

Отчёта по лабораторной работе NO.1

Операционные Системы

Джаллох Ишмаил

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	12
3.4	Установка программного обеспечения для создания документации	16
4	Выводы	18
5	Выполнение дополнительного задания	19
6	Ответы на контрольные вопросы	22
	Список литературы	24

Список иллюстраций

3.1	Окно Virtualbox	8
3.2	Создание виртуальной машины	9
3.3	Указание объема памяти	9
3.4	Выбор образа оптического диска	10
3.5	Выбранный образ оптического диска	10
3.6	Выбор языка интерфейса	11
3.7	Выбор раскладки клавиатуры	11
3.8	Завершение установки операционной системы	12
3.9	Запуск терминала	12
3.10	Обновления	12
3.11	Установка tmux и mc	13
3.12	Установка программного обеспечения для автоматического обновления	13
3.13	Запуск таймера	13
3.14	Поиск файла	13
3.15	Изменение файла	14
3.16	Перезагрузка виртуальной машины	14
3.17	Запуск терминального мультиплексора	14
3.18	Переключение на роль супер-пользователя	14
3.19	Установка пакета dkms	15
3.20	Примонтирование диска	15
3.21	Установка драйвера	15
3.22	Перезагрузка виртуальной машины	15
3.23	Поиск файла, вход в mc	15
3.24	Редактирование файла	16
3.25	Перезагрузка виртуальной машины	16
3.26	Переключение на роль супер-пользователя	16
3.27	Установка pandoc	16
3.28	Установка расширения pandoc	17
3.29	Установка texlive	17
5.1	Анализ последовательности загрузки системы	19
5.2	Поиск версии ядра	19
5.3	Поиск частоты процессора	20
5.4	Поиск модели процессора	20
5.5	Поиск объема доступной оперативной памяти	20
5.6	Поиск типа обнаруженного гипервизора	20

5.7	Последовательность монтирования файловых систем	21
-----	---	----

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 устанавливала и настраивала при выполнении лабораторной работы в курсе “Архитектура компьютера и Операционные системы (раздел”Архитектура компьютера”)“, поэтому сразу открываю окно приложения (рис. fig. 3.1).

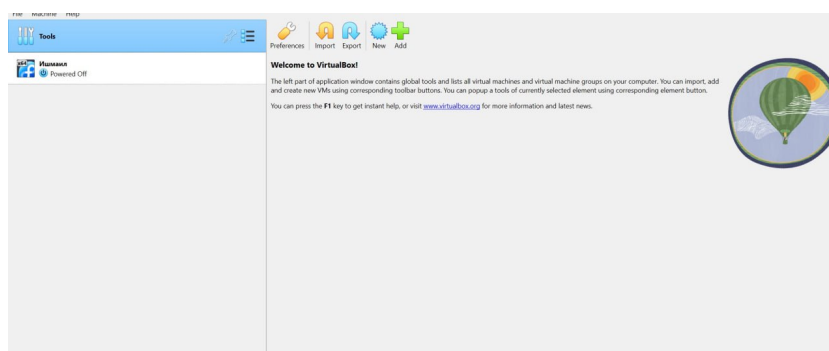


Рис. 3.1: Окно Virtualbox

Нажимая “создать”, создаю новую виртуальную машину, указываю ее имя, путь к папке машины по умолчанию меня устраивает, выбираю тип ОС и версию (рис. fig. 3.2).

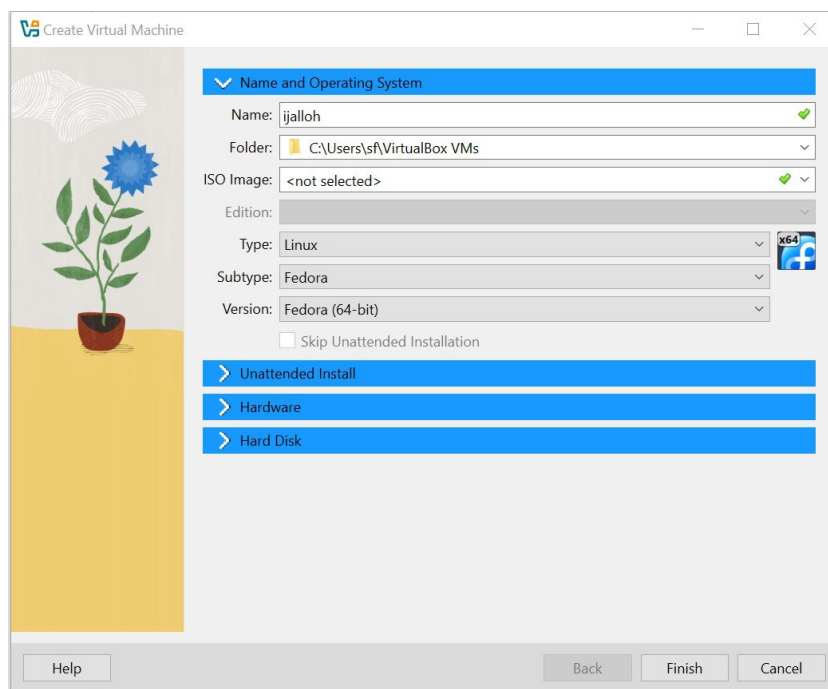


Рис. 3.2: Создание виртуальной машины

Указываю объем основной памяти виртуальной машины размером 4096МБ (рис. fig. 3.3).

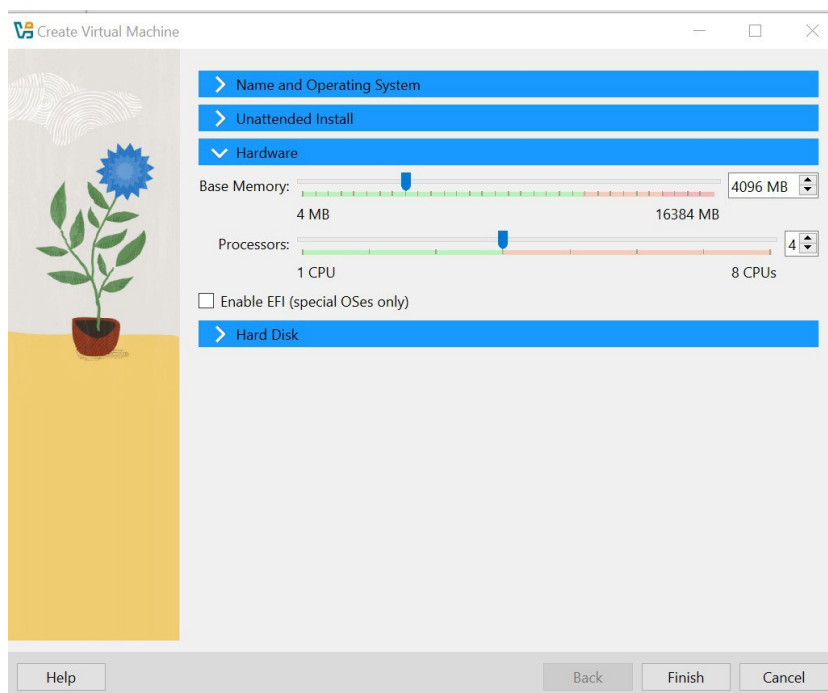


Рис. 3.3: Указание объема памяти

Выбираю в Virtualbox настройку своей виртуальной машины. Перехожу в “Носители”, добавляю новый привод оптических дисков и выбираю скачанный образ операционной системы Fedora (рис. fig. 3.4).

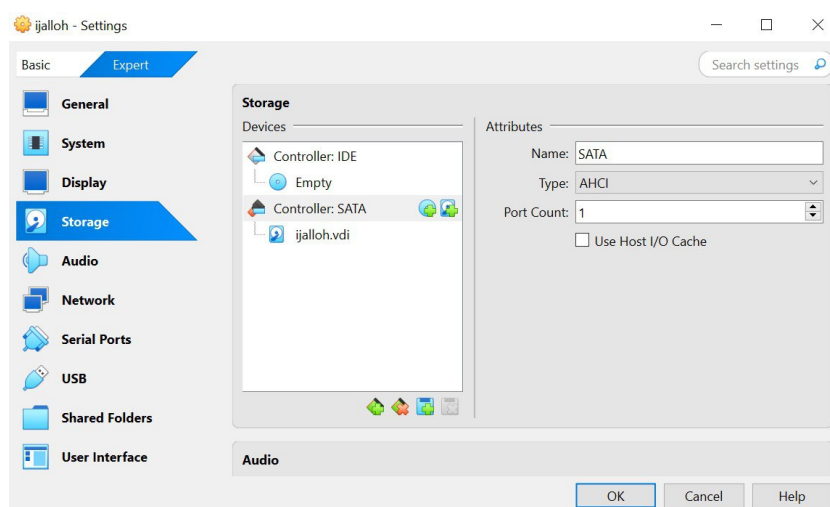


Рис. 3.4: Выбор образа оптического диска

Скачанный образ ОС был успешно выбран (рис. fig. 3.5).

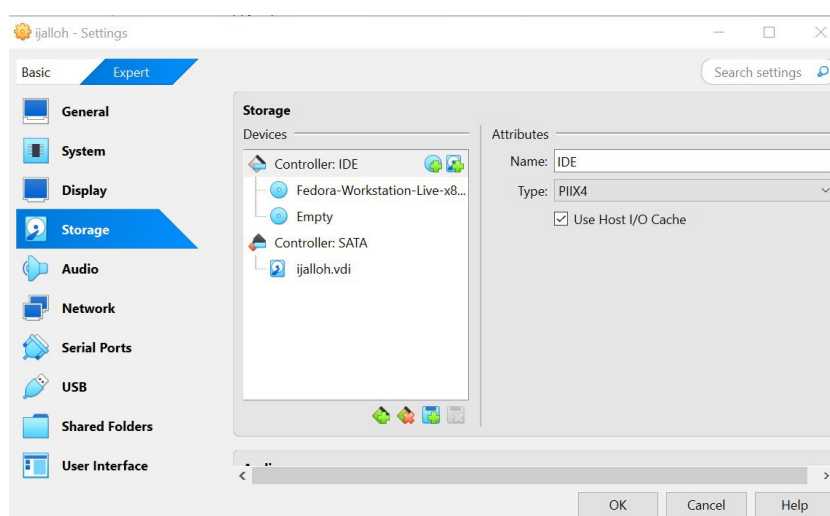


Рис. 3.5: Выбранный образ оптического диска

3.2 Установка операционной системы

Чтобы перейти к раскладке окон с табами, нажимаю Win+w. Выбираю язык для использования в процессе установки русски (рис. fig. 3.6).

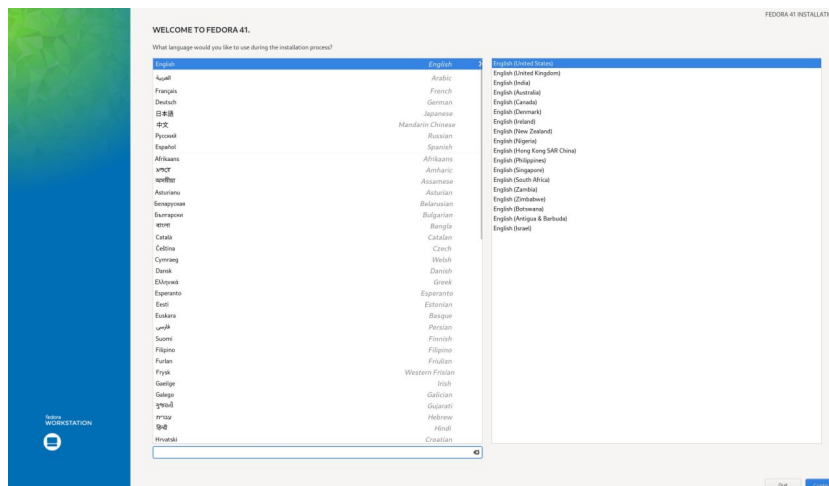


Рис. 3.6: Выбор языка интерфейса

Раскладку клавиатуры выбираю и русскую, и английскую (рис. fig. 3.7).

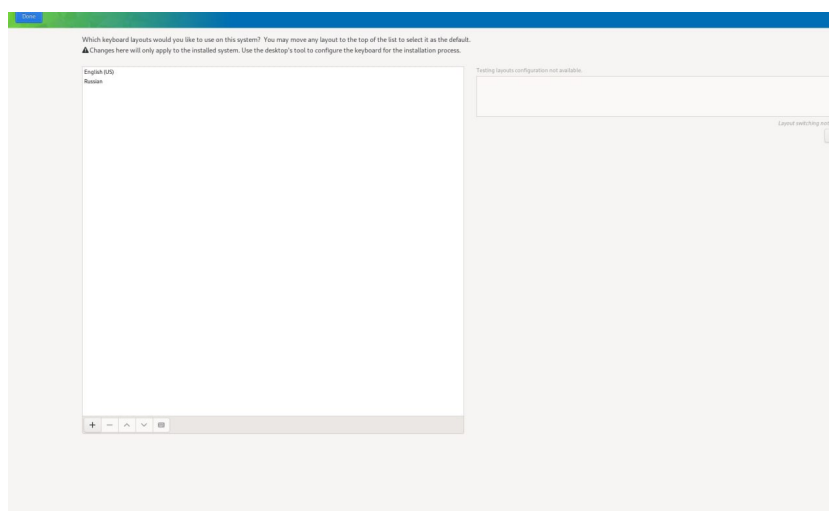


Рис. 3.7: Выбор раскладки клавиатуры

Далее операционная система устанавливается. После установки нажимаю “завершить установку” (рис. fig. 3.8).

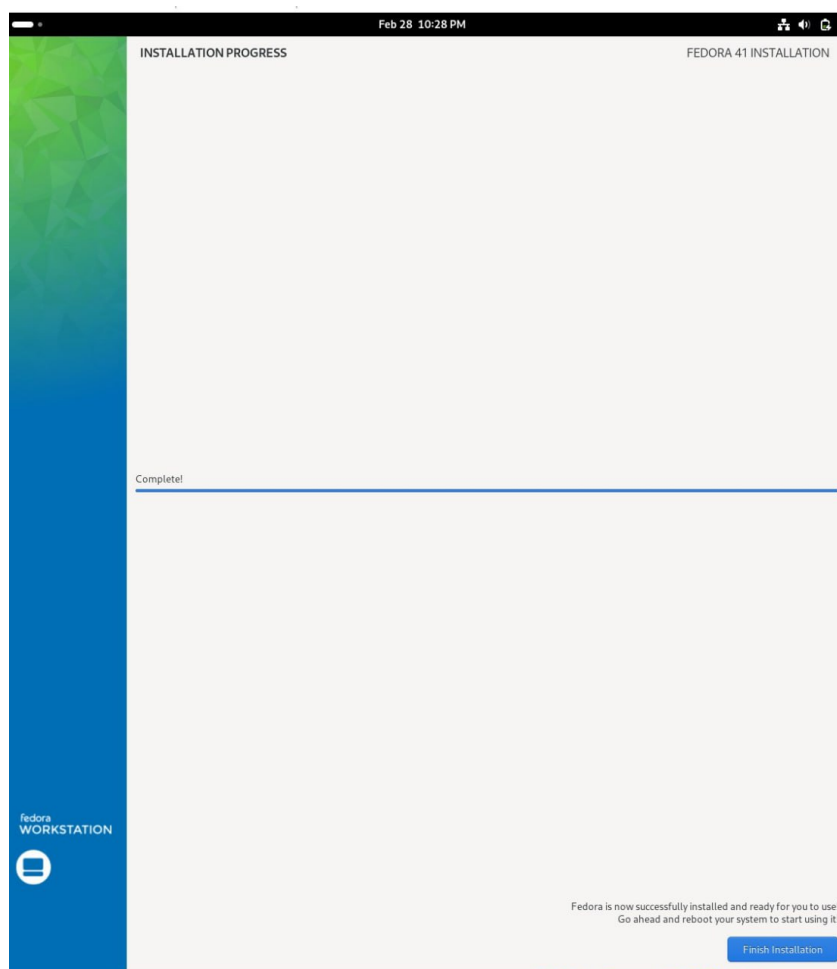


Рис. 3.8: Завершение установки операционной системы

3.3 Работа с операционной системой после установки

Нажимаю Win+Enter для запуска терминала и переключаюсь на роль супер-пользователя(рис. fig. 3.9).

```
ijalloh@vbox:~$ sudo -i
```

Рис. 3.9: Запуск терминала

Обновляю все пакеты (рис. fig. 3.10).

```
root@vbox:~# dnf -y update
```

Рис. 3.10: Обновления

Устанавливаю программы для удобства работы в консоли: `tmux` для открытия нескольких “вкладок” в одном терминале, `mc` в качестве файлового менеджера в терминале (рис. fig. 3.11).

Рис. 3.11: Установка t_{mix} и t_c

Устанавливаю программы для автоматического обновления (рис. fig. 3.12).

```
Complete.
root@vbox:~# dnf install dnf-automatic
Updating and loading repositories:
```

Рис. 3.12: Установка программного обеспечения для автоматического обновления

Запускаю таймер (рис. fig. 3.13).

Рис. 3.13: Запуск таймера

Перемещаюсь в директорию `/etc/selinux`, открываю `md`, ищу нужный файл (рис. 3.14).

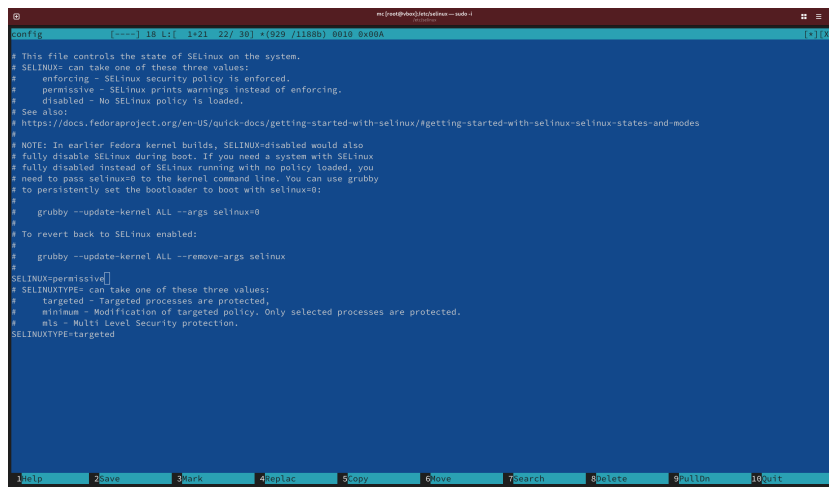
```

root@box:/etc/selinux - ssh - 1
ls -la /etc/selinux
total 28K
drwxr-xr-x. 2 root root 4K Feb 28 05:34 .
drwxr-xr-x. 2 root root 4K Feb 28 05:34 ..
-rw-r--r-- 1 root root 142 Feb 28 05:25 /targeted
-rw-r--r-- 1 root root 1222 Feb 28 05:18 .config_backup
-rw-r--r-- 1 root root 1187 Oct 24 17:57 config
-rw-r--r-- 1 root root 2668 Jul 18 2024 semanage.conf

```

Рис. 3.14: Поиск файла

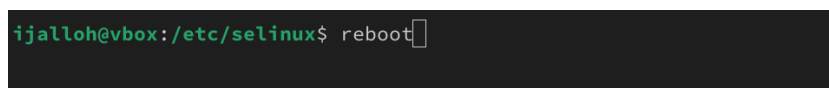
Изменяю открытый файл: SELINUX=enforcing меняю на значение SELINUX=permissive (рис. fig. 3.15).



```
config
# This file controls the state of SELinux on the system.
# SELinux can take one of three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
# See also:
#   https://docs.fedoraproject.org/en-US/quick-docs/getting-started-with-selinux/getting-started-with-selinux-states-and-modes
#
# NOTE: In earlier Fedora kernel builds, SELINUX=disabled would also
# fully disable SELinux during boot. If you need a system with SELinux
# fully disabled instead of SELinux running with no policy loaded, you
# need to pass selinux=0 to the kernel command line. You can use grubby
# to persistently set the bootloader to boot with selinux=0:
#
#   grubby --update-kernel ALL --args selinux=0
#
# To revert back to SELinux enabled:
#
#   grubby --update-kernel ALL --remove-args selinux
#
SELINUX=permissive
# SELINUXTYPE can take one of these three values:
#   targeted - Targeted processes are protected.
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mls - Multi level security protection.
SELINUXTYPE=targeted
```

Рис. 3.15: Изменение файла

Перезагружаю виртуальную машину (рис. fig. 3.16).



```
ijalloh@vbox:/etc/selinux$ reboot
```

Рис. 3.16: Перезагрузка виртуальной машины

Снова вхожу в ОС, снова запускаю терминал, запускаю терминальный мультиплексор (рис. fig. 3.17).



```
ijalloh@vbox:~$ tmux
```

Рис. 3.17: Запуск терминального мультиплексора

Переключаюсь на роль супер-пользователя (рис. fig. 3.18).



```
ijalloh@vbox:/$ sudo -i
[sudo] password for ijalloh:
root@vbox:~#
```

Рис. 3.18: Переключение на роль супер-пользователя

Устанавливаю пакет dkms (рис. fig. ??).

```
root@vbox:~# dnf install dkms
```

Рис. 3.19: Установка пакета dkms

В меню виртуальной машины подключаю образ диска гостевой ОС и примонтирую диск с помощью утилиты mount (рис. fig. 3.20).

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

Рис. 3.20: Примонтирование диска

Устанавливаю драйвера (рис. fig. 3.21).

```
root@vbox:~# /media/VBoxLinuxAdditions.run
Verifying archive integrity... 100% MD5 checksums are OK. All good.
Uncompressing VirtualBox 7.1.0 Guest Additions for Linux 100%
VirtualBox Guest Additions installer
This system appears to have a version of the VirtualBox Guest Additions
already installed. If it is part of the operating system and kept up-to-date,
there is most likely no need to replace it. If it is not up-to-date, you
should get a notification when you start the system. If you wish to replace
```

Рис. 3.21: Установка драйвера

Перезагружаю виртуальную машину (рис. fig. 3.22).

```
root@vbox:~# reboot
```

Рис. 3.22: Перезагрузка виртуальной машины

Перехожу в директорию /etc/X11/xorg.conf.d, открываю tc для удобства, открываю файл 00-keyboard.conf (рис. fig. 3.23).

```
ijalloh@vbox:~$ sudo -i
[sudo] password for ijalloh:
root@vbox:~# cd /etc/X11/xorg.conf.d/
root@vbox:/etc/X11/xorg.conf.d# mc
```

Рис. 3.23: Поиск файла, вход в tc

Редактирую конфигурационный файл (рис. fig. 3.24).

```
00-keyboard.conf [----] 87 L:[ 1+ 9 10/ 12] *(442 / 454b) 0010 0x00A [*][X]
# 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
# update this file.
Section "InputClass"
    Identifier "system-keyboard"
    MatchIsKeyboard "on"
    Option "XkbLayout" "us,ru"
    Option "XkbModel" "pc105"
    Option "XkbVariant" ",winkeys"
    Option "XkbOptions" "grp:alt_shift_toggle, compose:ralt,terminate:ctrl_alt_bksp"
EndSection
```

Рис. 3.24: Редактирование файла

Перезагружаю виртуальную машину (рис. fig. 3.25).

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

Рис. 3.25: Перезагрузка виртуальной машины

3.4 Установка программного обеспечения для создания документации

Запускаю терминал. Запускаю терминальный мультиплексор tmux, переключаюсь на роль супер-пользователя (рис. fig. 3.26).

```
ijalloh@vbox:~$ sudo -i
[sudo] password for ijalloh:
```

Рис. 3.26: Переключение на роль супер-пользователя

Устанавливаю pandoc с помощью утилиты dnf и флага -y, который автоматически на все вопросы системы отвечает “yes” (рис. fig. 3.27).

```
root@vbox:~# dnf -y install pandoc
Updating and loading repositories:
Repositories loaded.
Package Arch Version Repository Size
Installing:
pandoc x86_64 3.1.11.1-32.fc41 fedora 185.0 MiB
Installing dependencies:
pandoc-common noarch 3.1.11.1-31.fc41 fedora 1.9 MiB

Transaction Summary:
Installing: 2 packages

Total size of inbound packages is 27 MiB. Need to download 27 MiB.
After this operation, 187 MiB extra will be used (install 187 MiB, remove 0 B)
```

Рис. 3.27: Установка pandoc

Устанавливаю необходимые расширения для pandoc (рис. fig. 3.28).

```
root@vbox:~# pip install pandoc-fignos pandoc-eqnos pandoc-tablenos pandoc-secnos --user
```

Рис. 3.28: Установка расширения pandoc

Устанавливаю дистрибутив texlive (рис. fig. 3.29).

```
root@vbox:~# dnf -y install texlive texlive-*
```

Рис. 3.29: Установка texlive

4 Выводы

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

5 Выполнение дополнительного задания

Ввожу в терминале команду `dmesg`, чтобы проанализировать последовательность загрузки системы (рис. fig. 5.1).

```
ijalloh@vbox:~$ sudo dmesg
[ 0.000000] Linux version 6.13.4-200.fc41.x86_64 (mockbuild@1eec6c3659654d339658e9322f9b7a5a) (gcc (GCC) 14.2.1 20250110 (Red Hat 14.2.1-7), GNU ld version 2.43.1-5.fc41) #1 SMP PREEMPT_DYNAMIC Sat Feb 22 16:09:10 UTC 2025
[ 0.000000] Command line: BOOT_IMAGE=(hd0,gpt2)/vmlinuz-6.13.4-200.fc41.x86_64 root=UUID=bdba74b3-304a-432f-b6ba-7ffed7265137 ro rootflags=subvol=root rhgb quiet
[ 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 0x0000000001000000-0x000000000dfffff] usable
[ 0.000000] BIOS-e820: [mem 0x00000000dffff0000-0x00000000dfffffff] 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] BIOS-e820: [mem 0x00000000100000000-0x0000000011fffff] 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] DMI: Memory slots populated: 0/0
[ 0.000000] Hypervisor detected: KVM
[ 0.000000] kvm-clock: Using msrs 4b564d01 and 4b564d00
[ 0.000004] kvm-clock: using sched offset of 13468702786523 cycles
[ 0.000008] clocksource: kvm-clock: mask: 0xffffffffffffffff max_cycles: 0x1cd42e4dffb, max_idle_ns: 881590591483 ns
[ 0.000014] tsc: Detected 1497.604 Mhz processor
```

Рис. 5.1: Анализ последовательности загрузки системы

С помощью поиска, осуществляемого командой `'dmesg | grep -i'`, ищу версию ядра Linux: 6.1.10-200.fc37.x86_64 (рис. fig. 5.2).

```
ijalloh@vbox:~$ sudo dmesg | grep -i "Linux Version"
[ 0.000000] Linux version 6.13.4-200.fc41.x86_64 (mockbuild@1eec6c3659654d339658e9322f9b7a5a) (gcc (GCC) 14.2.1 20250110 (Red Hat 14.2.1-7), GNU ld version 2.43.1-5.fc41) #1 SMP PREEMPT_DYNAMIC Sat Feb 22 16:09:10 UTC 2025
```

Рис. 5.2: Поиск версии ядра

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. fig. 5.3).

```
ijalloh@vbox:~$ sudo dmesg | grep -i "processor"
[ 0.000014] tsc: Detected 1497.604 MHz processor
[ 0.417355] smpboot: Total of 4 processors activated (11980.83 BogoMIPS)
[ 0.440241] ACPI: Added _OSI(Processor Device)
[ 0.440243] ACPI: Added _OSI(Processor Aggregator Device)
```

Рис. 5.3: Поиск частоты процессора

Аналогично ищу модель процессора (рис. fig. 5.4).

```
ijalloh@vbox:~$ sudo dmesg | grep -i "CPU0"
[ 0.401326] smpboot: CPU0: Intel(R) Core(TM) i7-1065G7 CPU @ 1.30GHz (family: 0x6, model: 0x7e, stepping: 0x5)
```

Рис. 5.4: Поиск модели процессора

Объем доступной оперативной памяти ищу аналогично поиску частоты процессора, т. к. возникла та же проблема, что и там (рис. fig. 5.5).

```
ijalloh@vbox:~$ sudo dmesg | grep -i "memory"
[ 0.000000] DMI: Memory slots populated: 0/0
[ 0.003229] ACPI: Reserving FACP table memory at [mem 0xdfff00f0-0xdfff01e3]
[ 0.003230] ACPI: Reserving DSDT table memory at [mem 0xdfff0620-0xdfff2972]
[ 0.003231] ACPI: Reserving FACS table memory at [mem 0xdfff0200-0xdfff023f]
[ 0.003232] ACPI: Reserving FACS table memory at [mem 0xdfff0200-0xdfff023f]
[ 0.003233] ACPI: Reserving APIC table memory at [mem 0xdfff0240-0xdfff02ab]
[ 0.003233] ACPI: Reserving SSDT table memory at [mem 0xdfff02b0-0xdfff061b]
[ 0.005795] Early memory node ranges
[ 0.015115] PM: hibernation: Registered nosave memory: [mem 0x00000000-0x00000fff]
[ 0.015118] PM: hibernation: Registered nosave memory: [mem 0x0009f000-0x0009ffff]
[ 0.015119] PM: hibernation: Registered nosave memory: [mem 0x000a0000-0x000effff]
[ 0.015119] PM: hibernation: Registered nosave memory: [mem 0x000f0000-0x000fffff]
[ 0.015120] PM: hibernation: Registered nosave memory: [mem 0xdfff0000-0xdfffffff]
[ 0.015121] PM: hibernation: Registered nosave memory: [mem 0xe0000000-0xfebfffff]
[ 0.015122] PM: hibernation: Registered nosave memory: [mem 0xf0000000-0xf000ffff]
```

Рис. 5.5: Поиск объема доступной оперативной памяти

Нахожу тип обнаруженного гипервизора (рис. fig. 5.6).

```
ijalloh@vbox:~$ sudo dmesg | grep -i "Hypervisor"
[ 0.000000] Hypervisor detected: KVM
[ 0.263577] SRBDS: Unknown: Dependent on hypervisor status
```

Рис. 5.6: Поиск типа обнаруженного гипервизора

Последовательность монтирования файловых систем можно посмотреть, введя в поиск по результату dmesg слово mount (рис. fig. 5.7).

```
ijalloh@vbox:~$ sudo dmesg | grep -i "mount"
[ 0.301056] Mount-cache hash table entries: 8192 (order: 4, 65536 bytes, linear)
[ 0.301066] Mountpoint-cache hash table entries: 8192 (order: 4, 65536 bytes, linear)
[ 3.821677] BTRFS: device label fedora devid 1 transid 494 /dev/sda3 (8:3) scanned by mount (453)
[ 3.829386] BTRFS info (device sda3): first mount of filesystem bdba74b3-304a-432f-b6ba-7ffed7265137
[ 6.692759] systemd[1]: run-credentials-systemd\x2djournal.service.mount: Deactivated successfully.
[ 6.706902] systemd[1]: Set up automount proc-sys-fs-binfmt_misc.automount - Arbitrary Executable File Formats File System Automount Point.
[ 6.725839] systemd[1]: Listening on systemd-mountfsd.socket - DDI File System Mounter Socket.
[ 6.748344] systemd[1]: Mounting dev-hugepages.mount - Huge Pages File System...
[ 6.756638] systemd[1]: Mounting dev-mqueue.mount - POSIX Message Queue File System...
[ 6.760263] systemd[1]: Mounting sys-kernel-debug.mount - Kernel Debug File System...
[ 6.764210] systemd[1]: Mounting sys-kernel-tracing.mount - Kernel Trace File System...
[ 6.874804] systemd[1]: Starting systemd-remount-fs.service - Remount Root and Kernel File Systems...
[ 6.906630] systemd[1]: Mounted dev-hugepages.mount - Huge Pages File System.
[ 6.908418] systemd[1]: Mounted dev-mqueue.mount - POSIX Message Queue File System.
[ 6.908924] systemd[1]: Mounted sys-kernel-debug.mount - Kernel Debug File System
```

Рис. 5.7: Последовательность монтирования файловых систем

6 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 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 объёма 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.

Список литературы

::: {#refs} 1. Dash P. Getting started with oracle vm virtualbox. Packt Publishing Ltd, 2013. 86 p. 2. Colvin H. Virtualbox: An ultimate guide book on virtualization with virtualbox. CreateSpace Independent Publishing Platform, 2015. 70 p. 3. van Vugt S. Red hat rhcsa/rhce 7 cert guide : Red hat enterprise linux 7 (ex200 and ex300). Pearson IT Certification, 2016. 1008 p. 4. Робачевский А., Немнюгин С., Стесик О. Операционная система unix. 2-е изд. Санкт-Петербург: БХВ-Петербург, 2010. 656 p. 5. Немец Э. et al. Unix и Linux: руководство системного администратора. 4-е изд. Вильямс, 2014. 1312 p. 6. Колисниченко Д.Н. Самоучитель системного администратора Linux. СПб.: БХВ-Петербург, 2011. 544 p. 7. Robbins A. Bash pocket reference. O'Reilly Media, 2016. 156 p.