

Лабораторная работа №1

Операционные системы

Кучмар София Игоревна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	19

Список иллюстраций

3.1	Настройка папки в VirtualBox	7
3.2	Создание виртуальной машины. Часть 1	8
3.3	Создание виртуальной машины. Часть 2	8
3.4	Задание размера диска	9
3.5	Настройка общего буфера обмена и функции Drag'n'Drop	9
3.6	Поддержка UEFI	10
3.7	Поддержка UEFI	10
3.8	Общая папка	11
3.9	Язык интерфейса	11
3.10	Место установки	12
3.11	Переключаемся на роль супер-пользователя	12
3.12	Обновим все пакеты	12
3.13	Программы для удобства работы в консоли	13
3.14	Программное обеспечение для автоматического обновления	13
3.15	Запустим таймер	13
3.16	Отключение SELinux	13
3.17	Создадим конфигурационный файл	14
3.18	Отредактируем конфигурационный файл	14
3.19	Создадим пользователя	14
3.20	Зададим пароль для пользователя	14
3.21	Установим имя хоста	15
3.22	Установим средство pandoc	15
3.23	Установка ghcup	16
3.24	Установка ghc	16
3.25	Установка cabal	16
3.26	Обновление cabal	16
3.27	Установка pandoc-crossref	16
3.28	Установка texlive	17
3.29	Проверка после установки texlive	17
3.30	Версия ядра Linux и частота процессора	17
3.31	Модель процессора	17
3.32	Объем доступной оперативной памяти	18
3.33	Тип обнаруженного гипервизора и файловой системы корневого раздела	18

Список таблиц

1 Цель работы

В рамках данной лабораторной работы мы приобретем практические навыки установки и настройки операционной системы Linux на виртуальную машину Virtualbox.

2 Задание

В процессе выполнения лабораторной работы мы научимся устанавливать операционную систему на диск, запустив соответствующее приложение для установки. После установки мы проведем необходимые обновления, повысим комфорт работы с системой, настроим автоматическое обновление и отключим SELinux. Мы также настроим раскладку клавиатуры, установим имя пользователя и название хоста. В заключение, мы установим необходимое программное обеспечение для создания документации, а именно научимся работать с языком разметки Markdown и установим пакет texlive. В результате мы сделаем рабочее окружение для дальнейших исследований и разработок.

3 Выполнение лабораторной работы

Проведём настройку папки в VirtualBox. В меню выберем Файл, Настройки. Выберем Общие, поле Папка для машин по умолчанию и зададим новое значение (рис. 3.1).

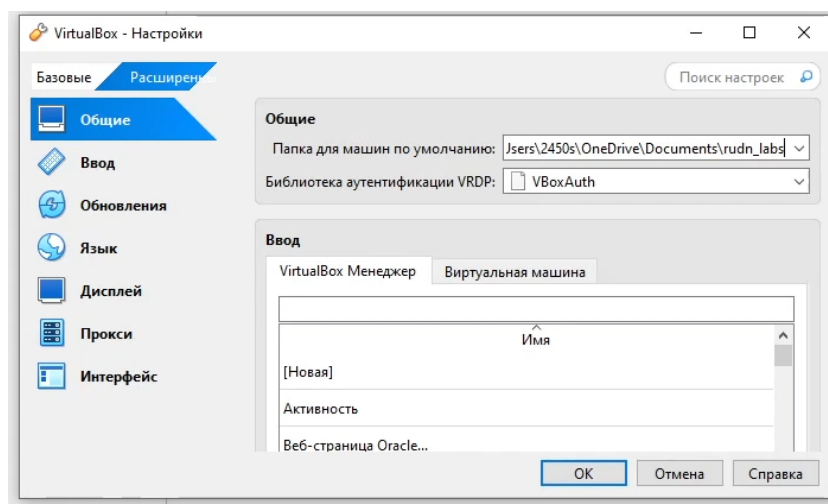


Рис. 3.1: Настройка папки в VirtualBox

Создадим виртуальную машину. Выберем нужный iso, зададим имя, папку и тип (рис. 3.2).

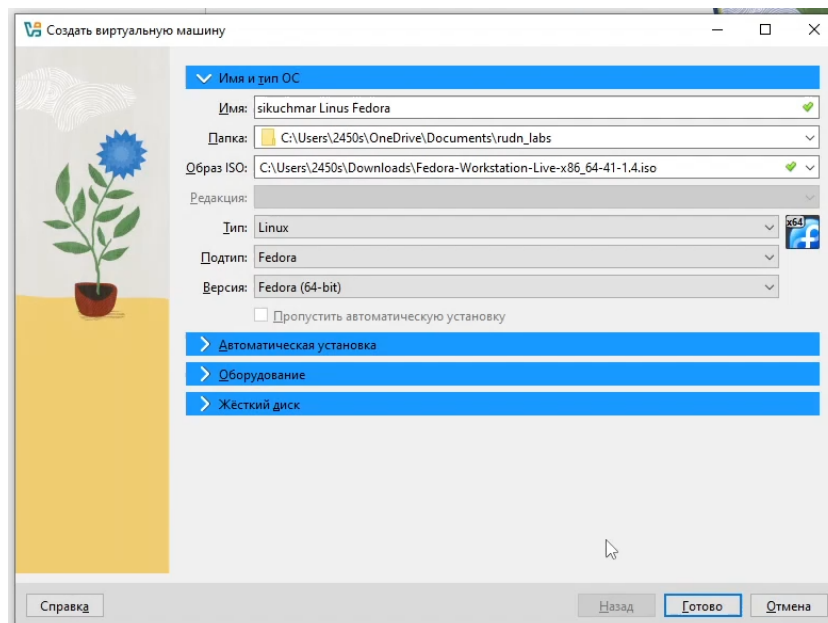


Рис. 3.2: Создание виртуальной машины. Часть 1

Укажем размер основной памяти виртуальной машины — от 2048 МБ (рис. 3.3).

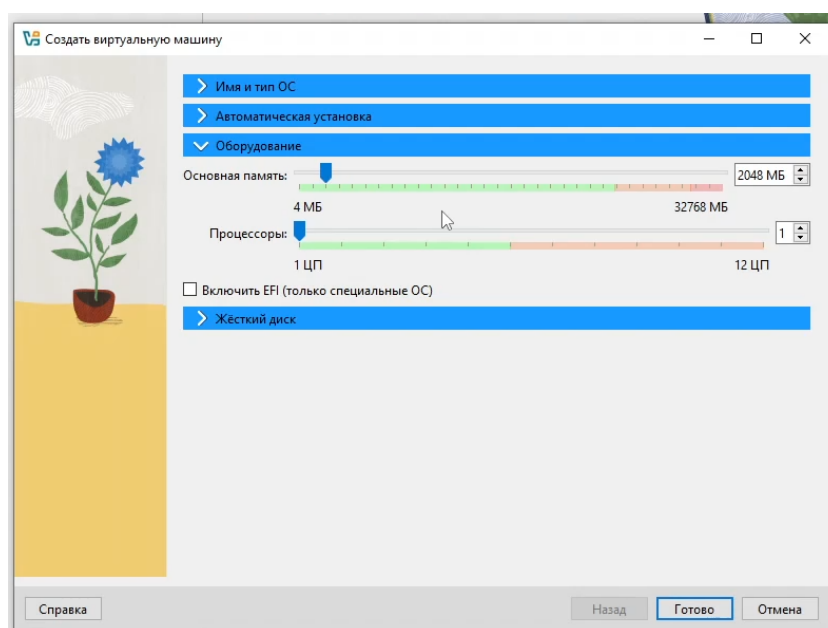


Рис. 3.3: Создание виртуальной машины. Часть 2

Зададим размер диска — 80 ГБ, его расположение (рис. 3.4).

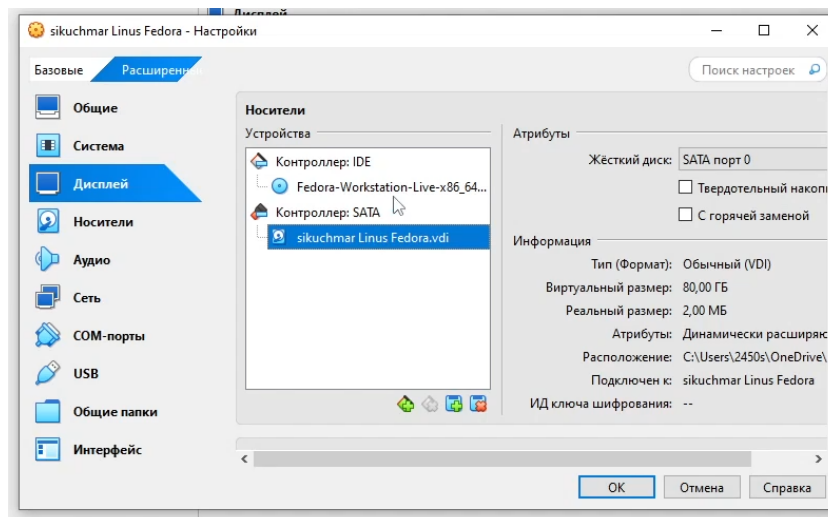


Рис. 3.4: Задание размера диска

Включим двунаправленный общий буфер обмена и функцию Drag'n'Drop (рис. 3.5).

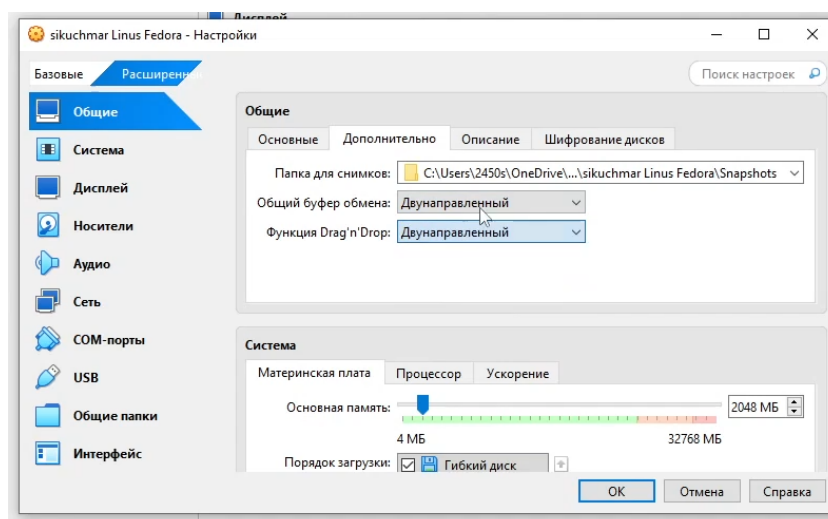


Рис. 3.5: Настройка общего буфера обмена и функции Drag'n'Drop

Включим поддержку UEFI (рис. 3.6).

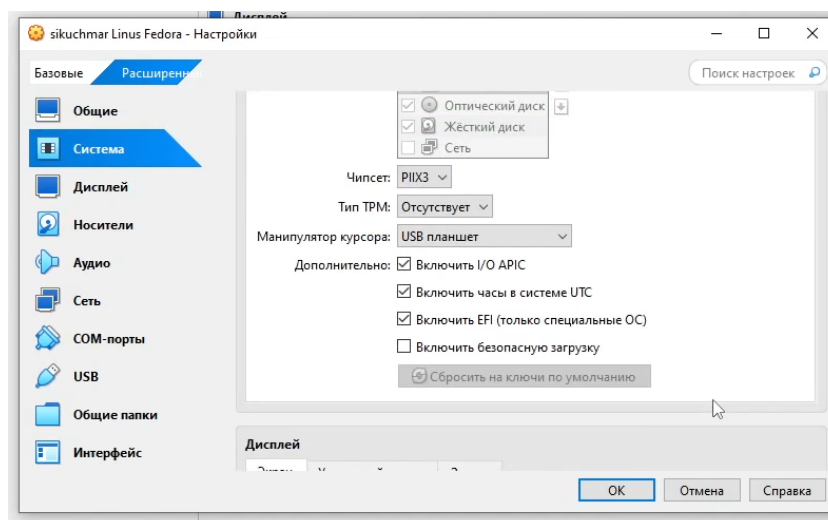


Рис. 3.6: Поддержка UEFI

В качестве графического контроллера поставим VMSVGA и включим ускорение 3D (рис. 3.7).

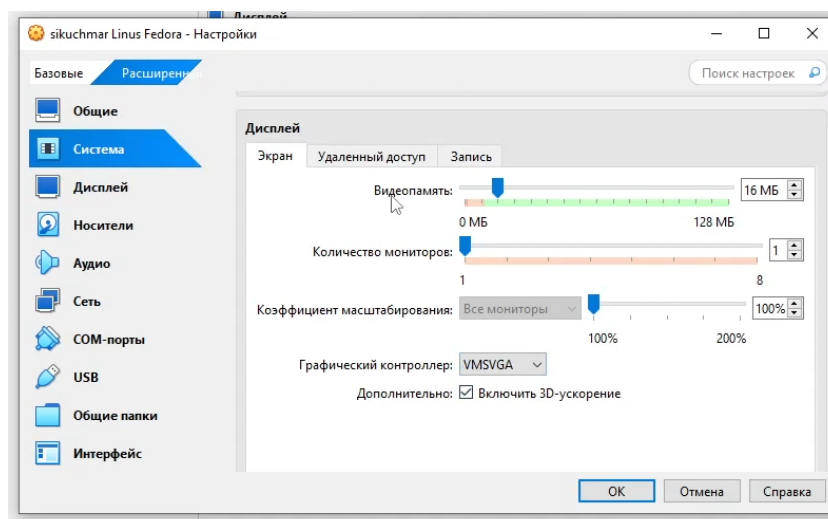


Рис. 3.7: Поддержка UEFI

Добавим общую папку и сохраним (рис. 3.8).

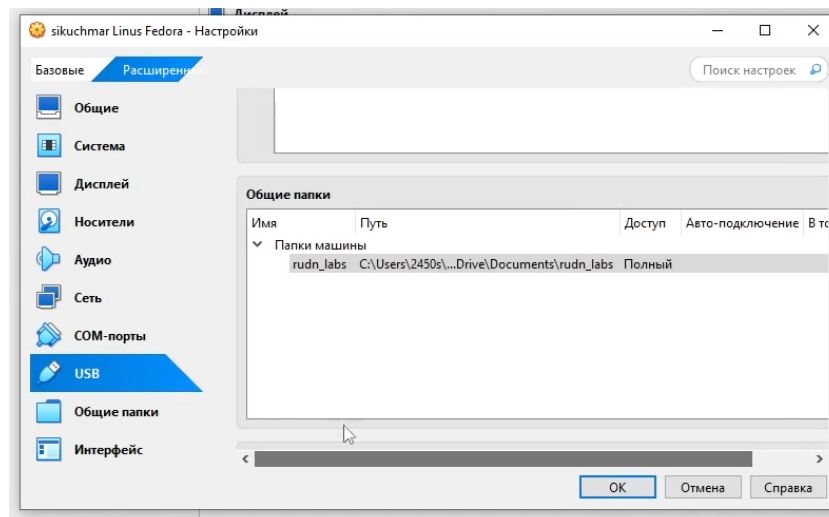


Рис. 3.8: Общая папка

Запустим виртуальную машину. Выберем язык интерфейса (рис. 3.9).

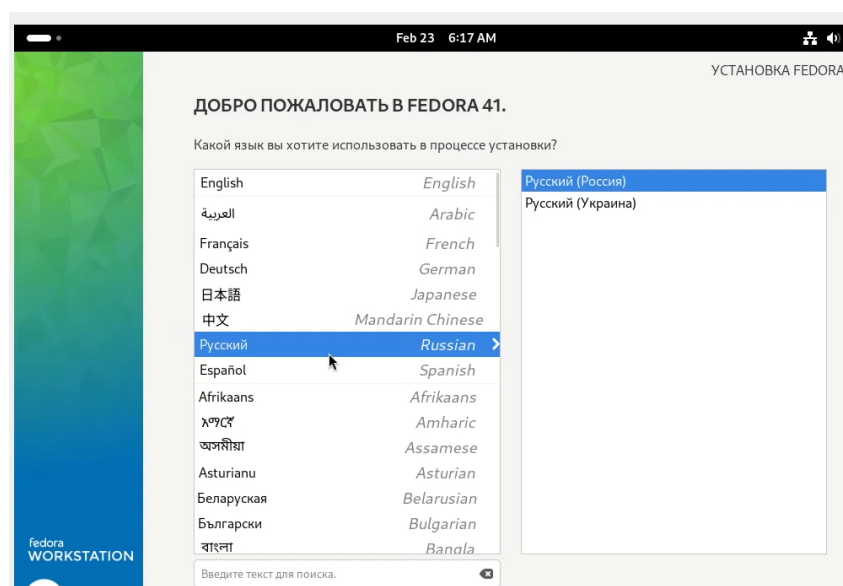


Рис. 3.9: Язык интерфейса

Место установки оставим по-умолчанию. Так же создадим имя пользователя и установим пароль (рис. 3.10).

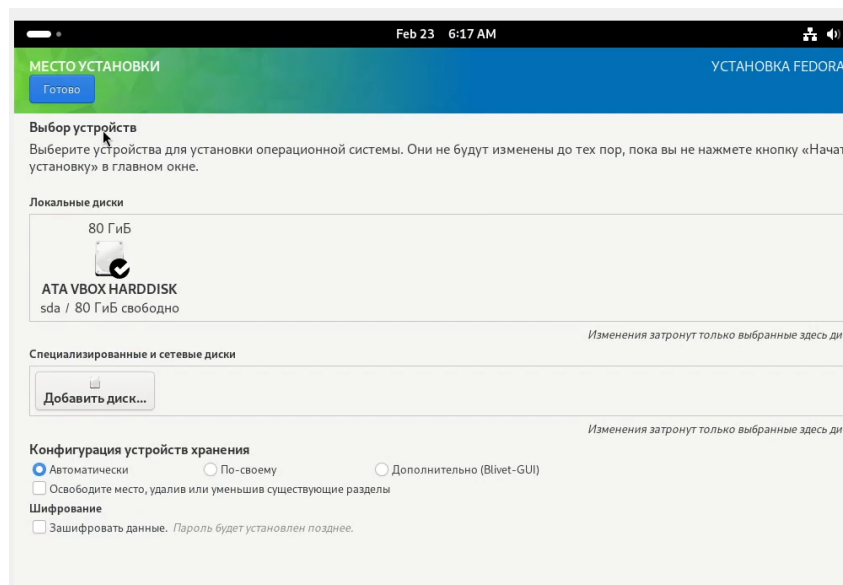


Рис. 3.10: Место установки

После установки переключимся на роль супер-пользователя (рис. 3.11).

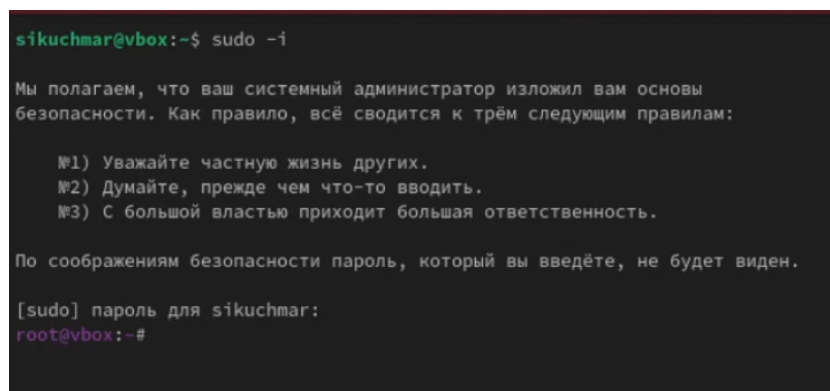


Рис. 3.11: Переключаемся на роль супер-пользователя

Обновим все пакеты (рис. 3.12).

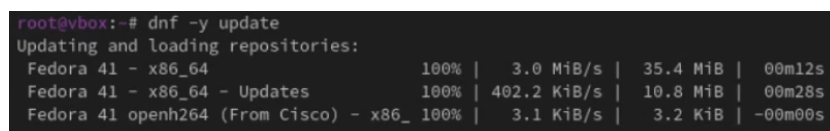


Рис. 3.12: Обновим все пакеты

Установим программы для удобства работы в консоли (рис. 3.13).

```

root@vbox:~# dnf -y install tmux mc
Updating and loading repositories:
Repositories loaded.
Пакет "tmux-3.5a-2.fc41.x86_64" уже установлен.

```

Рис. 3.13: Программы для удобства работы в консоли

Для автоматического обновления установим программное обеспечение (рис. 3.14).

```

root@vbox:~# dnf install dnf-automatic
Updating and loading repositories:
Repositories loaded.

```

Рис. 3.14: Программное обеспечение для автоматического обновления

Запустим таймер (рис. 3.15).

```

root@vbox:~# systemctl enable --now dnf-automatic.timer
Created symlink '/etc/systemd/system/timers.target.wants/dnf-automatic.timer' →
'/usr/lib/systemd/system/dnf-automatic.timer'.

```

Рис. 3.15: Запустим таймер

Отключение SELinux. В файле /etc/selinux/config заменим значение SELinux с enforcing на permissive (рис. 3.16).

```

config  [-M--] 18 L:[ 9+13 22/ 30] *(929 /1188b) 0010 0x00A [*][X]
#
# 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 pro
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted

```

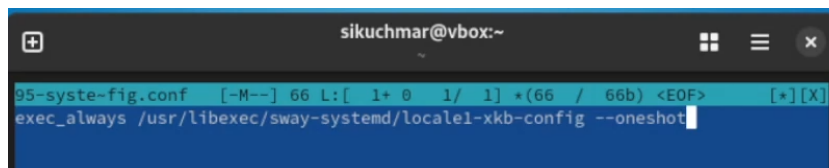
Рис. 3.16: Отключение SELinux

Настройка раскладки клавиатуры. Создадим конфигурационный файл `~/.config/sway/config.d/95-system-keyboard-config.conf` (рис. 3.17).

```
sikuchmar@vbox:~$ mkdir -p ~/.config/sway/config.d
sikuchmar@vbox:~$ touch ~/.config/sway/config.d/95-system-keyboard-config.conf
```

Рис. 3.17: Создадим конфигурационный файл

Отредактируем конфигурационный файл `95-system-keyboard-config.conf` и `/etc/X11/xorg.conf.d/00-keyboard.conf` (рис. 3.18).



```
sikuchmar@vbox:~$ nano ~/.config/sway/config.d/95-system-keyboard-config.conf
95-system-keyboard-config.conf [~M--] 66 L:[ 1+ 0 1/ 1] *(66 / 66b) <EOF> [*][X]
exec_always /usr/libexec/sway-systemd/locale1-xkb-config --oneshot
```

Рис. 3.18: Отредактируем конфигурационный файл

Установка имени пользователя и названия хоста. Создадим пользователя (рис. 3.19).

```
sikuchmar@vbox:~$ sudo -i
[sudo] пароль для sikuchmar:
root@vbox:~# adduser -G wheel sikuchmar
useradd: пользователь «sikuchmar» уже существует
root@vbox:~#
```

Рис. 3.19: Создадим пользователя

Зададим пароль для пользователя (рис. 3.20).

```
root@vbox:~# passwd sikuchmar
Новый пароль:
Повторите ввод нового пароля:
passwd: пароль успешно обновлён
root@vbox:~#
```

Рис. 3.20: Зададим пароль для пользователя

Установим имя хоста и проверим, что имя хоста установлено верно (рис. 3.21).

```
root@ybox:~# hostnamectl set-hostname sikuchmar
root@ybox:~# hostnamectl
  Static hostname: sikuchmar
        Icon name: computer-vm
        Chassis: vm
        Machine ID: 0250865a71af4a2ba085d57953a8ae9d
        Boot ID: 0e41ab913a834c0dad2c574b9d0b85dc
        Product UUID: 30f728a2-49bb-8145-97cd-d1243b80b4fa
        Virtualization: oracle
        Operating System: Fedora Linux 41 (Workstation Edition)
        CPE OS Name: cpe:/o:fedoraproject:fedora:41
        OS Support End: Tue 2025-05-13
        OS Support Remaining: 2month 2w 3d
        Kernel: Linux 6.11.4-301.fc41.x86_64
        Architecture: x86-64
        Hardware Vendor: innotek GmbH
        Hardware Model: VirtualBox
        Hardware Serial: VirtualBox-a228f730-bb49-4581-97cd-d12...
        Firmware Version: VirtualBox
        Firmware Date: Fri 2006-12-01
        Firmware Age: 18y 2month 3w 3d
```

Рис. 3.21: Установим имя хоста

Установка программного обеспечения для создания документации. Работа с языком разметки Markdown. Установим средство pandoc для работы с языком разметки Markdown (рис. 3.22).

```
root@ybox:~# 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.
```

Рис. 3.22: Установим средство pandoc

Установка ghcup (рис. 3.23).

```

root@vbox:~# curl --proto '=https' --tlsv1.2 -sSf https://get-ghcup.haskell.org | sh

Welcome to Haskell!

This script can download and install the following binaries:
* ghcup - The Haskell toolchain installer
* ghc   - The Glasgow Haskell Compiler
* cabal - The Cabal build tool for managing Haskell software
* stack - A cross-platform program for developing Haskell projects (similar to cabal)
* hls   - (optional) A language server for developers to integrate with their editor/IDE

```

Рис. 3.23: Установка ghcup

Установка ghc (рис. 3.24).

```

root@sikuchmar:~# ghcup install ghc
[ Info ] downloading: https://raw.githubusercontent.com/haskell/ghcup-metadata/master/ghcup-0.0.9.yaml as file /root/.ghcup/cache/ghcup-0.0.9.yaml

```

% Total	% Received	% Xferd	Average Speed	Time Dload	Time Upload	Time Total	Time Spent	Time Left	Current Speed
0	0	0	0	0	0	0	0	0	0

Рис. 3.24: Установка ghc

Установка cabal (рис. 3.25).

```

root@sikuchmar:~# ghcup install cabal
[ Warn ] New ghc version available. If you want to install this latest version, run 'ghcup install ghc 9.12.1'
[ Warn ] New cabal version available. If you want to install this latest version, run 'ghcup install cabal 3.14.1.1'
[ Warn ] [GHCup-00110] cabal-3.12.1.0 is already installed; if you really want to reinstall it, you may want to run 'ghcup install cabal --force 3.12.1.0'

```

Рис. 3.25: Установка cabal

Обновление cabal (рис. 3.26).

```

root@sikuchmar:~# cabal v2-update
Downloading the latest package list from hackage.haskell.org
Package list of hackage.haskell.org has been updated.
The index-state is set to 2025-02-23T13:50:17Z.
To revert to previous state run:
  cabal v2-update 'hackage.haskell.org,2025-02-23T12:30:20Z'
root@sikuchmar:~#

```

Рис. 3.26: Обновление cabal

Установка pandoc-crossref (рис. 3.27).

```

root@sikuchmar:~# cabal v2-install --install-method=copy pandoc-cli pandoc-crossref
Resolving dependencies...

```

Рис. 3.27: Установка pandoc-crossref

Установка texlive (рис. 3.28).

```
root@sikuchmar:~# dnf install texlive texlive-\\*
Updating and loading repositories:
Repositories loaded.
```

Рис. 3.28: Установка texlive

Проверим, что всё установилось (рис. 3.29).

```
root@sikuchmar:~# lualatex
This is LuaHBTeX, Version 1.17.0 (TeX Live 2023)
restricted system commands enabled.
**
Please type the name of your input file.
**^C
root@sikuchmar:~# pdflatex
This is pdfTeX, Version 3.141592653-2.6-1.40.25 (TeX Live 2023) (preloaded format=pdflatex)
restricted \writel8 enabled.
**
! End of file on the terminal... why?
root@sikuchmar:~# xellatex
This is XeTeX, Version 3.141592653-2.6-0.999995 (TeX Live 2023) (preloaded format=xellatex)
restricted \writel8 enabled.
**
! End of file on the terminal... why?
```

Рис. 3.29: Проверка после установки texlive

Домашнее задание. С помощью грег получим информацию о версии ядра Linux и частоте процессора (рис. 3.30).

```
root@sikuchmar:~# dmesg | grep -i "Linux version"
[ 0.000000] Linux version 6.11.4-301.fc41.x86_64 (mockbuild@09b6b61418589428cb880a7020233b56f) (gcc (GCC) 14.2.1 20
Red Hat 14.2.1-3), GNU ld version 2.43.1-2.fc41) #1 SMP PREEMPT_DYNAMIC Sun Oct 20 15:02:33 UTC 2024
root@sikuchmar:~# dmesg | grep -i "Detected Mhz processor"
[ 0.000005] tsc: Detected 2591.998 MHz processor
[ 6.430626] e1000 0000:00:03:00:eth0: (PCI:33MHz:32-bit) 08:00:27:f2:b1:71
```

Рис. 3.30: Версия ядра Linux и частота процессора

С помощью грег получим информацию о модели процессора (рис. 3.31).

```
root@sikuchmar:~# dmesg | grep -i "CPU0:"
[ 0.196831] smpboot: CPU0: 11th Gen Intel(R) Core(TM) i5-11400F @ 2.60GHz (family: 0x6, model: 0xa7,
```

Рис. 3.31: Модель процессора

С помощью грег получим информацию о объёме доступной оперативной памяти (рис. 3.32).

```

root@sikuchmar:~# dmesg | grep -i "RAM"
[ 0.000000] BIOS-provided physical RAM map:
[ 0.000058] total RAM covered: 8192M
[ 0.000753] gran_size: 64K chunk_size: 64K num_reg: 5 lose cover RAM: 0G
[ 0.001201] RAMDISK: [mem 0xda056000-0xdbfb4fff]
[ 0.024461] Booted with the nomodeset parameter. Only the system framebuffer will be available
[ 0.024480] Unknown kernel command line parameters "rhgb BOOT_IMAGE=(hd0,gpt2)/vmlinuz-6.11.4-301.fc41.x86_64", w

```

Рис. 3.32: Объём доступной оперативной памяти

С помощью `grep` получим информацию о типах обнаруженного гипервизора и файловой системы корневого раздела (рис. 3.33).

```

root@sikuchmar:~# dmesg | grep -i "Hypervisor detected"
[ 0.000000] Hypervisor detected: KVM
root@sikuchmar:~# df -T / | awk 'NR==2 {print $2}'
btrfs

```

Рис. 3.33: Тип обнаруженного гипервизора и файловой системы корневого раздела

4 Выводы

В рамках лабораторной работы мы установили Linux на Virtualbox, получив практические навыки по установке ОС на диск, запуску инсталлятора, проведению обновлений, повышению удобства работы, настройке автоматического обновления, отключению SELinux, настройке раскладки клавиатуры, установке имени пользователя и хоста, а также установке необходимого ПО для работы с Markdown и пакета texlive, что позволит создать полноценное рабочее окружение для дальнейших исследований и разработок.