

Отчёт по лабораторной работе №1

Бондарь Татьяна Владимировна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
4.1	Создание виртуальной машины	8
4.2	После установки	9
4.2.1	Обновления	9
4.2.2	Повышение комфорта работы. Отключение SELinux	10
4.3	Настройка раскладки клавиатуры	11
4.3.1	Автоматическое обновление	11
4.4	Установка программного обеспечения для создания документации	12
4.5	Домашнее задание	13
5	Контрольные вопросы	16
6	Выводы	18
	Список литературы	19

Список иллюстраций

4.1	Конфигурации новой виртуальной машины	8
4.2	Установка ОС	9
4.3	Обновление пакетов	10
4.4	Отключение SELinux	10
4.5	Создание конфигурационного файла	11
4.6	Редактирование файла	11
4.7	Редактирование файла	11
4.8	Установка ПО	12
4.9	Редактирование файла	12
4.10	Скачивание pandoc, pandoc-crossref	12
4.11	Перенос файлов в нужный каталог	13
4.12	Установка TexLive	13
4.13	Выполнение команды	14
4.14	Получение необходимой информации ч.1	14
4.15	Получение необходимой информации ч.2	14
4.16	Тип файловой системы корневого каталога	15

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)Запуск VirtualBox и создание новой виртуальной машины (операционная система Linux, Fedora). 2)Настройка установки ОС. 3)Перезапуск виртуальной машины и установка драйверов для VirtualBox. 4)Подключение образа диска дополнений гостевой ОС. 5)Установка необходимого ПО для создания документации. 6)Выполнение домашнего задания.

3 Теоретическое введение

Операционная система - это комплекс взаимосвязанных программ, который действует как интерфейс между приложениями и пользователями с одной стороны и аппаратурой компьютера с другой стороны. VirtualBox - это специальное средство для виртуализации, позволяющее запускать операционную систему внутри другой. С помощью VirtualBox мы можем также настраивать сеть, обмениваться файлами и делать многое другое.

4 Выполнение лабораторной работы

4.1 Создание виртуальной машины

1. Создаем новую виртуальную машину, указываем имя. Указываем размер основной памяти, задаем размер диска. Добавляем новый привод оптических дисков и выбираем образ операционной системы Fedora. (рис. fig. 4.1).

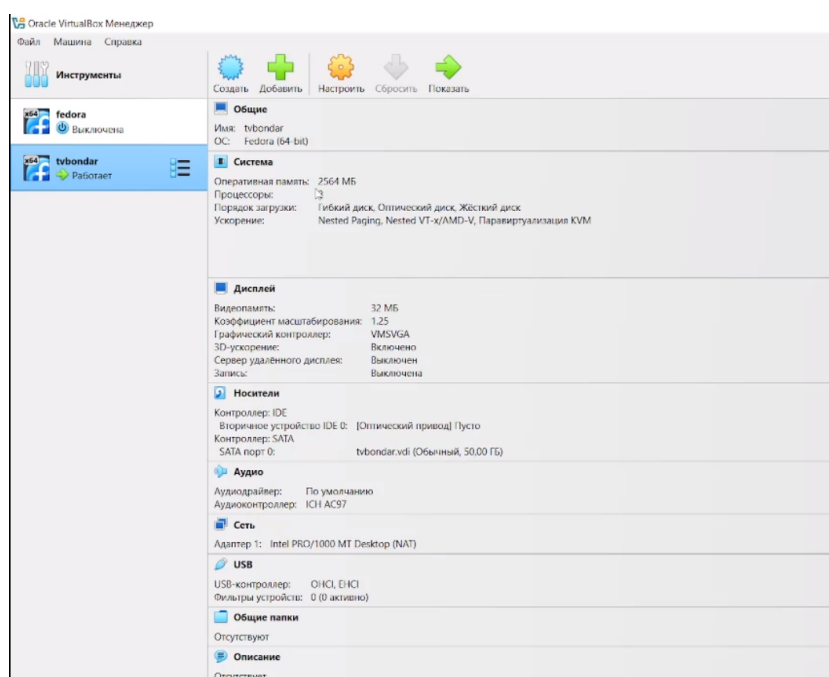


Рис. 4.1: Конфигурации новой виртуальной машины

2. Производим установку операционной системы. (рис. fig. 4.2).

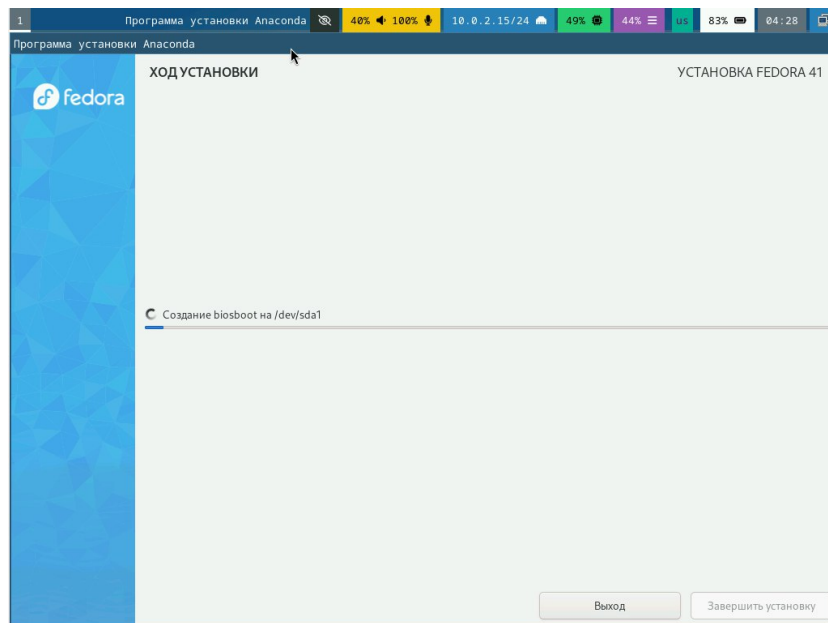
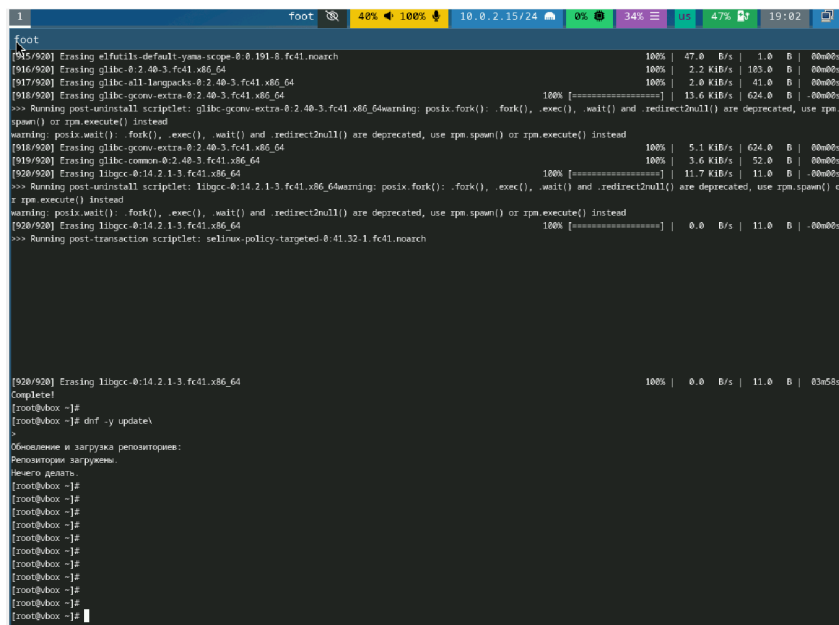


Рис. 4.2: Установка ОС

4.2 После установки

4.2.1 Обновления

3. Входим в ОС под своей учетной записью. Открываем терминал. Переключаемся на роль суперпользователя. Производим установку обновлений. (рис. fig. 4.3).



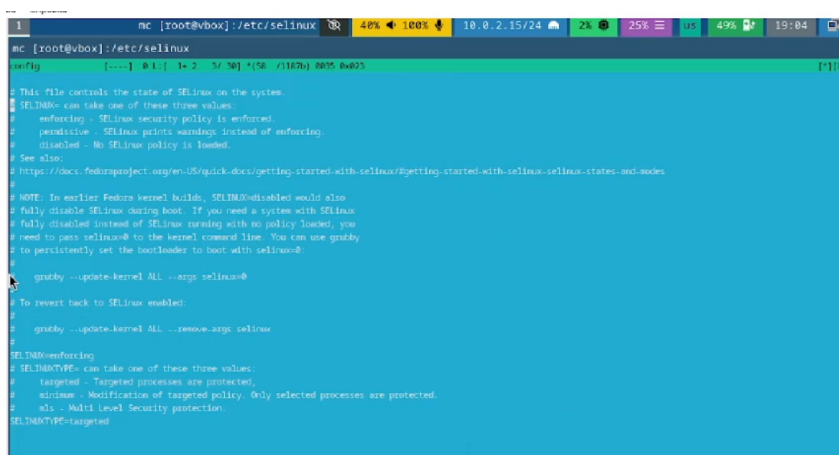
```
root
[915/920] Erasing elfutils-default-yama-scope-0.0.191-8.fc41.noarch 100% | 47.0 B/s | 1.0 B | 00m00s
[916/920] Erasing glibc-0.2.40-3.fc41.x86_64 100% | 2.2 KiB/s | 103.0 B | 00m00s
[917/920] Erasing glibc-all-langpacks-0.2.40-3.fc41.x86_64 100% | 2.0 KiB/s | 41.0 B | 00m00s
[918/920] Erasing glibc-gconv-extra-0.2.40-3.fc41.x86_64 100% [=====] | 13.6 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
[918/920] Erasing glibc-gconv-extra-0.2.40-3.fc41.x86_64 100% | 5.1 KiB/s | 624.0 B | 00m00s
[919/920] Erasing glibc-common-0.2.40-3.fc41.x86_64 100% | 3.6 KiB/s | 52.0 B | 00m00s
[920/920] Erasing libgcc-0:14.2.1-3.fc41.x86_64 100% [=====] | 11.7 KiB/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
[920/920] Erasing libgcc-0:14.2.1-3.fc41.x86_64 100% [=====] | 0.0 B/s | 11.0 B | 00m00s
>>> Running post-transaction scriptlet: selinux-policy-targeted-0.41.32-1.fc41.noarch

[920/920] Erasing libgcc-0:14.2.1-3.fc41.x86_64 100% | 0.0 B/s | 11.0 B | 03m54s
Complete!
[root@vbox ~]#
[root@vbox ~]# dnf -y update
>
Обновление и загрузка репозитория:
Репозитории загрузки.
Нечего делать.
[root@vbox ~]#
[root@vbox ~]#
[root@vbox ~]#
[root@vbox ~]#
[root@vbox ~]#
[root@vbox ~]#
[root@vbox ~]#
[root@vbox ~]#
[root@vbox ~]#
[root@vbox ~]#
[root@vbox ~]#
```

Рис. 4.3: Обновление пакетов

4.2.2 Повышение комфорта работы. Отключение SELinux

- Устанавливаем программу tmux. (рис. fig. 4.4). Запускаем tmux, открываем midnight commander, в файле /etc/selinux/config заменяем значение SELINUX=enforcing на SELINUX=permissive (рис. fig. 4.4).



```
mc [root@vbox] /etc/selinux
mc [root@vbox] /etc/selinux
config [-----] 0.1.1 3+2 3/30 5156 /11676 8805 64035 [71]

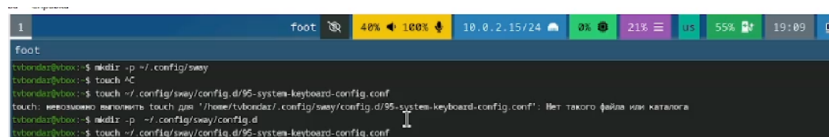
# This file controls the state of SELinux on the system.
# SELinux can take one of three values:
# enforcing - SELinux security policy is enforced.
# permissive - SELinux prints warnings instead of enforcing.
# disabled - No SELinux policy is loaded.
# See also:
# https://docs.fedoraproject.org/en-US/quick-docs/getting-started-with-selinux/#getting-started-with-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=enforcing
# SELINUXTYPE can take one of three values:
# targeted - Targeted processes are protected.
# minimum - Modification of targeted policy. Only selected processes are protected.
# all - Multi-level Security protection.
SELINUXTYPE=targeted
```

Рис. 4.4: Отключение SELinux

- Перегружаем виртуальную машину.

4.3 Настройка раскладки клавиатуры

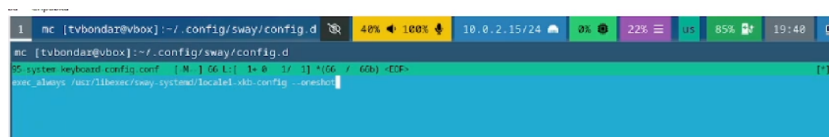
5. Создаем конфигурационный файл `~/.config/sway/config.d/95-system-keyboard-config.conf`. (рис. fig. 4.5).



```
foot
tvbonda@vbox:~$ mkdir -p ~/.config/sway
tvbonda@vbox:~$ touch ~/.config/sway
tvbonda@vbox:~$ touch ~/.config/sway/config.d/95-system-keyboard-config.conf
touch: невозможно создать файл '/home/tvbonda/.config/sway/config.d/95-system-keyboard-config.conf': Нет такого файла или каталога
tvbonda@vbox:~$ mkdir -p ~/.config/sway/config.d
tvbonda@vbox:~$ touch ~/.config/sway/config.d/95-system-keyboard-config.conf
```

Рис. 4.5: Создание конфигурационного файла

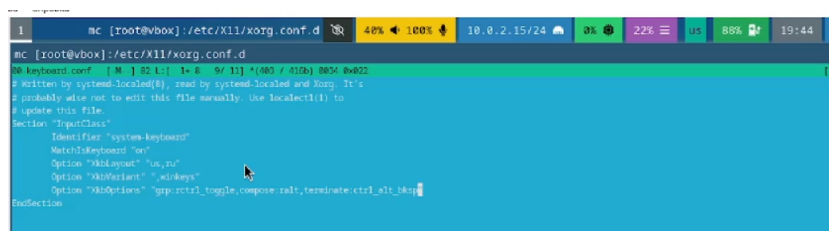
6. Отредактируем конфигурационный файл. (рис. fig. 4.6).



```
mc [tvbonda@vbox]:~/.config/sway/config.d
mc [tvbonda@vbox]:~/.config/sway/config.d
95-system-keyboard-config.conf [M] 06 L: 1+8 1/ 1] *06 / 60b) <CP>
enter always /usr/libexec/sway-systemd/localed.vkb.config --swscho
```

Рис. 4.6: Редактирование файла

7. Отредактируем конфигурационный файл `/etc/X11/xorg.conf.d/00-keyboard.conf`. Перегрузим виртуальную машину. (рис. fig. 4.7).



```
mc [root@vbox]:/etc/X11/xorg.conf.d
mc [root@vbox]:/etc/X11/xorg.conf.d
00-keyboard.conf [M] 82 L: 1+8 9/ 11] *403 / 410b) 0034 0x032
# Written by systemd-localed(4), read by systemd-localed 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,en"
    Option "XkbOptions" "ctrl:alt::caps"
    Option "XkbOptions" "grp:ctrl_alt_toggle,compose:ctrl:alt::caps,terminate:ctrl:alt::backspace"
EndSection
```

Рис. 4.7: Редактирование файла

4.3.1 Автоматическое обновление

8. Установим ПО для автоматического обновления (рис. fig. 4.8).

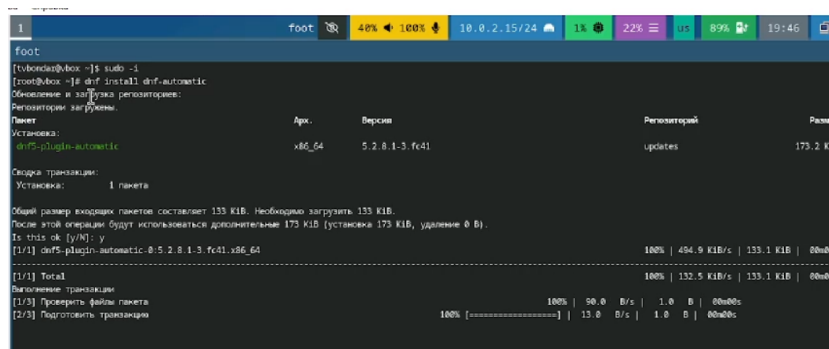


Рис. 4.8: Установка ПО

9. Отредактируем необходимый конфигурационный файл. Запустим таймер. (рис. fig. 4.9).



Рис. 4.9: Редактирование файла

4.4 Установка программного обеспечения для создания документации

10. Скачиваем pandoc и pandoc-crossref из репозитория Github. (рис. fig. 4.10).

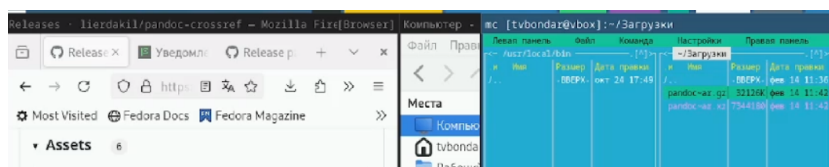


Рис. 4.10: Скачивание pandoc, pandoc-crossref

11. Переносим необходимые файлы из архивов в каталог /usr/local/bin. (рис. fig. 4.11).

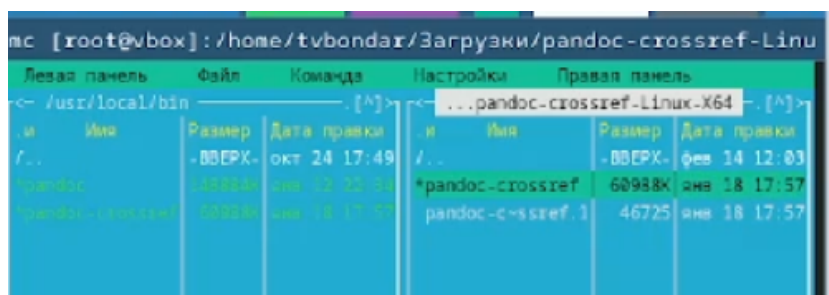


Рис. 4.11: Перенос файлов в нужный каталог

12. Установим дистрибутив TexLive (рис. fig. 4.12).

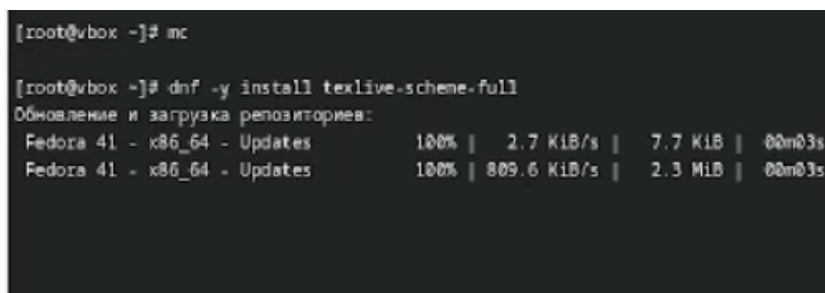


Рис. 4.12: Установка TexLive

4.5 Домашнее задание

13. Посмотрим порядок загрузки системы с помощью команды dmesg. (рис. fig. 4.13).

```

[ 0.000000] Linux version 6.12.11-200.fc41.x86_64 (mockbuild@fc085b49e2e66083907ced8b4d4f464) (gcc (GCC) 14.2.1 20250110 (Red Hat 14.2.1-7), GNU ld version 2.43.1-5.7)
c41) #1 SMP PREEMPT_DYNAMIC Fri Jan 24 04:59:58 UTC 2025
[ 0.000000] Command line: BOOT_IMAGE=(hdd,gpt2)/vmlinuz-6.12.11-200.fc41.x86_64 root=UUID=fca7b66f-8741-4844-9771-85638d7cfd33 ro rootflags=subvol=root nomodeset vga=791 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 0x0000000000000000-0x00000000000000ff] usable
[ 0.000000] BIOS-e820: [mem 0x00000000000000fc-0x00000000000000ff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000000000-0x00000000000000ff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000000000-0x00000000000000ff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000000000-0x00000000000000ff] usable
[ 0.000000] BIOS-e820: [mem 0x0000000000000000-0x00000000000000ff] ACPI data
[ 0.000000] BIOS-e820: [mem 0x0000000000000000-0x00000000000000ff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000000000-0x00000000000000ff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000000000-0x00000000000000ff] reserved
[ 0.000000] NX (Execute Disable) protection: active
[ 0.000000] APIC: Static calls initialized
[ 0.000000] SMBIOS 2.5 present:
[ 0.000000] DMI: Evteck GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/04/2006
[ 0.000000] DMI: Memory slots populated: 8/8
[ 0.000000] Hypervisor detected: KVM
[ 0.000000] kvm-clock: Using msrc 40564001 and 40564000
[ 0.000000] kvm-clock: using sched offset of 965918726 cycles
[ 0.000015] clocksource: kvm-clock: mask: 0xfffffffffff max_cycles: 0x102eddfb, max_idle_ns: 88159059483 ns
[ 0.000025] tsc: Detected 2595.116 MHz processor
[ 0.000000] ...

```

Рис. 4.13: Выполнение команды

14. Получаем информацию о версии ядра Linux, частоте процессора, модели процессора, объеме доступной оперативной памяти, типе обнаруженного гипервизора. (рис. fig. 4.14). (рис. fig. 4.15)

```

[tubondar@vbox ~]$ sudo -i
root@vbox:~# dmesg | less
root@vbox:~#
root@vbox:~# dmesg | less
root@vbox:~#
root@vbox:~# dmesg | grep -i "Linux version"
[ 0.000000] Linux version 6.12.11-200.fc41.x86_64 (mockbuild@fc085b49e2e66083907ced8b4d4f464) (gcc (GCC) 14.2.1 20250110 (Red Hat 14.2.1-7), GNU ld version 2.43.1-5.7)
c41) #1 SMP PREEMPT_DYNAMIC Fri Jan 24 04:59:58 UTC 2025
root@vbox:~# dmesg | grep -i "processor"
[ 0.000025] tsc: Detected 2595.116 MHz processor
[ 0.225268] smpboot: Total of 2 processors activated (18380.58 BogoMIPS)
[ 0.225268] ACPI: Added _OSI (Processor Device)
[ 0.225268] ACPI: Added _OSI (Processor Aggregator Device)
root@vbox:~# dmesg | grep -i "CPU"
[ 0.213868] smpboot: CPU0: AMD Ryzen 3 3200U with Radeon Graphics (family: 0x17, model: 0x18, stepping: 0x1)
root@vbox:~# dmesg | grep -i "memory"
[ 0.000000] DMI: Memory slots populated: 8/8
[ 0.007332] ACPI: Reserving FACP table memory at [mem 0xa03f00f0-0xa03f01e3]
[ 0.007333] ACPI: Reserving DSDT table memory at [mem 0xa03f0c18-0xa03f2962]
[ 0.007334] ACPI: Reserving FACS table memory at [mem 0xa03f0200-0xa03f023f]
[ 0.007335] ACPI: Reserving FACS table memory at [mem 0xa03f0200-0xa03f023f]
[ 0.007336] ACPI: Reserving APIC table memory at [mem 0xa03f0240-0xa03f025b]
[ 0.007337] ACPI: Reserving SSDT table memory at [mem 0xa03f02a8-0xa03f060b]
[ 0.007365] Early memory node ranges:
[ 0.018381] PM: hibernation: Registered noave memory: [mem 0x00000000-0x000000ff]
[ 0.018402] PM: hibernation: Registered noave memory: [mem 0x00000000-0x000000ff]
[ 0.018403] PM: hibernation: Registered noave memory: [mem 0x00000000-0x000000ff]
[ 0.018404] PM: hibernation: Registered noave memory: [mem 0x00000000-0x000000ff]
[ 0.018405] PM: hibernation: Registered noave memory: [mem 0x00000000-0x000000ff]
[ 0.018406] PM: hibernation: Registered noave memory: [mem 0x00000000-0x000000ff]
[ 0.114980] Freeing SMP alternatives memory: 48K
[ 0.125598] memory: 2677824K/2629888K available (21530K kernel code, 6429K rodata, 16788K rodata, 4888K init, 4728K boot, 148552K reserved, 8K cma-reserved)

```

Рис. 4.14: Получение необходимой информации ч.1

```

> dmesg | grep -i "Hypervisor detected"
grep: detected: Нет такого файла или каталога
root@vbox:~# dmesg | grep -i "Hypervisor detected"
[ 0.000000] Hypervisor detected: KVM
root@vbox:~#

```

Рис. 4.15: Получение необходимой информации ч.2

15. Получаем информацию о последовательности монтирования файловых систем. Тип файловой системы - Ext4. (рис. fig. 4.16).

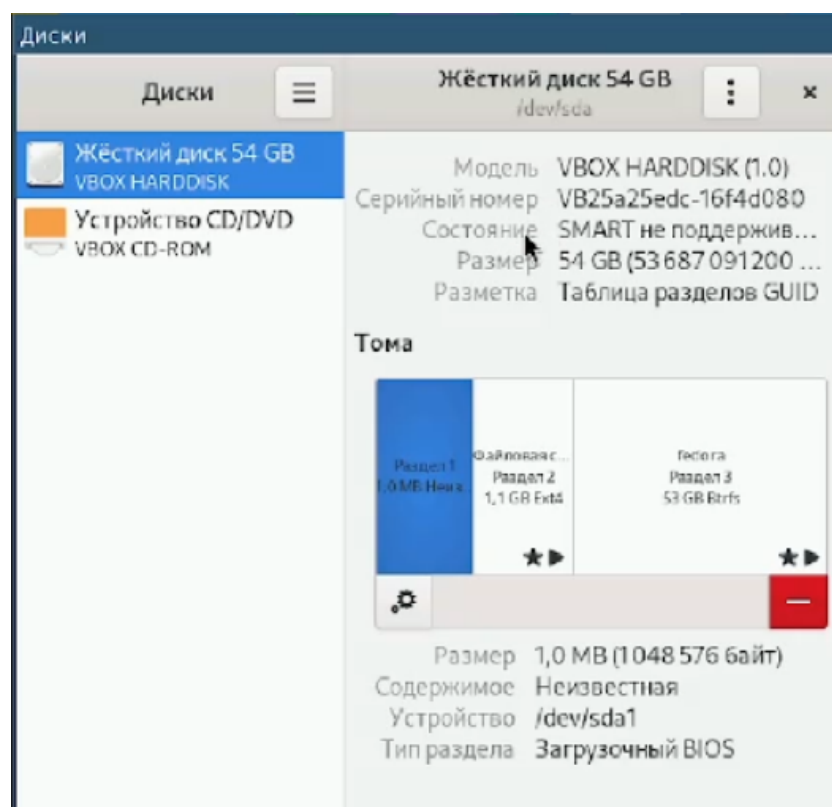


Рис. 4.16: Тип файловой системы корневого каталога

5 Контрольные вопросы

1) Какую информацию содержит учетная запись пользователя?

Имя пользователя, зашифрованный пароль пользователя, идентификационный номер пользователя, идентификационный номер группы пользователя, домашний каталог пользователя, командный интерпретатор пользователя.

2) Укажите команды терминала и приведите примеры: -для получения справки по команде: `man man cd` -для перемещения по файловой системе: `cd cd ~/Downloads` -для просмотра содержимого каталога: `ls ls ~/Downloads` -для определения объема каталога: `du du Downloads` -для создания каталогов: `mkdir mkdir ~/Downloads/New` -для создания файлов: `touch touch retouch` -для удаления каталогов: `rm rm dir1` -для удаления файлов: `rm -r rm -r text.txt` -для задания определенных прав на файл или каталог: `chmod + x chmod +x text.txt` -для просмотра истории команд: `history`

3) Что такое файловая система? Приведите примеры с краткой характеристикой.

Файловая система - это часть операционной системы, назначение которой состоит в том, чтобы обеспечить пользователю удобный интерфейс при работе с данными, хранящимися на диске, и обеспечить совместное использование файлов несколькими пользователями и процессорами. Примеры файловых систем: Ext2, Ext3, Ext4 или Extended Filesystem - стандартная файловая система для Linux. NTFS (New Technology File System): Стандартная файловая система для Windows.

4) Как посмотреть, какие файловые системы подмонтированы в ОС?

С помощью команды `mount`

5) Как удалить зависший процесс?

С помощью команды `kill`.

6 Выводы

В результате выполнения лабораторной работы мы приобрели навыки установки операционной системы на виртуальную машину, а также настройки минимально необходимых для дальнейшей работы сервисов.

Список литературы

- 1) Кулябов Д. С. Введение в операционную систему UNIX - Лекция.
- 2) Таненбаум Э., Бос Х. Современные операционные системы. - 4-е изд. -СПб. : Питер, 2015. - 1120 с.