

Лабораторная работа №1

Бондарь Т. В.

Российский университет дружбы народов, Москва, Россия

Информация

- Бондарь Татьяна Владимировна
- НКАбд-01-24, студ. билет №1132246711
- Российский университет дружбы народов
- https://github.com/tvbondar/study_2024-2025_os-intro

.. ..

Выполнение лабораторной работы №1

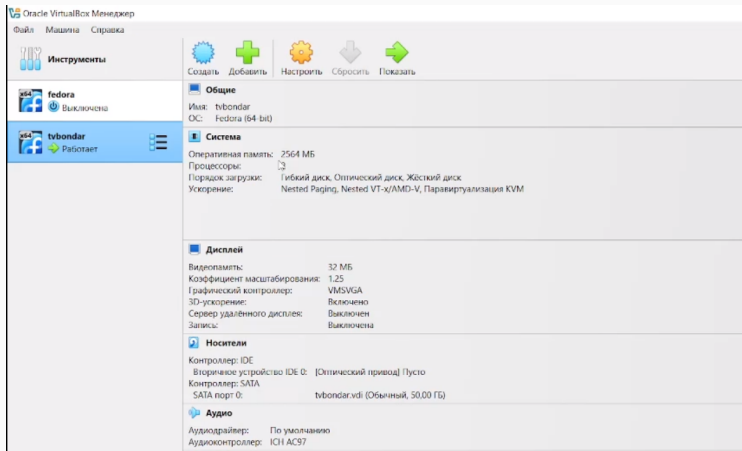
Целью данной работы является приобретение практических навыков установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов.

- 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.

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.

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.



2. Производим установку операционной системы.

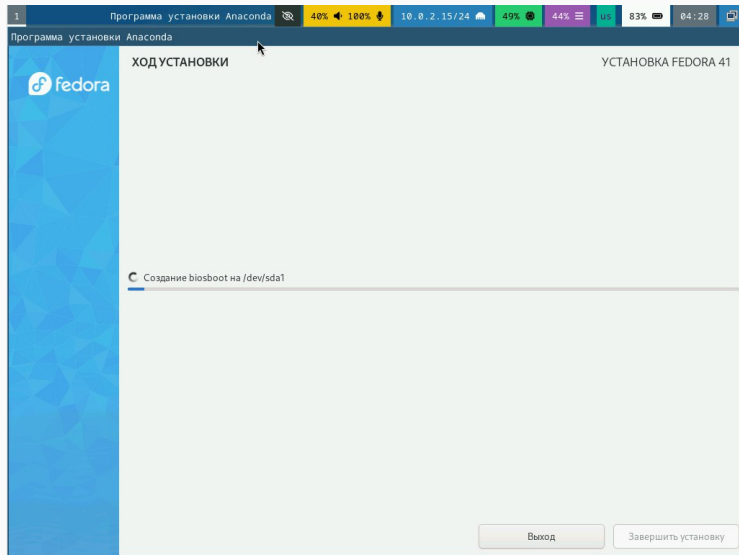


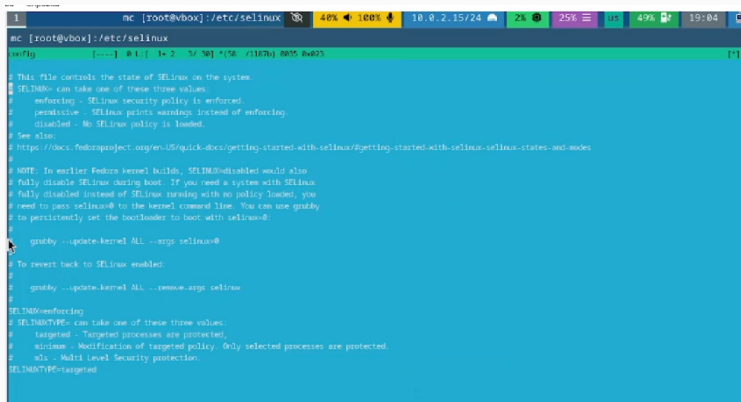
Рис. 2: Установка ОС

3. Входим в ОС под своей учетной записью. Открываем терминал. Переключаемся на роль суперпользователя. Производим установку обновлений.

```
1 root 48% 100% 10.0.2.15/24 0% 34% 47% 19:02
root
[915/920] Erasing elfutils-default-yama-scope-0.8.191-8.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() o
r 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 | 03m58s
Complete!
[root@vbox ~]#
[root@vbox ~]# dnf -y update\
>
Обновление и загрузка репозитория:
Репозитории загружены.
Нечего делать.
[root@vbox ~]#
[root@vbox ~]#
[root@vbox ~]#
[root@vbox ~]#
[root@vbox ~]#
[root@vbox ~]#
```

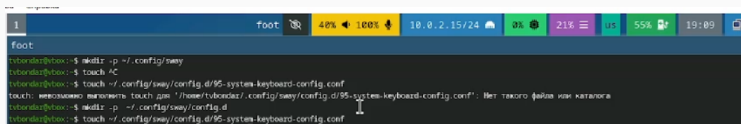
4. Устанавливаем программу tmux. Запускаем tmux, открываем midnight commander, в файле `/etc/selinux/config` заменяем значение `SELINUX=enforcing` на `SELINUX=permissive`



```
1 nc [root@vbox]:/etc/selinux
nc [root@vbox]:/etc/selinux
# 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/#getting-started-with-selinux-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 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
```

5. Создаем конфигурационный файл

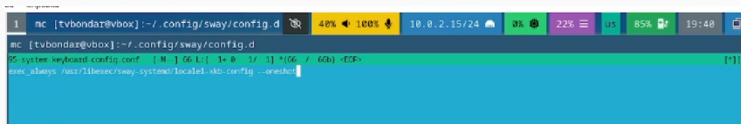
~/config/sway/config.d/95-system-keyboard-config.conf.



```
1 foot 40% 100% 10.0.2.15/24 0% 21% US 55% 19:09
tvbondar@vbox:~$ mkdir -p ~/.config/sway
tvbondar@vbox:~$ touch AC
tvbondar@vbox:~$ touch ~/.config/sway/config.d/95-system-keyboard-config.conf
touch: невозможно создать touch для '/home/tvbondar/.config/sway/config.d/95-system-keyboard-config.conf': Нет такого файла или каталога
tvbondar@vbox:~$ mkdir -p ~/.config/sway/config.d
tvbondar@vbox:~$ touch ~/.config/sway/config.d/95-system-keyboard-config.conf
```

Рис. 5: Создание конфигурационного файла

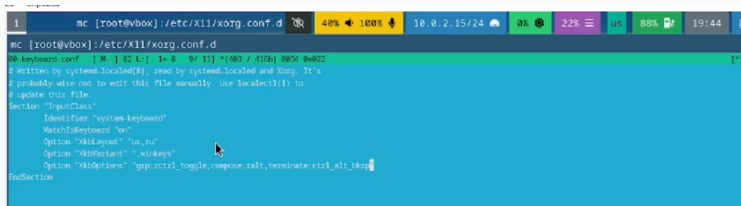
6. Отредактируем конфигурационный файл.



```
1 nc [tvbondar@vbox]:~/.config/sway/config.d 40% 100% 10.0.2.15/24 0% 22% US 85% 19:48
nc [tvbondar@vbox]:~/.config/sway/config.d
95-system-keyboard-config.conf [M] 06.1.1 1+0 1/ 1 *(06 / 60b) -f0p>
exec_always /usr/libexec/sway-systemd/locale1.xb-config --oneshot
```

Рис. 6: Редактирование файла

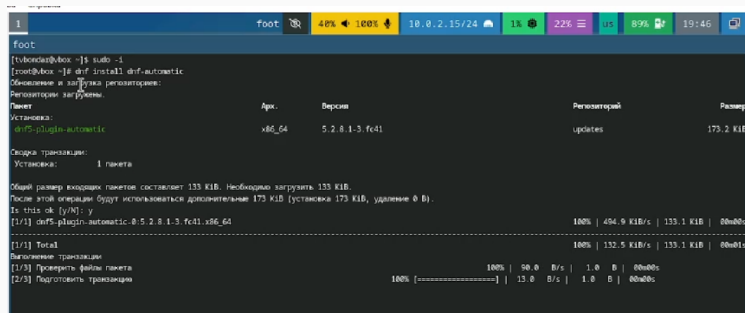
7. Отредактируем конфигурационный файл `/etc/X11/xorg.conf.d/00-keyboard.conf`.
Перезагрузим виртуальную машину.



```
mc [root@vbox]:/etc/X11/xorg.conf.d
00-keyboard.conf  [N] 82 L: [ 1+ 8 9/ 11] *(403 / 410b) 0034 0x022 [*]
# 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,en"
    Option "XkbVariant" "",wslkeys
    Option "XkbOptions" "grp:ctrl_toggle,compose:ralt,terminate:ctrl_alt_bksp"
EndSection
```

Рис. 7: Редактирование файла

8. Установим ПО для автоматического обновления



```
1 foot 48% 100% 10.0.2.15/24 1% 22% 115 89% 19:46

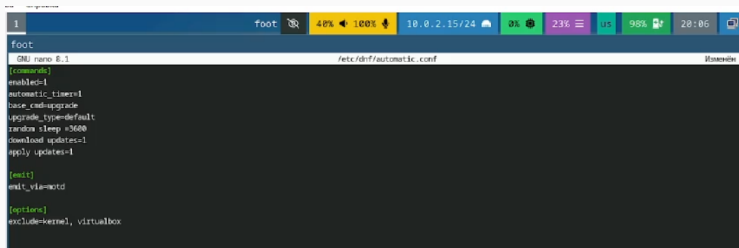
foot
[tvbondar@vbox ~]$ sudo -i
[root@vbox ~]# dnf install dnf-automatic
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет Арх. Версия Репозиторий Размер
Установка:
dnf5-plugin-automatic x86_64 5.2.8.1-3.fc41 updates 173.2 KiB

Сводка транзакции:
  Установка: 1 пакета

Общий размер входящих пакетов составляет 133 KiB. Необходимо загрузить 133 KiB.
После этой операции будут использоваться дополнительные 173 KiB (установка 173 KiB, удаление 0 B).
Is this ok [y/N]: y
[1/1] dnf5-plugin-automatic-0:5.2.8.1-3.fc41.x86_64 100% | 494.9 KiB/s | 133.1 KiB | 00m00s
-----
[1/1] Total 100% | 132.5 KiB/s | 133.1 KiB | 00m01s
Выполнение транзакции
[1/3] Проверить файлы пакета 100% | 90.0 B/s | 1.0 B | 00m00s
[2/3] Подготовить транзакция 100% [=====] | 13.0 B/s | 1.0 B | 00m00s
```

Рис. 8: Установка ПО

9. Отредактируем необходимый конфигурационный файл. Запустим таймер.



```
1
foot 40% 100% 10.0.2.15/24 0% 23% 10.0.2.15 98% 20:06
GNU nano 8.1 /etc/default/automatic.conf
[commands]
enabled=1
automatic_timer=1
base_cmd=upgrade
upgrade_type=default
random_sleep=3600
download_updates=1
apply_updates=1

[exit]
exit_via=notd

[options]
exclude=kernel, virtualbox
```

Рис. 9: Редактирование файла

10. Скачиваем pandoc и pandoc-crossref из репозиториев Github.

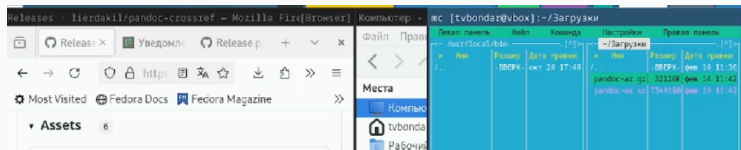
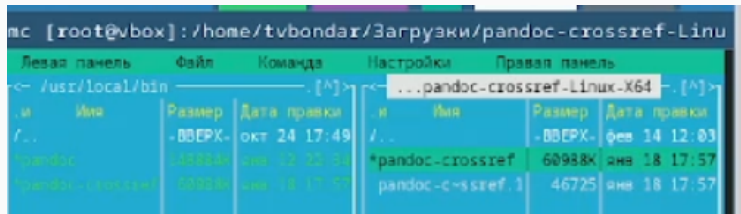


Рис. 10: Скачивание pandoc, pandoc-crossref

11. Переносим необходимые файлы из архивов в каталог `/usr/local/bin`.



12. Установим дистрибутив TexLive

```
[root@vbox ~]# nc

[root@vbox ~]# dnf -y install texlive-scheme-full
Обновление и загрузка репозитория:
Fedora 41 - x86_64 - Updates      100% | 2.7 KiB/s | 7.7 KiB | 00m03s
Fedora 41 - x86_64 - Updates      100% | 809.6 KiB/s | 2.3 MiB | 00m03s
```

Рис. 12: Установка TexLive

13. Посмотрим порядок загрузки системы с помощью команды dmesg.

```
[ 0.000000] Linux version 6.12.11-200.fc41.x86_64 (mockbuild@8c85b49e2e66068390f7ce0d84d4464) (gcc (GCC) 14.2.1 20250110 (Red Hat 14.2.1-7), GNU ld version 2.43.1-5.fc41) #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-0741-4044-9f71-0663b67cf033 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 0x0000000000000000-0x000000000009fbff] usable
[ 0.000000] BIOS-e820: [mem 0x000000000009fc00-0x00000000000fffff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000000fa000-0x00000000000fffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000100000-0x00000000003effff] usable
[ 0.000000] BIOS-e820: [mem 0x00000000003f0000-0x00000000003fffff] ACPI data
[ 0.000000] BIOS-e820: [mem 0x00000000fec00000-0x00000000fec0ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000fee00000-0x00000000fee0ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000fff00000-0x00000000ffffffff] 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] DMI: Memory slots populated: 0/0
[ 0.000000] Hypervisor detected: KVM
[ 0.000000] kvm-clock: Using msrc 4b564d01 and 4b564d00
[ 0.000000] kvm-clock: using sched offset of 9609018726 cycles
[ 0.000015] clocksource: kvm-clock: mask: 0xffffffffffffff max_cycles: 0x1cd42e8dffb, max_idle_ns: 881590591483 ns
[ 0.000025] tsc: Detected 2595.126 MHz processor
```

Рис. 13: Выполнение команды

14. Получаем информацию о версии ядра Linux, частоте процессора, модели процессора, объеме доступной оперативной памяти, типе обнаруженного гипервизора.

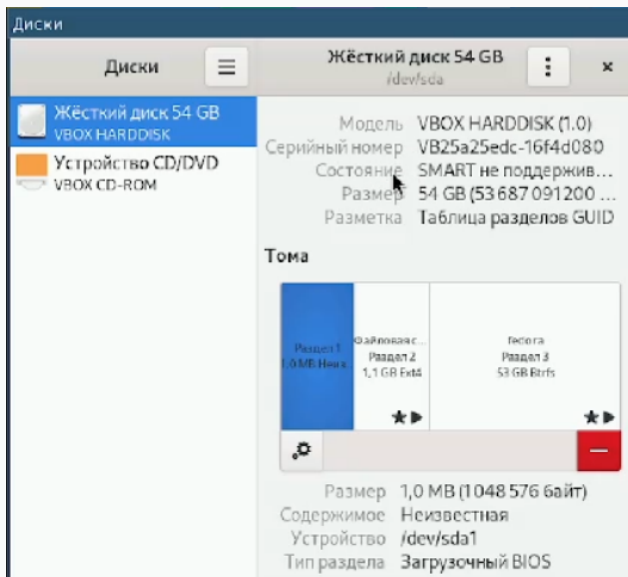
```
[rtubondar@vbox ~]$ sudo -i
[root@vbox ~]# dmesg |less
[root@vbox ~]#
[root@vbox ~]# dmesg |less
[root@vbox ~]# grep -i "Linux version"
[ 0.000000] Linux version 6.12.11-280.fc41 x86_64 (mockbuild@fc85b49e2e66468390f7ce4db4d4f454) (gcc (GCC) 14.2.1 20250110 (Red Hat 14.2.1-7), GNU ld version 2.43.1-5.fc41) #1 SMP PREEMPT_DYNAMIC Fri Jan 24 04:59:58 UTC 2025
[root@vbox ~]# dmesg | grep -i "processor"
[ 0.000015] tsc: Detected 2595.126 MHz processor
[ 0.225268] smpboot: Total of 2 processors activated (18380.58 BogoMIPS)
[ 0.255264] ACPI: Added _OSI(Processor Device)
[ 0.255268] ACPI: Added _OSI(Processor Aggregator Device)
[root@vbox ~]# dmesg | grep -i "CPU"
[ 0.213668] smpboot: CPU0: AMD Ryzen 3 3250U with Radeon Graphics (family: 0x17, model: 0x18, stepping: 0x1)
[root@vbox ~]# dmesg | grep -i "memory"
[ 0.000000] DMI: Memory slots populated: 0/8
[ 0.007332] ACPI: Reserving FACP table memory at [mem 0xa83f00f0-0xa83f81e3]
[ 0.007333] ACPI: Reserving DSDT table memory at [mem 0xa83f0618-0xa83f2962]
[ 0.007334] ACPI: Reserving FACS table memory at [mem 0xa83f0200-0xa83f823f]
[ 0.007335] ACPI: Reserving FACS table memory at [mem 0xa83f0200-0xa83f823f]
[ 0.007336] ACPI: Reserving APIC table memory at [mem 0xa83f0248-0xa83f829b]
[ 0.007337] ACPI: Reserving SSDT table memory at [mem 0xa83f02a8-0xa83f868b]
[ 0.007969] Early memory node ranges
[ 0.018401] PM: hibernation: Registered noave memory: [mem 0x00000000-0x00000fff]
[ 0.018403] PM: hibernation: Registered noave memory: [mem 0x0000f000-0x0000ffff]
[ 0.018404] PM: hibernation: Registered noave memory: [mem 0x00000000-0x0000ffff]
[ 0.018405] PM: hibernation: Registered noave memory: [mem 0x0000f000-0x0000ffff]
[ 0.018406] PM: hibernation: Registered noave memory: [mem 0xa83f0000-0xa83fffff]
[ 0.114900] Freeing SMP alternatives memory: 48K
[ 0.225650] Memory: 2477824K/262588K available (2152K kernel code, 5429K rodata, 1678K rodata, 488K init, 471K bss, 148552K reserved, 0K cma-reserved)
```

Рис. 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. 15: Получение необходимой информации ч.2

15. Получаем информацию о последовательности монтирования файловых систем. Тип файловой системы - Ext4.



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.

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