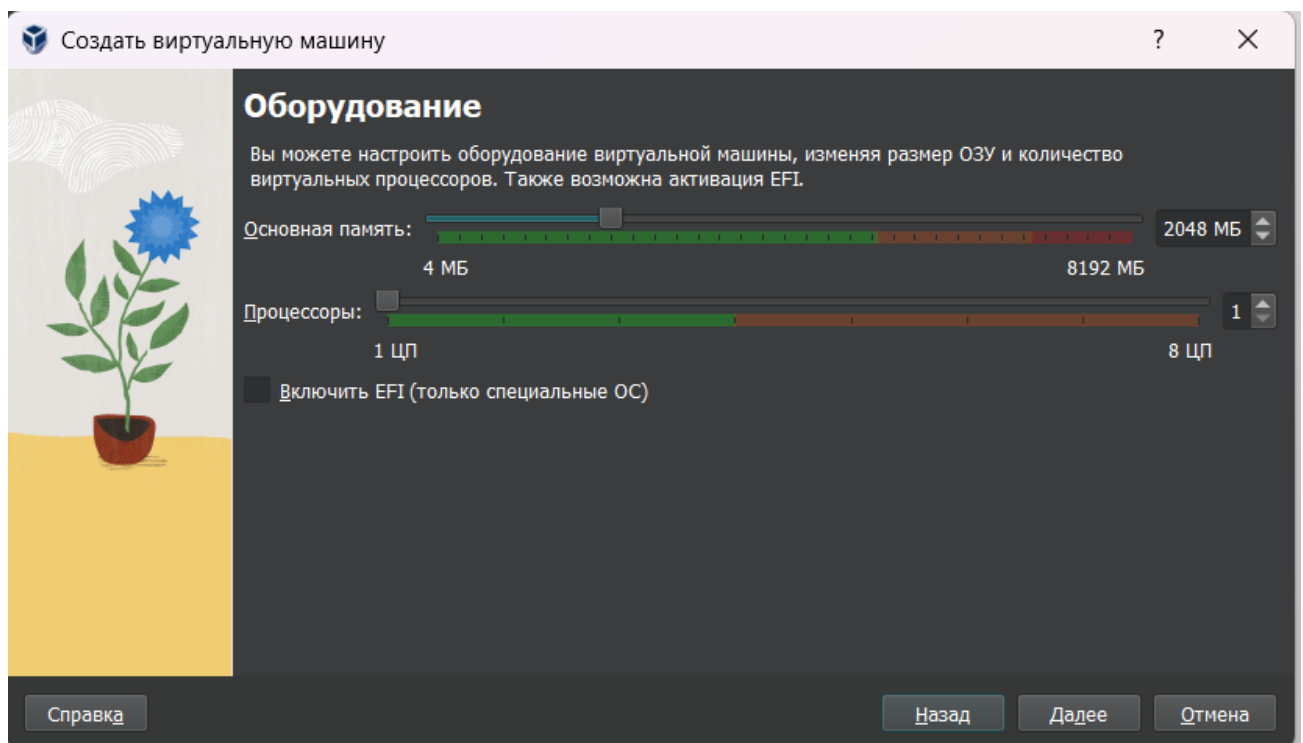


Рис 3

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

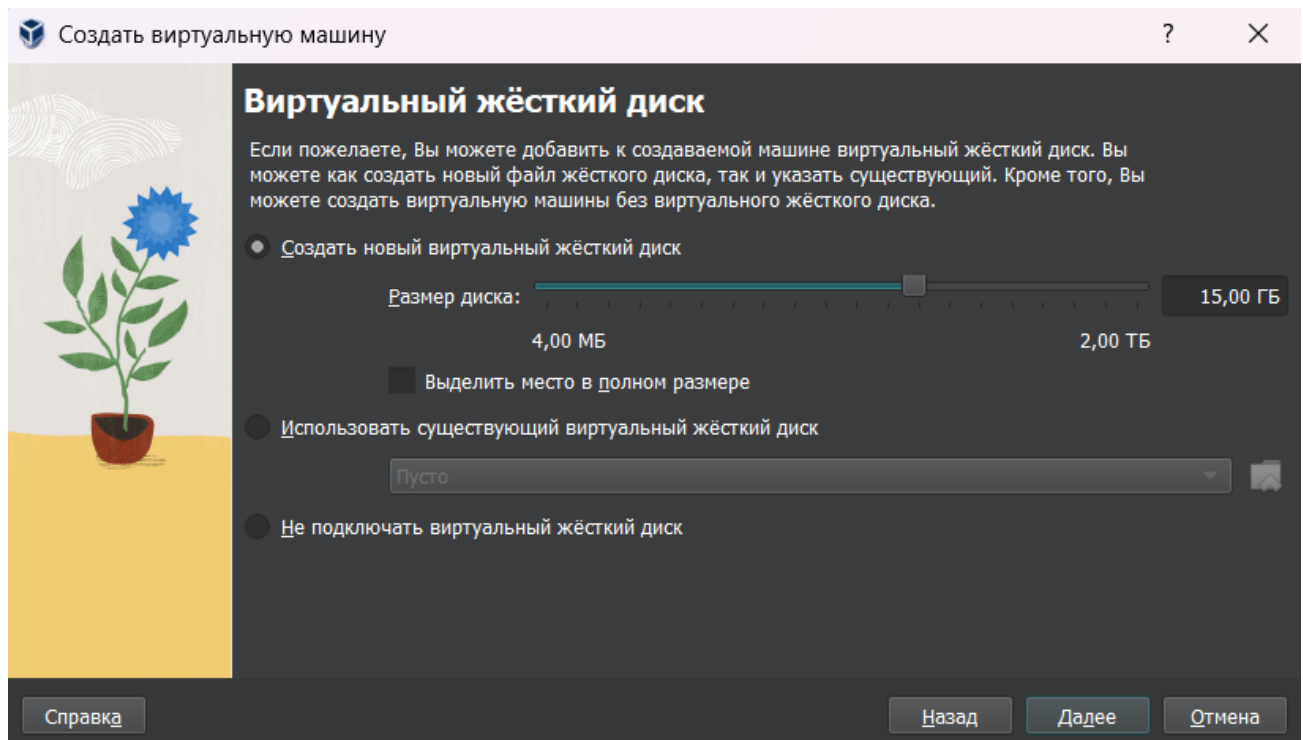


Рис 4

Создание носителя (рис. 5).

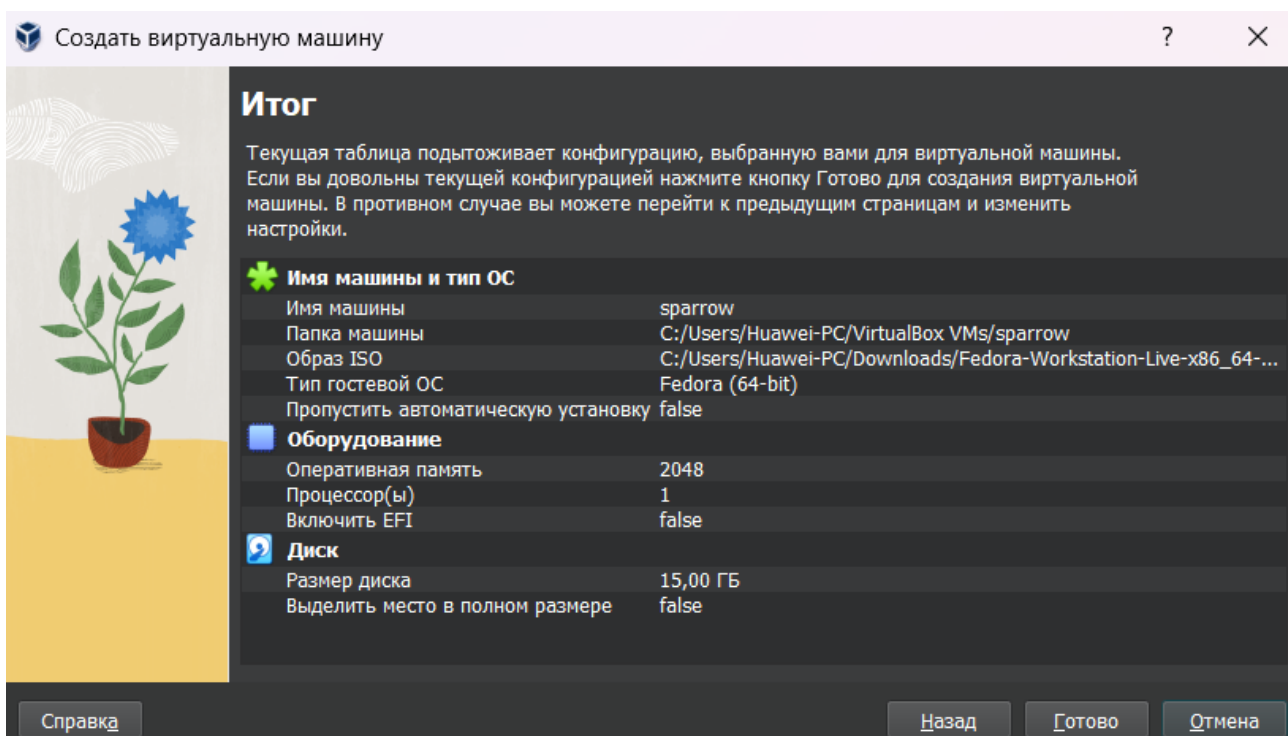


Рис 5

3.2 Установка операционной системы

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



Рис 8

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

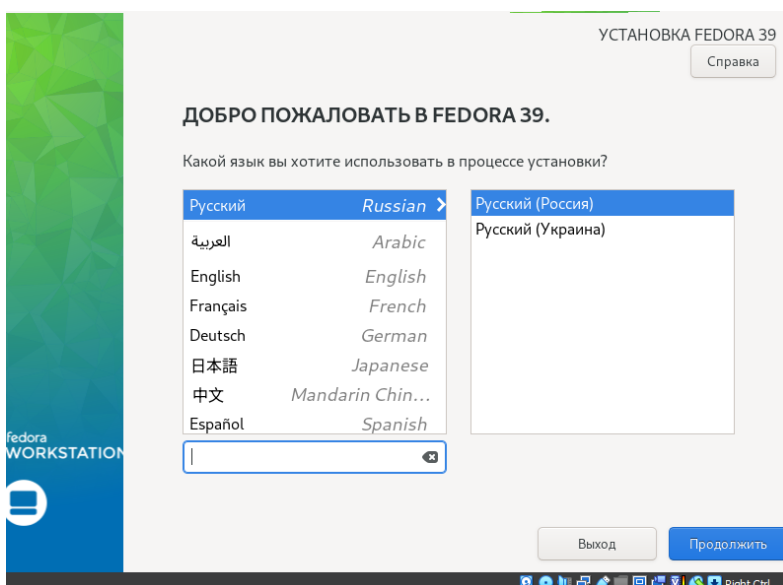


Рис 10

Выбираю раскладку клавиатуры русскую (рис. 12).

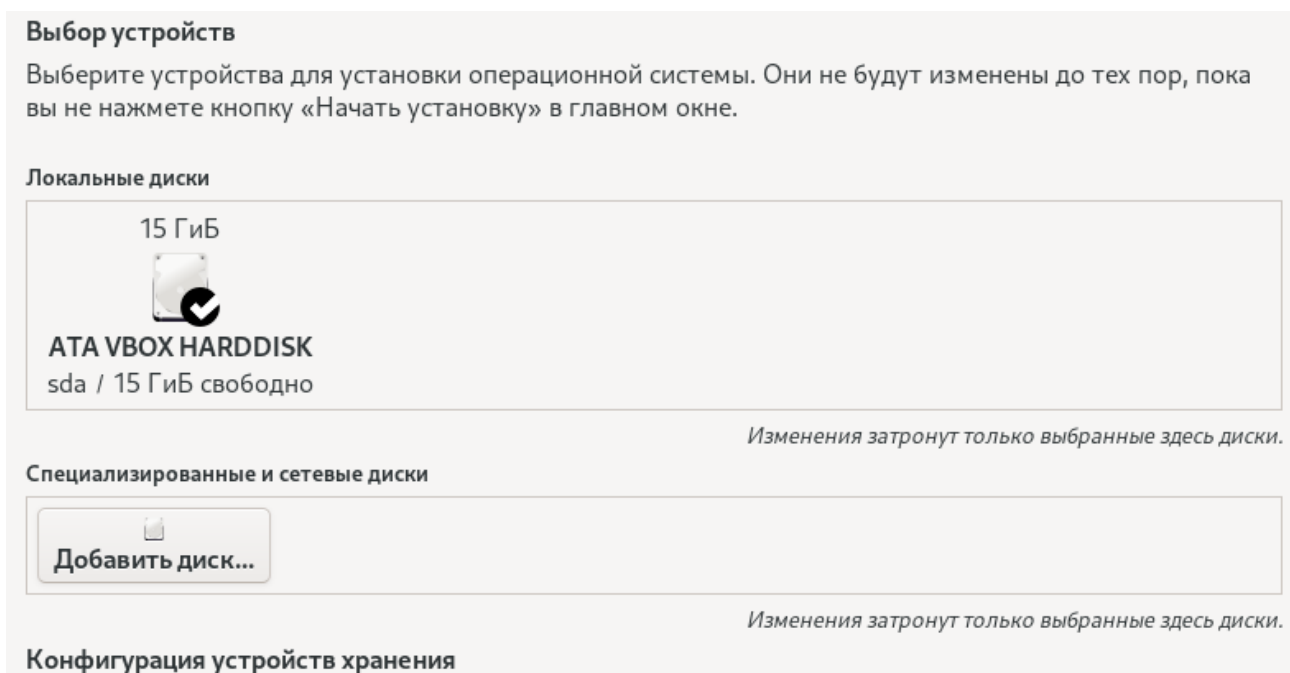


Рис 12

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

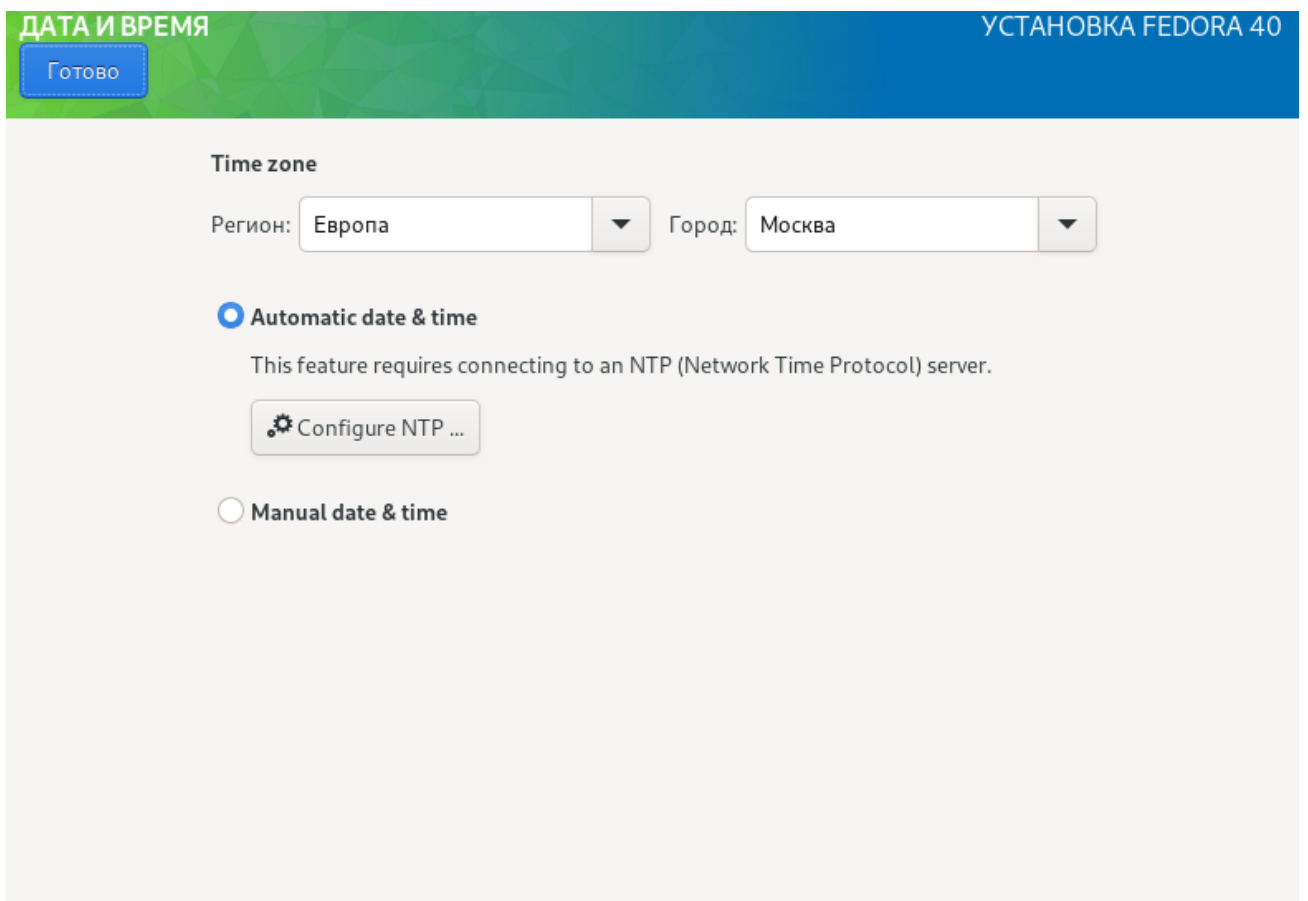


Рис 13

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

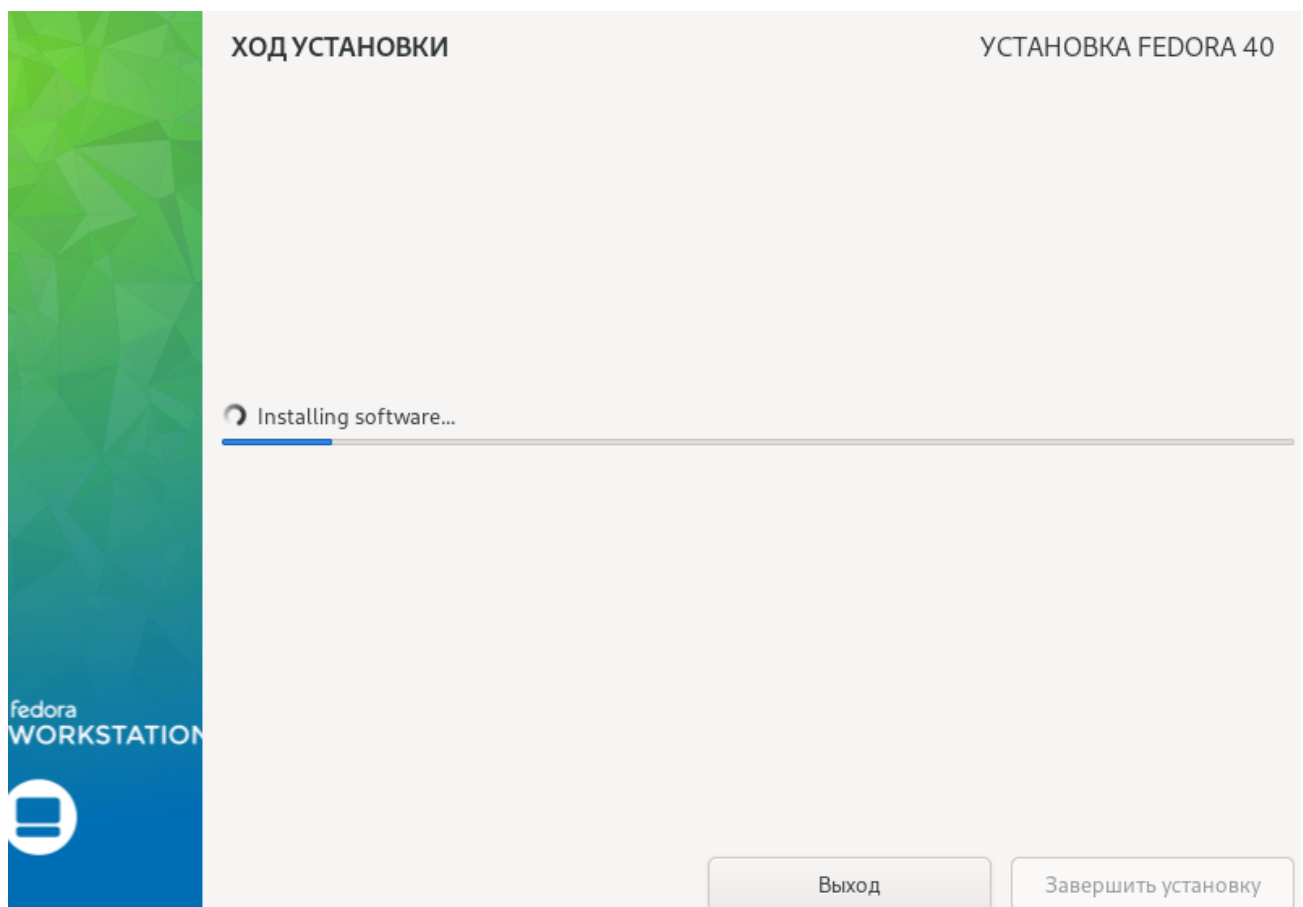


Рис 14

3.3 Работа с операционной системой после установки

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

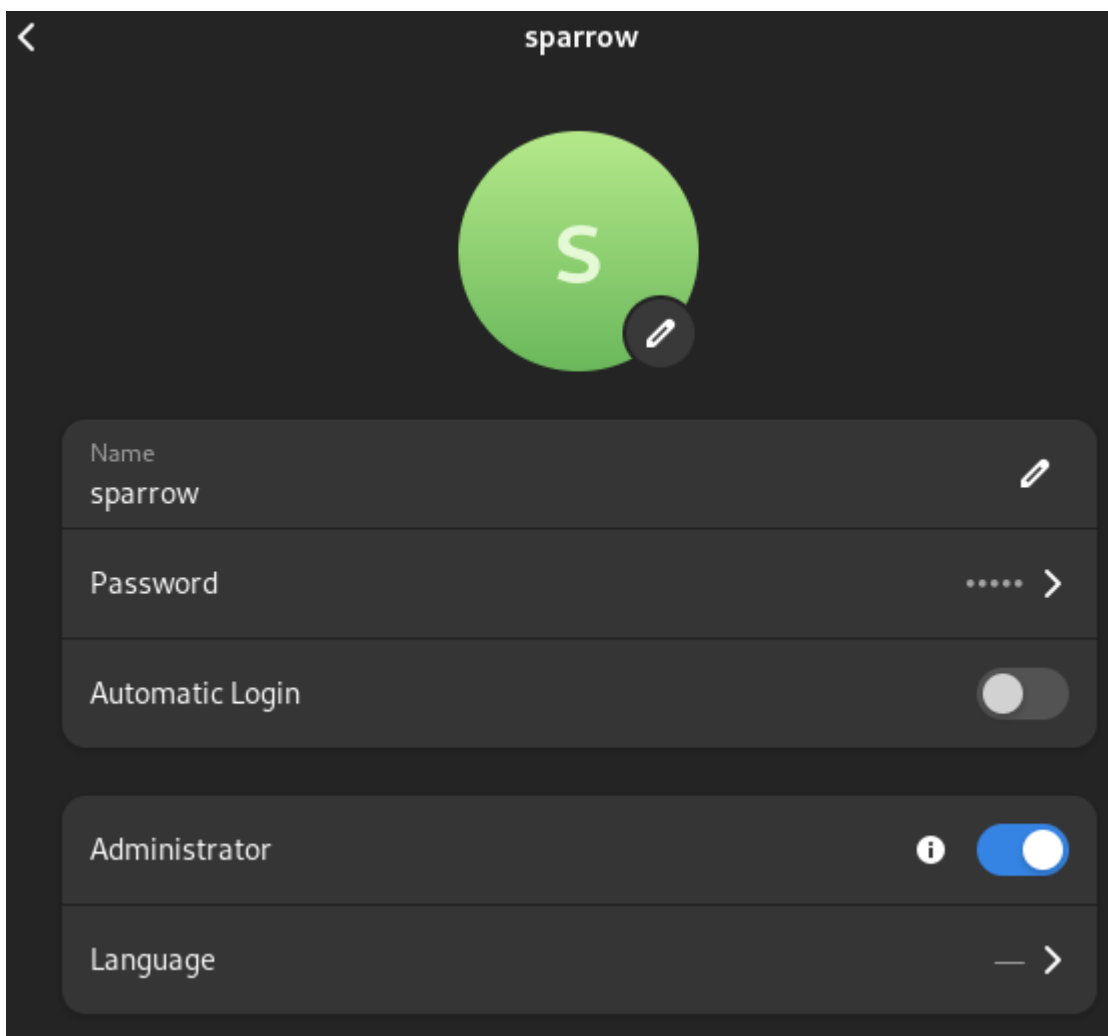


Рис 17

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

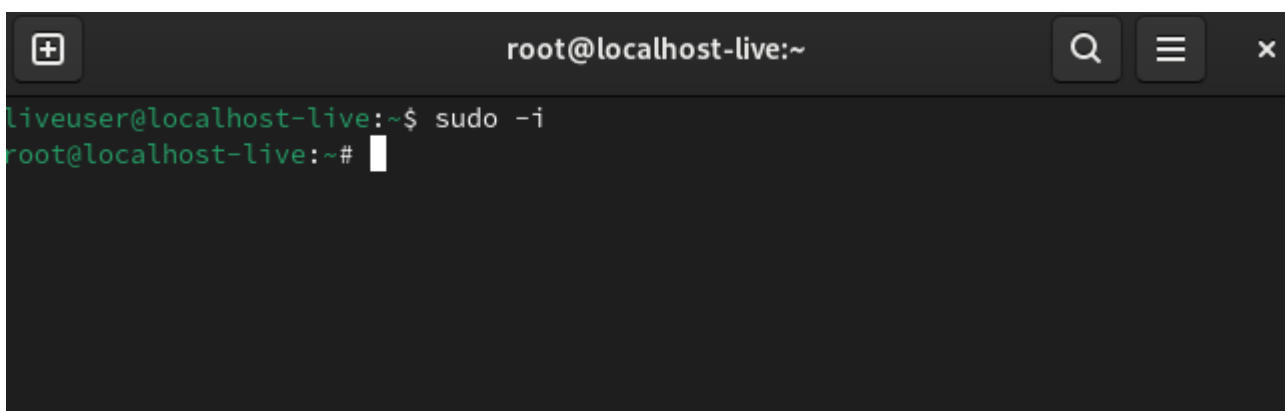


Рис 18

Обновляю все пакеты (рис. 19).


```

root@localhost-live:~# dnf -y update
Fedora 40 - x86_64                                4.2 MB/s | 20 MB      00:04
Fedora 40 openh264 (From Cisco) - x86_64         1.8 kB/s | 1.4 kB     00:00
Fedora 40 - x86_64 - Updates                      2.8 MB/s | 7.7 MB     00:02
Dependencies resolved.

```

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@localhost-live:~# dnf -y install tmux mc
Last metadata expiration check: 0:04:50 ago on Fri 21 Jun 2024 07:29:53 PM EDT.
Package tmux-3.4-1.fc40.x86_64 is already installed.
Dependencies resolved.
=====
Package            Architecture Version                Repository            Size
=====
Installing:
mc                  x86_64         1:4.8.31-1.fc40       fedora                1.9 M
Installing dependencies:
gpm-libs           x86_64         1.20.7-46.fc40       fedora                20 k
slang               x86_64         2.3.3-5.fc40         fedora                415 k

Transaction Summary
=====

```

Рис 20

Устанавливаю программы для автоматического обновления (рис. 21).

```

root@localhost-live:~# dnf install dnf-automatic
Last metadata expiration check: 0:07:12 ago on Fri 21 Jun 2024 07:29:53 PM EDT.
Dependencies resolved.
=====
Package            Architecture Version                Repository            Size
=====
Installing:
dnf-automatic       noarch           4.19.2-1.fc40       updates              45 k
Upgrading:
dnf                  noarch           4.19.2-1.fc40       updates              504 k
dnf-data             noarch           4.19.2-1.fc40       updates              40 k

```

Рис 21

Запускаю таймер (рис. 22).

```
root@localhost-live:~# 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@localhost-live:~#
```

Рис 22

Командой reboot перезапускаю ОС

Рис 23

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

Снова захожу в ОС, снова запускаю терминал, запускаю терминальный мультиплексор tmux (рис. 24).

```
root@localhost-live:~#
```

Рис 24

Переключаюсь на роль суперпользователя (рис. 25).

```
root@localhost-live:~# sudo -i
```

Рис 25

Устанавливаю средства разработки (рис. 26).

```
kernel-devel-6.9.4-200.fc40.x86_64
libserf-1.3.10-5.fc40.x86_64
libxcrypt-devel-4.4.36-5.fc40.x86_64
libzstd-devel-1.5.5-5.fc40.x86_64
m4-1.4.19-9.fc40.x86_64
make-1:4.4.1-6.fc40.x86_64
openssl-devel-1:3.2.1-2.fc40.x86_64
patch-2.7.6-24.fc40.x86_64
patchutils-0.4.2-13.fc40.x86_64
subversion-1.14.3-5.fc40.x86_64
subversion-libs-1.14.3-5.fc40.x86_64
systemtap-5.1-1.fc40.x86_64
systemtap-client-5.1-1.fc40.x86_64
systemtap-devel-5.1-1.fc40.x86_64
systemtap-runtime-5.1-1.fc40.x86_64
tbb-2021.11.0-5.fc40.x86_64
utf8proc-2.7.0-7.fc40.x86_64
xapian-core-libs-1.4.23-2.fc40.x86_64
xz-devel-1:5.4.6-3.fc40.x86_64
zlib-ng-compat-devel-2.1.6-2.fc40.x86_64

complete!
oot@localhost-live:~#
[0] 0:sudo* "localhost-live" 19:45 21-
```

Рис 26

Устанавливаю пакет DKMS (рис. 27).

```

Installing      : dkms-3.0.13-1.fc40.noarch                    5/5
Running scriptlet: dkms-3.0.13-1.fc40.noarch                    5/5
Created symlink /etc/systemd/system/multi-user.target.wants/dkms.service → /usr/lib/systemd/system/dkms.service.

Running scriptlet: kernel-modules-core-6.9.4-200.fc40.x86_64    5/5
Running scriptlet: kernel-core-6.9.4-200.fc40.x86_64            5/5
kms: running auto installation service for kernel 6.9.4-200.fc40.x86_64
kms: autoinstall for kernel 6.9.4-200.fc40.x86_64 Done.
kms: running auto installation service for kernel 6.9.4-200.fc40.x86_64
kms: autoinstall for kernel 6.9.4-200.fc40.x86_64 Done.

Running scriptlet: dkms-3.0.13-1.fc40.noarch                    5/5

Installed:
  dkms-3.0.13-1.fc40.noarch
  kernel-core-6.9.4-200.fc40.x86_64
  kernel-devel-matched-6.9.4-200.fc40.x86_64
  kernel-modules-core-6.9.4-200.fc40.x86_64
  openssl-1:3.2.1-2.fc40.x86_64

Complete!
root@localhost-live:~#
[0] 0:sudo* "localhost-live" 19:48 21-Jun-24

```

Рис 27

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

```

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

```

Рис 28

Перезагружаю виртуальную машину (рис. 29).

```

mount: /media: WARNING: source write-protected, mounted read-only.
root@localhost-live:~# reboot
[0] 0:sudo*

```

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 30

Редактирую конфигурационный файл (рис. 31).

```
# Written by systemd-locale(8), read by systemd-locale and Xorg. It's
# probably wise not to edit this file manually. Use localectl(1) to
# instruct systemd-locale 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 31

Перезагружаю ОС (рис. 32).

```
xorg.conf.d# reboot
```

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@localhost-live:~# sudo -i
root@localhost-live:~# dnf -y install pandoc
Last metadata expiration check: 0:09:36 ago on Fri 21 Jun 2024 07:55:58 PM EDT.
Dependencies resolved.
=====
Package                Architecture Version           Repository      Size
=====
Installing:
pandoc                 x86_64         3.1.3-29.fc40    fedora          26 M
Installing dependencies:
pandoc-common          noarch        3.1.3-29.fc40    fedora          604 k
Transaction Summary
=====
Install 2 Packages

Total download size: 26 M
Installed size: 192 M
Downloading Packages:
(1/2): pandoc-common-3.1.3-29.fc40.noarch.rpm 3.9 MB/s | 604 kB    00:00
(2/2): pandoc-3.1.3-29.fc40.x86_64.rpm       4.8 MB/s | 26 MB     00:05
-----
Total                                         4.0 MB/s | 26 MB     00:06
Running transaction check
```

Устанавливаю необходимые расширения для pandoc (рис. 34).

```
root@localhost-live:~# pip install pandoc-fignos pandoc-tablenos
```

Рис 34

Устанавливаю дистрибутив texlive (рис. 35).

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

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-mario-11:svn65453-69.fc39.noarch
texlive-zwgttdate-11:svn15878.0-69.fc39.noarch
texlive-zwpgelayout-11:svn63074-69.fc39.noarch
texlive-zx-calculus-11:svn60838-69.fc39.noarch
texlive-zxjafbfont-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
zzip-lib-0.13.72-5.fc39.x86_64
```

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 (CID) (группа, к к-рой относится пользователь. Она, как минимум, одна, по умолчанию - одна), полное имя (full name) (Могут быть ФИО), домашний каталог (home directory) (каталог, в к-рый попадает пользователь после входа в систему и в к-ром хранятся его данные), начальная оболочка (login shell) (командная оболочка, к-рая запускается при входе в систему).

2. Для получения справки по команде: -help; 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 определённых 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).

```
root@localhost-live:~# dmesg
[    0.000000] Linux version 6.8.5-301.fc40.x86_64 (mockbuild@0bc0cc78c12e4762acf61c209bd02e96) (gcc (GCC) 14.0.1 20240328 (Red Hat 14.0.1-0), GNU ld version 2.41-34.fc40) #1 SMP PREEMPT_DYNAMIC Thu Apr 11 20:00:10 UTC 2024
[    0.000000] Command line: BOOT_IMAGE=/images/pxeboot/vmlinuz root=live:CDLABEL=Fedora-WS-Live-40-1-14 rd.live.image quiet rhgb
[    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 0x0000000000100000-0x0000000000c79effff] usable
[    0.000000] BIOS-e820: [mem 0x0000000000c79f0000-0x0000000000c79fffff] ACPI data
[    0.000000] BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000fee00000-0x00000000fee00fff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000fffc0000-0x00000000ffffffff] reserved
[    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
[    0.000003] kvm-clock: using sched offset of 1465599341670 cycles
```

Рис 36

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

```

root@localhost-live:~# dmesg | grep -i "Linux version"
[    0.000000] Linux version 6.8.5-301.fc40.x86_64 (mockbuild@0bc0cc78c12e4762acf61c209bd02e96) (gcc (GCC) 14.0.1 20240328 (Red Hat 14.0.1-0), GNU ld version 2.41-34.fc40) #1 SMP PREEMPT_DYNAMIC Thu Apr 11 20:00:10 UTC 2024

```

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@localhost-live:~# dmesg | grep -i "Detected Mhz processor"
root@localhost-live:~# dmesg | grep -i "processor"
[    0.000008] tsc: Detected 2495.980 MHz processor
[    0.193071] smpboot: Total of 2 processors activated (9983.92 BogoMIPS)
[    0.207894] ACPI: Added _OSI(Processor Device)
[    0.207894] ACPI: Added _OSI(Processor Aggregator Device)
root@localhost-live:~#

```

Рис 38

Аналогично ищу модель процессора (рис. 39).

```

root@localhost-live:~# dmesg | grep -i "Memory"
[    0.001740] ACPI: Reserving FACP table memory at [mem 0xc79f00f0-0xc79f01e3]
[    0.001741] ACPI: Reserving DSDT table memory at [mem 0xc79f0610-0xc79f2962]
[    0.001742] ACPI: Reserving FACS table memory at [mem 0xc79f0200-0xc79f023f]
[    0.001742] ACPI: Reserving FACS table memory at [mem 0xc79f0200-0xc79f023f]
[    0.001743] ACPI: Reserving APIC table memory at [mem 0xc79f0240-0xc79f029b]
[    0.001743] ACPI: Reserving SSDT table memory at [mem 0xc79f02a0-0xc79f060b]
[    0.002672] Early memory node ranges
[    0.009312] PM: hibernation: Registered nosave memory: [mem 0x00000000-0x00000fff]
[    0.009314] PM: hibernation: Registered nosave memory: [mem 0x0009f000-0x0009ffff]
[    0.009314] PM: hibernation: Registered nosave memory: [mem 0x000a0000-0x000effff]
[    0.009315] PM: hibernation: Registered nosave memory: [mem 0x000f0000-0x000fffff]
[    0.021096] Memory: 3016352K/3270200K available (20480K kernel code, 3275K rw data, 14792K rodata, 4628K init, 4776K bss, 253588K reserved, 0K cma-reserved)
[    0.082894] Freeing SMP alternatives memory: 48K
[    0.106153] x86/mm: Memory block size: 128MB

```

Рис 39

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


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

Рис 40

Нахожу тип обнаруженного гипервизора (рис. 41).

```
root@localhost-live:~# fdisk -l
Disk /dev/sda: 15 GiB, 16106127360 bytes, 31457280 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: 74BA7B25-BA8A-4947-AA4F-4CEED47929B8

   Device   Start      End  Sectors  Size Type
/dev/sda1    2048      4095     2048    1M BIOS boot
/dev/sda2    4096   2101247  2097152    1G Linux extended boot
/dev/sda3  2101248 31455231 29353984   14G Linux filesystem

Disk /dev/loop0: 1.97 GiB, 2110570496 bytes, 4122208 sectors
Units: sectors of 1 * 512 = 512 bytes
```

Рис 41

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

```

root@localhost-live:~# dmesg | grep -i "mount"
[ 0.082894] Mount-cache hash table entries: 8192 (order: 4, 65536 bytes, linear)
[ 0.082894] Mountpoint-cache hash table entries: 8192 (order: 4, 65536 bytes, linear)
[ 6.195128] EXT4-fs (dm-0): mounted filesystem 57b2da08-9667-4695-9f1f-13d099a97592 r/w with ordered data mode. Quota mode: none.
[ 9.116641] systemd[1]: Set up automount proc-sys-fs-binfmt_misc.automount - Arbitrary Executable File Formats File System Automount Point.
[ 9.129805] systemd[1]: Mounting dev-hugepages.mount - Huge Pages File System
...
[ 9.132476] systemd[1]: Mounting dev-mqueue.mount - POSIX Message Queue File System...
[ 9.138903] systemd[1]: Mounting sys-kernel-debug.mount - Kernel Debug File System...
[ 9.144352] systemd[1]: Mounting sys-kernel-tracing.mount - Kernel Trace File System...
[ 9.240470] systemd[1]: Starting systemd-remount-fs.service - Remount Root and Kernel File Systems...
[ 9.256706] systemd[1]: Mounted dev-hugepages.mount - Huge Pages File System.
[ 9.256813] systemd[1]: Mounted dev-mqueue.mount - POSIX Message Queue File System.
[ 9.256874] systemd[1]: Mounted sys-kernel-debug.mount - Kernel Debug File System.

```

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.