

Лабораторная работа №1

Отчёт

Ермишина Мария Кирилловна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Домашнее задание	18
4	Выводы	20
5	Контрольные вопросы	21
	Список литературы	22

Список иллюстраций

2.1	Интерфейс VirtualBox	6
2.2	Имя и образ VM	7
2.3	Настройки VM	7
2.4	Создание VM	8
2.5	Установка Liveinst	9
2.6	Экран с проведёнными настройками	10
2.7	Обновление	11
2.8	Повышение комфорта работы	11
2.9	Автоматическое обновление	12
2.10	Отключение SELinux	13
2.11	Запуск tmux	14
2.12	Создание и редак. конфиг. файла	15
2.13	Редакт. 2 конфиг. файла	15
2.14	Смена имени хоста	16
2.15	Скачивание pandoc	17
2.16	Скачивание Texlive	17
3.1	Выполнение Д/З 1в	18
3.2	Выполнение Д/З 2	19

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. [1]

2 Выполнение лабораторной работы

1. Создание виртуальной машины. Для начала запускаем программу VirtualBox. В открывшемся окне мы видим интерфейс приложения. Выбираем кнопку “Создать”. (рис. 2.1)

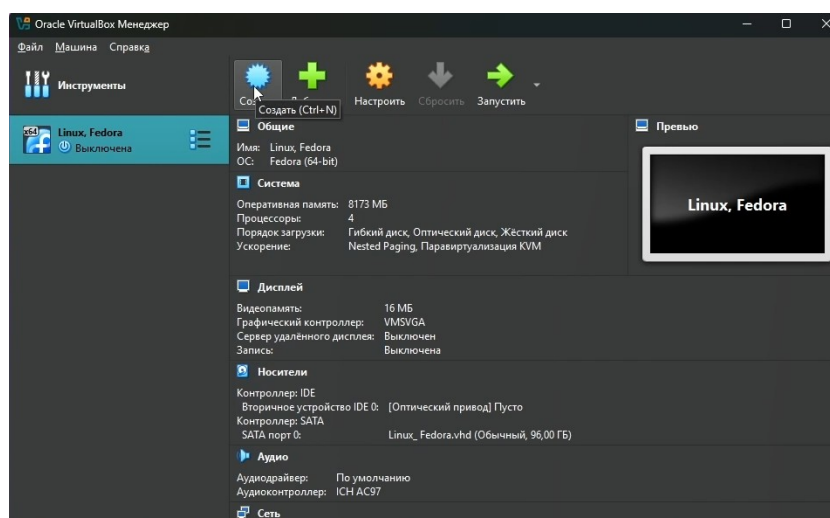


Рис. 2.1: Интерфейс VirtualBox

В новом окне нам необходимо указать имя для виртуальной машины, а также выбрать образ ISO. (рис. 2.2)

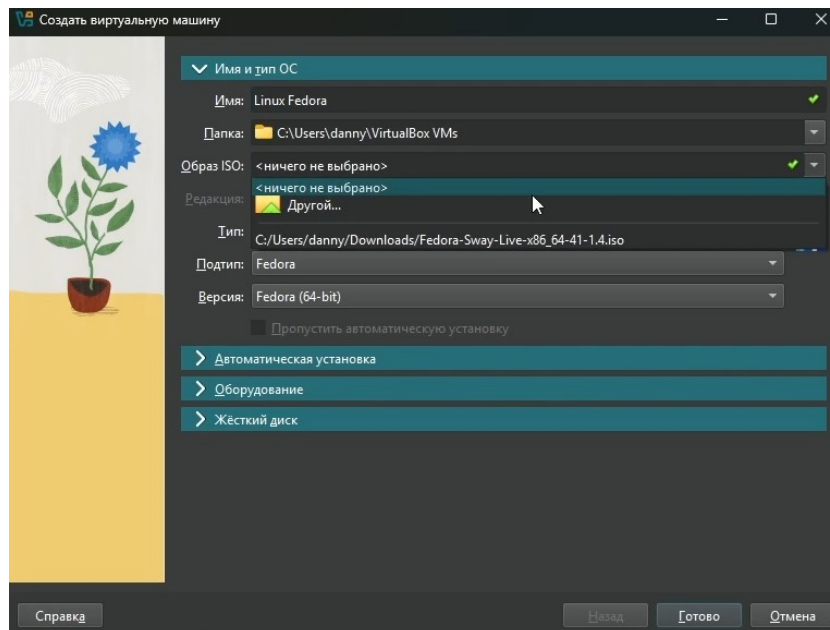


Рис. 2.2: Имя и образ VM

Далее необходимо указать размер основной памяти, а также процессоры, необходимые для работы виртуальной машины. (рис. 2.3).

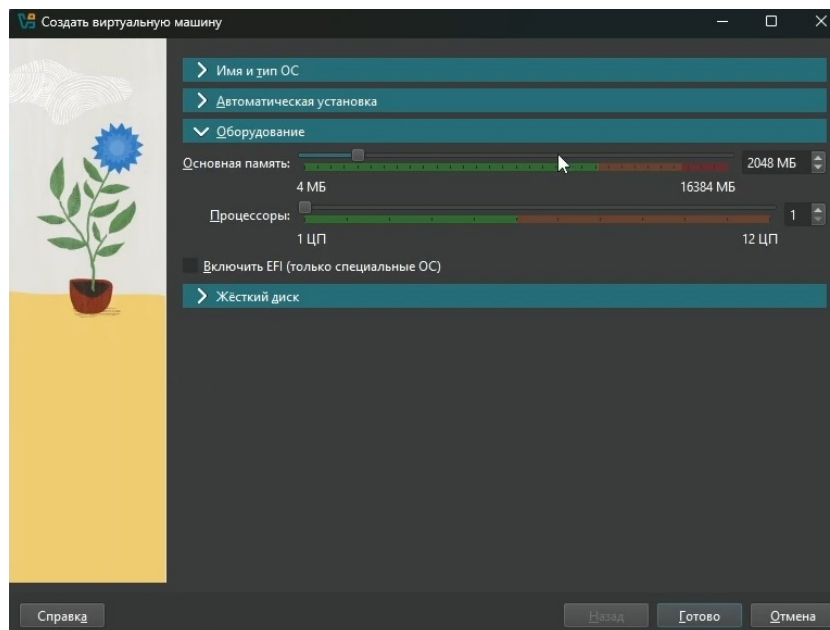


Рис. 2.3: Настройки VM

Создаём виртуальный жёсткий диск (не меньше 80 ГБ) и создаём машину. (рис.

2.4)

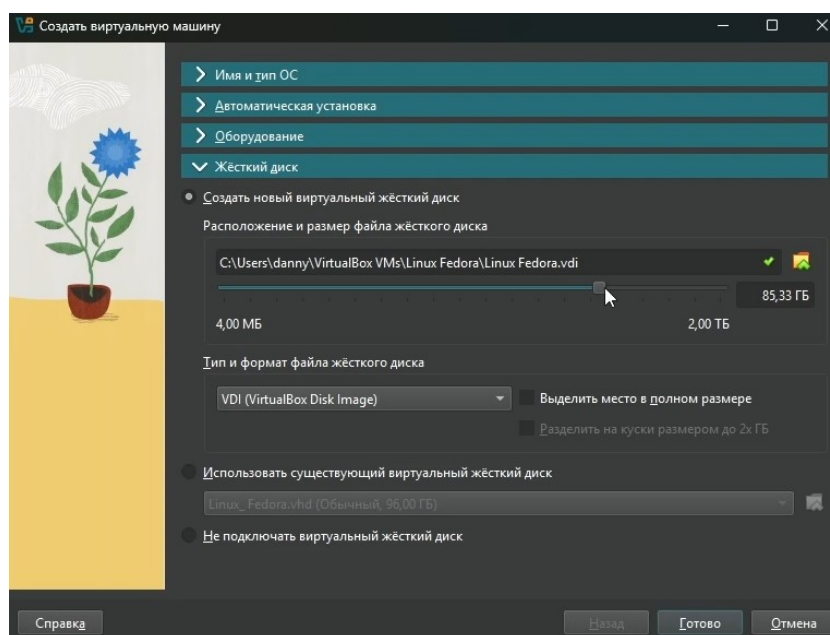


Рис. 2.4: Создание VM

2. Установка операционной системы.
3. Запуск приложения для установки системы. После запуска виртуальной машины необходимо нажать сочетание клавиш Win + D, а после вводим Liveinst и начинаем загрузку. (рис. 2.5)

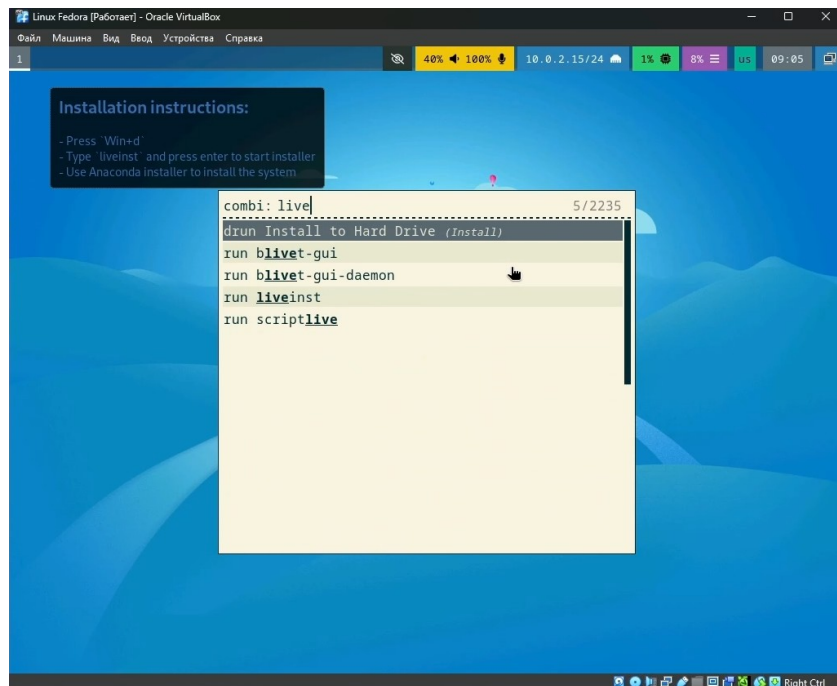


Рис. 2.5: Установка Liveinst

2. Переходим к настройкам до установки: (рис. 2.6)

- В появившемся окне необходимо выбрать язык интерфейса и (при необходимости) скорректировать часовой пояс, раскладку.
- Место установки ОС оставляем без изменения.
- Устанавливаем имена и пароли для пользователя, пользователя root и сетевое имя компьютера.

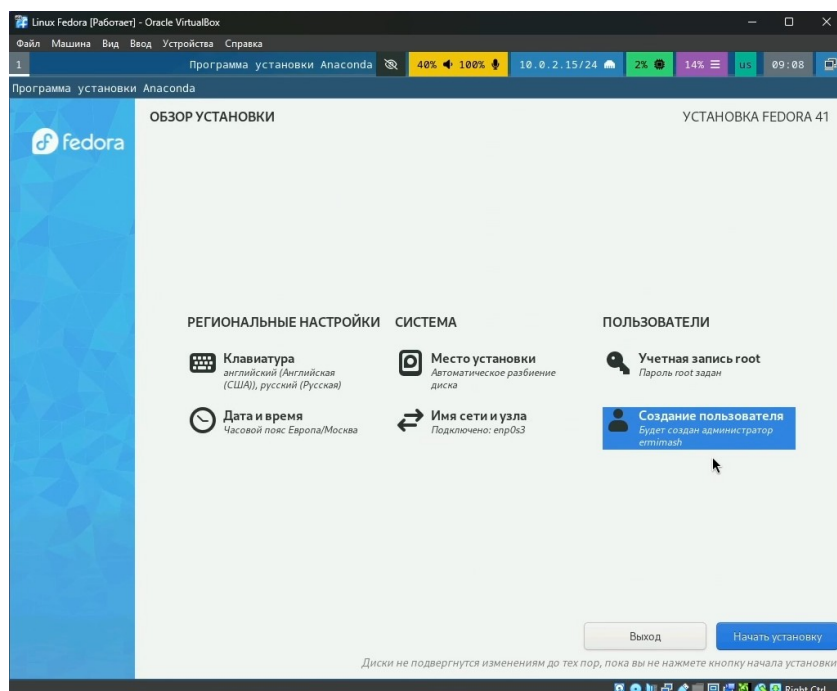


Рис. 2.6: Экран с проведёнными настройками

После корректно перезапускаем виртуальную машину. Если оптический диск не отключился автоматически - отключаем его самостоятельно.

3. После установки. Входим в ОС под записью, созданной при установке. Запускаем терминал с помощью комбинации Win + Enter и переключаемся на роль супер-пользователя с помощью команды `sudo -i`.
4. Обновления (рис. 2.7) Устанавливаем средства разработки с помощью следующей команды:

- `sudo dnf -y group install development-tools`

Обновляем все пакеты: - `sudo dnf -y update`

```

[110/143] Installing glibc-devel-0.2.40-21.fc41.x86_64 100% | 10.1 MiB/s | 2.3 MiB | 00m00s
[111/143] Installing gcc-0.14.2.1-7.fc41.x86_64 100% | 112.9 MiB/s | 103.4 MiB | 00m01s
[112/143] Installing cmake-filesystem-0.3.30-8-1.fc41.x86_64 100% | 384.3 KiB/s | 7.3 KiB | 00m00s
[113/143] Installing zlib-ng-compat-devel-0.2.2.3-2.fc41.x86_64 100% | 11.8 MiB/s | 100.5 KiB | 00m00s
[114/143] Installing elfutils-libelf-devel-0.0.192-9.fc41.x86_64 100% | 1.5 MiB/s | 55.0 KiB | 00m00s
[115/143] Installing xz-devel-1.5.6-2.2.fc41.x86_64 100% | 14.9 MiB/s | 259.4 KiB | 00m00s
[116/143] Installing elfutils-devel-0.0.192-9.fc41.x86_64 100% | 9.5 MiB/s | 184.9 KiB | 00m00s
[117/143] Installing systemtap-devel-0.5.2-1.fc41.x86_64 100% | 41.0 MiB/s | 9.2 MiB | 00m00s
[118/143] Installing systemtap-0.5.2-1.fc41.x86_64 100% | 1.8 KiB/s | 124.0 B | 00m00s
[119/143] Installing kernel-devel-0.6.13.5-200.fc41.x86_64 100% | 1.3 MiB/s | 81.4 MiB | 01m01s
[120/143] Installing dosygen-2:1.12.0-2.fc41.x86_64 100% | 256.3 MiB/s | 19.2 MiB | 00m00s
[121/143] Installing patchutils-0.0.4-2.14.fc41.x86_64 100% | 10.1 MiB/s | 216.4 KiB | 00m00s
[122/143] Installing subversion-0:1.14.5-1.fc41.x86_64 100% | 56.9 MiB/s | 5.1 MiB | 00m00s
[123/143] Installing gettext-0.22.5-6.fc41.x86_64 100% | 55.7 MiB/s | 5.2 MiB | 00m00s
[124/143] Installing apr-util-ldb-0:1.6.3-21.fc41.x86_64 100% | 2.6 MiB/s | 16.1 KiB | 00m00s
[125/143] Installing apr-util-openssl-0:1.6.3-21.fc41.x86_64 100% | 3.0 MiB/s | 24.4 KiB | 00m00s
[126/143] Installing perl-MOD Perl-0:1.17.514.fc41.x86_64 100% | 4.7 MiB/s | 33.7 KiB | 00m00s
[127/143] Installing elfutils-debuginfo-client-devel-0.0.192-9.fc41.x86_64 100% | 985.9 KiB/s | 12.8 KiB | 00m00s
[128/143] Upgrading elfutils-0.0.192-9.fc41.x86_64 100% | 30.8 MiB/s | 2.7 MiB | 00m00s
[129/143] Installing diffstat-0:1.66-2.fc41.x86_64 100% | 9.5 MiB/s | 78.0 KiB | 00m00s
[130/143] Erasing elfutils-0.0.191-8.fc41.x86_64 100% | 4.5 KiB/s | 64.0 B | 00m00s
[131/143] Erasing elfutils-debuginfo-client-0.0.191-8.fc41.x86_64 100% | 1.8 KiB/s | 13.0 B | 00m00s
[132/143] Erasing elfutils-libs-0.0.191-8.fc41.x86_64 100% | 1.7 KiB/s | 12.0 B | 00m00s
[133/143] Erasing cpp-0:14.2.1-3.fc41.x86_64 100% | 3.5 KiB/s | 36.0 B | 00m00s
[134/143] Erasing elfutils-libelf-0.0.191-8.fc41.x86_64 100% | 2.3 KiB/s | 14.0 B | 00m00s
[135/143] Erasing libxcrypt-0:4.4.36-7.fc41.x86_64 100% | 1.3 KiB/s | 18.0 B | 00m00s
[136/143] Erasing libgomp-0:14.2.1-3.fc41.x86_64 100% | 1.3 KiB/s | 8.0 B | 00m00s
[137/143] Erasing libstdc++-0:13.6-2.fc41.x86_64 100% | 1.1 KiB/s | 8.0 B | 00m00s
[138/143] Erasing zlib-ng-compat-0:2.1.7-3.fc41.x86_64 100% | 625.0 B/s | 5.0 B | 00m00s
[139/143] Erasing glibc-0:2.40-3.fc41.x86_64 100% | 8.4 KiB/s | 103.0 B | 00m00s
[140/143] Erasing glibc-all-langpacks-0:2.40-3.fc41.x86_64 100% | 3.3 KiB/s | 41.0 B | 00m00s
[141/143] Erasing glibc-gconv-extra-0:2.40-3.fc41.x86_64 100% | 40.7 KiB/s | 624.0 B | 00m00s
>>> Running post-uninstall scriptlet: glibc-gconv-extra-0:2.40-3.fc41.x86_64warning: posix.fork(): .fork(), .exec(), .wait() and .redirect2null() are deprecated, use rpm.spawn() or rpm.execute() instead
warning: posix.wait(): .fork(), .exec(), .wait() and .redirect2null() are deprecated, use rpm.spawn() or rpm.execute() instead
[141/143] Erasing glibc-gconv-extra-0:2.40-3.fc41.x86_64 100% | 11.5 KiB/s | 624.0 B | 00m00s
[142/143] Erasing glibc-common-0:2.40-3.fc41.x86_64 100% | 7.3 KiB/s | 52.0 B | 00m00s
[143/143] Erasing libgcc-0:14.2.1-3.fc41.x86_64 100% | 0.0 B/s | 11.0 B | 00m00s
>>> Running post-uninstall scriptlet: libgcc-0:14.2.1-3.fc41.x86_64warning: posix.fork(): .fork(), .exec(), .wait() and .redirect2null() are deprecated, use rpm.spawn() or rpm.execute() instead
warning: posix.wait(): .fork(), .exec(), .wait() and .redirect2null() are deprecated, use rpm.spawn() or rpm.execute() instead
[143/143] Erasing libgcc-0:14.2.1-3.fc41.x86_64 100% | 6.0 B/s | 11.0 B | 00m00s
Complete!
[root@terminash ~]# sudo dnf -y update

```

Рис. 2.7: Обновление

2. Повышение комфорта работы (рис. 2.8) Программы для удобства работы в консоли:

- `sudo dnf -y install tmux mc`

Другой вариант консоли: - `sudo dnf -y install kitty`

```

[root@terminash ~]# sudo dnf -y install tmux mc
Обновление и загрузка репозитория:
Репозиторий загружен.
Пакет "tmux-3.5a-2.fc41.x86_64" уже установлен.

Пакет Арх. Версия Репозиторий Размер
Установка: mc x86_64 1:4.8.32-1.fc41 updates 7.2 MiB
Установка зависимостей: gpm-libs x86_64 1:20.7-48.fc41 fedora 27.7 MiB
Итого: 2 пакетов
Общий размер входных пакетов составляет 2 MiB. Необходимо загрузить 2 MiB.
После этой операции будут использоваться дополнительные 7 MiB (установка 7 MiB, удаление 0 B).
[1/2] gpm-libs-0:1.20.7-48.fc41.x86_64 100% | 182.0 KiB/s | 20.2 KiB | 00m00s
[2/2] mc-1:4.8.32-1.fc41.x86_64 100% | 8.5 MiB/s | 1.9 MiB | 00m00s
-----
[2/2] Total 100% | 2.0 MiB/s | 2.0 MiB | 00m01s
Выполнение транзакции
[1/4] Проверить файлы пакета 100% | 250.0 B/s | 2.0 B | 00m00s
[2/4] Подготовить транзакцию 100% | 12.0 B/s | 2.0 B | 00m00s
[3/4] Установка gpm-libs-0:1.20.7-48.fc41.x86_64 100% | 1.5 MiB/s | 28.5 KiB | 00m00s
[4/4] Установка mc-1:4.8.32-1.fc41.x86_64 100% | 11.5 MiB/s | 7.2 MiB | 00m01s
Завершено!
[root@terminash ~]#

```

Рис. 2.8: Повышение комфорта работы

3. Автоматическое обновление (рис. 2.9) Установка программного обеспечения:

- `sudo dnf -y install dnf-automatic`

Запустите таймер: - `sudo systemctl enable --now dnf-automatic.timer`

```
[root@ermimash ~]# sudo dnf -y install dnf-automatic
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет
Репозиторий
Арх.
Версия
Разм
ер
Установка:
dnf5-plugin-automatic
x86_64
5.2.10.0-2.fc41
178.6 K
с41
updates
iB

Сводка транзакции:
Установка: 1 пакета

Общий размер входящих пакетов составляет 141 KiB. Необходимо загрузить 141 KiB.
После этой операции будут использоваться дополнительные 179 KiB (установка 179 KiB,
удаление 0 B).
[1/1] dnf5-plugin-automatic-0:5.2.10.0-2.fc41.x86_64
100% | 994.7 KiB/s | 141.3 KiB | 00m00s
0s
-----
--
[1/1] Total
100% | 244.8 KiB/s | 141.3 KiB | 00m00s
1s
Выполнение транзакции
[1/3] Проверить файлы пакета
100% | 250.0 B/s | 1.0 B | 00m00s
[2/3] Подготовить транзакцию
100% | 6.0 B/s | 1.0 B | 00m00s
[3/3] Установка dnf5-plugin-automatic-0:5.2.10.0-2.fc41.x86_64
100% | 255.9 KiB/s | 180.7 KiB | 00m01s
Завершено!
[root@ermimash ~]# sudo systemctl enable --now dnf-automatic.timer
Created symlink '/etc/systemd/system/timers.target.wants/dnf5-automatic.timer' -> '/usr/lib/systemd/system/dnf5-automatic.timer'.
```

Рис. 2.9: Автоматическое обновление

4. Отключение SELinux (рис. 2.10) В данном курсе мы не будем рассматривать работу с системой безопасности SELinux. Поэтому отключим его. С помощью nano в файле `/etc/selinux/config` замените значение “SELINUX=enforcing” на “SELINUX=permissive”.

После перезапускаем виртуальную машину: - `sudo systemctl reboot`

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these 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/#get
#
# 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. 2.10: Отключение SELinux

4. Настройка раскладки клавиатуры. Запускаем терминал с помощью Win + Enter. Запускаем мультиплексор tmux. (рис. 2.11)



Рис. 2.11: Запуск tmux

Создайте конфигурационный файл: (рис. 2.12) - `mkdir -p ~/.config/sway - touch ~/.config/sway/config.d/95-system-keyboard-config.conf`

Далее редактируем созданный нами файл: (рис. 2.12) - `exec_always /usr/libexec/sway-systemd/locale1-xkb-config --oneshot`

```
ermimash@ermimash:~$ mkdir -p ~/.config/sway/config.d
ermimash@ermimash:~$ touch ~/.config/sway/config.d/95-system-keyboard-config.conf
ermimash@ermimash:~$ exec_always /usr/libexec/sway-systemd/locale1-xkb-config --ones
hot
-bash: exec_always: команда не найдена
ermimash@ermimash:~$ nano ~/.config/sway/config.d/95-system-keyboard-config.conf
```

Рис. 2.12: Создание и редакт. конфиг. файла

Переключаемся на роль супер-пользователя: - `sudo -i`

Редактируем второй конфигурационный файл, заменяя текст в нём на указанный ниже текст: (рис. 2.13) Section “InputClass” Identifier “system-keyboard” MatchIsKeyboard “on” Option “XkbLayout” “us,ru” Option “XkbVariant” “,winkeys” Option “XkbOptions” “grp:rctrl_toggle,compose:ralt,terminate:ctrl_alt_bksp” EndSection

```
Section "InputClass"
    Identifier "system-keyboard"
    MatchIsKeyboard "on"
    Option "XkbLayout" "us,ru"
    Option "XkbModel" "pc105"
    Option "XkbVariant" ",,"
    Option "XkbOptions" "grp:alt_shift_toggle"
EndSection
```

Рис. 2.13: Редакт. 2 конфиг. файла

Перезагружаем виртуальную машину.

5. Установка имени пользователя и названия хоста. (рис. 2.14) Если при установке виртуальной машины вы задали имя пользователя или имя хоста, не удовлетворяющее соглашению об именовании, то вам необходимо исправить это.

В терминале запускаем терминальный мультиплексор с помощью команды `tmux` и переключаемся на супер-пользователя. Установите имя хоста с помощью следующей команды: - `hostnamectl set-hostname username`

Проверяем, что имя хоста установлено верно: - `hostnamectl`

```
[ermimash@ermimash ~]$ hostnamectl set-hostname ermimash
[ermimash@ermimash ~]$ hostnamectl
  Static hostname: ermimash
        Icon name: computer-vm
        Chassis: vm
        Machine ID: 67a4cde8ffb04f4abe296bc411f0d2cc
        Boot ID: 03a0cc204a9c474297fd9c5e1f15f1bb
  Virtualization: oracle
  Operating System: Fedora Linux 41 (Sway)
        CPE OS Name: cpe:/o:fedoraproject:fedora:41
        OS Support End: Mon 2025-12-15
  OS Support Remaining: 9month 1w 4d
        Kernel: Linux 6.13.5-200.fc41.x86_64
  Architecture: x86-64
  Hardware Vendor: innotek GmbH
  Hardware Model: VirtualBox
  Firmware Version: VirtualBox
  Firmware Date: Fri 2006-12-01
  Firmware Age: 18y 3month 2d
[ermimash@ermimash ~]$
```

Рис. 2.14: Смена имени хоста

6. Установка программного обеспечения для создания документации. В терминале запускаем терминальный мультиплексор с помощью команды `tmux` и переключаемся на супер-пользователя.

7. Работа с языком разметки Markdown Для работы с языком разметки Markdown. Устанавливаем его:

- `sudo dnf -y install pandoc`

Для работы с перекрёстными ссылками мы используем пакет `pandoc-crossref`. Скачиваем необходимую версию `pandoc-crossref` (<https://github.com/lierdakil/pandoc-crossref/releases>).

Распакуйте архивы и помещаем их в каталог `/usr/local/bin`. (рис. 2.15)


```

[ermimash@ermimash ~]$ cd Загрузки
[ermimash@ermimash Загрузки]$ ls
pandoc-crossref-Linux.tar.xz  pandoc-crossref-Linux-X64.tar.xz
[ermimash@ermimash Загрузки]$ rm pandoc-crossref-Linux-X64.tar.xz
[ermimash@ermimash Загрузки]$ taz -xvf pandoc-crossref-Linux.tar.xz
bash: taz: команда не найдена
[ermimash@ermimash Загрузки]$ ls
pandoc-crossref-Linux.tar.xz
[ermimash@ermimash Загрузки]$ tar -xvf pandoc-crossref-Linux.tar.xz
pandoc-crossref
pandoc-crossref.1
[ermimash@ermimash Загрузки]$ ls
pandoc-crossref  pandoc-crossref.1  pandoc-crossref-Linux.tar.xz
[ermimash@ermimash Загрузки]$ sudo mv pandoc-crossref /usr/local/bin
[sudo] пароль для ermimash:
[ermimash@ermimash Загрузки]$ cd
[ermimash@ermimash ~]$

```

Рис. 2.15: Скачивание pandoc

2. Texlive (рис. 2.16) Устанавливаем дистрибутив с помощью следующей команды:

- `sudo dnf -y install texlive-scheme-full`

```

[4582/4603] Установка texlive-csla 100% | 3.6 MiB/s | 89.6 KiB | 00m00s
[4583/4603] Установка texlive-coll 100% | 24.2 KiB/s | 124.0 B | 00m00s
[4584/4603] Установка texlive-emoj 100% | 91.6 MiB/s | 1.6 MiB | 00m00s
[4585/4603] Установка texlive-coll 100% | 15.1 KiB/s | 124.0 B | 00m00s
[4586/4603] Установка texlive-pgf- 100% | 12.2 MiB/s | 362.2 KiB | 00m00s
[4587/4603] Установка texlive-coll 100% | 17.3 KiB/s | 124.0 B | 00m00s
[4588/4603] Установка texlive-coll 100% | 13.5 KiB/s | 124.0 B | 00m00s
[4589/4603] Установка texlive-sche 100% | 20.2 KiB/s | 124.0 B | 00m00s
[4590/4603] Установка gstreamer1-p 100% | 7.0 MiB/s | 187.4 KiB | 00m00s
[4591/4603] Установка evince-0:46. 100% | 5.6 MiB/s | 10.5 MiB | 00m02s
[4592/4603] Установка evince-djvu- 100% | 3.7 MiB/s | 63.8 KiB | 00m00s
[4593/4603] Установка perl-List-Mo 100% | 7.3 MiB/s | 165.1 KiB | 00m00s
[4594/4603] Установка perl-Lexical 100% | 2.0 MiB/s | 55.9 KiB | 00m00s
[4595/4603] Установка rubygem-rdoc 100% | 14.1 MiB/s | 1.8 MiB | 00m00s
[4596/4603] Установка perl-GD-Barc 100% | 79.4 MiB/s | 10.6 MiB | 00m00s
[4597/4603] Установка perl-IO-Comp 100% | 1.2 MiB/s | 39.0 KiB | 00m00s
[4598/4603] Установка rubygem-bigr 100% | 8.9 MiB/s | 136.6 KiB | 00m00s
[4599/4603] Установка rubygem-bund 100% | 9.2 MiB/s | 1.5 MiB | 00m00s
[4600/4603] Установка qt5-qttransl 100% | 81.3 MiB/s | 14.8 MiB | 00m00s
[4601/4603] Установка iio-sensor-p 100% | 2.7 MiB/s | 163.5 KiB | 00m00s
[4602/4603] Установка perl-Class-X 100% | 5.2 MiB/s | 101.6 KiB | 00m00s
[4603/4603] Установка perl-PerlIO- 100% | 109.0 B/s | 44.2 KiB | 06m54s
Завершено!

```

Рис. 2.16: Скачивание Texlive

3 Домашнее задание

В окне терминала проанализируем последовательность загрузки системы, выполнив команду `dmesg`. Используем следующую команду: `- dmesg | less`

А после используем команду для поиска: `- dmesg | grep -i "то, что ищем"`

Получаем следующую информацию.

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

```
[root@ermimash ~]# dmesg | grep -i "Linux version"
[    0.000000] Linux version 6.13.5-200.fc41.x86_64 (mockbuild@be03da54f8364b379359fe70f52a8f23) (gcc (GCC) 14.2.1 20250110 (Red Hat 14.2.1-7), GNU ld version 2.43.1-5.fc41) #1 SMP PREEMPT_DYNAMIC Thu Feb 27 15:07:31 UTC 2025
[root@ermimash ~]# dmesg | grep -i "Detected Mhz processor"
[root@ermimash ~]# dmesg | grep -i "Detected Mhz processor"
[root@ermimash ~]# dmesg | grep -i "Mhz processor"
[    0.000008] tsc: Detected 2591.998 Mhz processor
[root@ermimash ~]# dmesg | grep -i "CPU0"
[    0.221162] smpboot: CPU0: 11th Gen Intel(R) Core(TM) i5-11400F @ 2.60GHz (family: 0x6, model: 0xa7, stepping: 0x1)
```

Рис. 3.1: Выполнение Д/З 1в

```

[ 0.230522] Memory: 8254456K/8567352K available (22528K kernel code, 4456K r/w data
, 16892K rodata, 4924K init, 4632K bss, 305852K reserved, 0K cma-reserved)
[ 0.232190] x86/mm: Memory block size: 128MB
[ 0.693261] Freeing initrd memory: 26320K
[ 0.731785] Non-volatile memory driver v1.3
[ 1.083560] Freeing unused decrypted memory: 2028K
[ 1.084262] Freeing unused kernel image (initmem) memory: 4924K
[ 1.085391] Freeing unused kernel image (rodata/data gap) memory: 1540K
[ 4.041356] systemd[1]: Listening on systemd-oomd.socket - Userspace Out-Of-Memory
(OOM) Killer Socket.
[root@ermimash ~]# dmesg | grep -i "Hypervisor detected"
[ 0.000000] Hypervisor detected: KVM
[root@ermimash ~]# dmesg | grep -i "filesystem"
[ 2.570902] BTRFS info (device sda3): first mount of filesystem b047707a-f7d8-40a
1-9297-ed78d03fbd3f
[ 5.274153] EXT4-fs (sda2): mounted filesystem b9c46105-8653-490c-8b15-fe8c8fd448
6f r/w with ordered data mode. Quota mode: none.
[root@ermimash ~]#

```

Рис. 3.2: Выполнение Д/З 2

4 Выводы

В ходе работы приобретены практических навыков установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов.

5 Контрольные вопросы

1. Системное имя, идентификатор пользователя, идентификатор группы, полное имя, домашний каталог, начальная оболочка.
2. 1) `man` (прим. `man ls`); 2) `cd` ; 3) `ls` (если необходимо - уточ.); 4) `du -s $ 5) rm` ; 6) `chmod` (прим. `chmod 777 filename.txt`); 7) `history`.
3. Порядок, опред. способ организации, хранения и именования данных на носителях информации. Прим.: `ext2`, макс. размер 16гб-2гб, макс. размер тома 2гб-32гб, сущ. ед. корневой каталог, макс. длина имени файла - 266 байт.
4. `mount`.
5. Kill (PID - получаем с помощью `ps аху | grep`)

Список литературы

1. Д. К. Лабораторная работа № 1. RUDN.