

Отчет по лабораторной работе №1

Операционные системы

Ицков Андрей Станиславович.

Российский университет дружбы народов, Москва, Россия

Информация

- Ицков Андрей Станиславович
- НКАбд-02-2024 № Студенческого билета: 1132246737
- Российский университет дружбы народов
- https://github.com/a-its/study_2024-2025_os-intro

[]

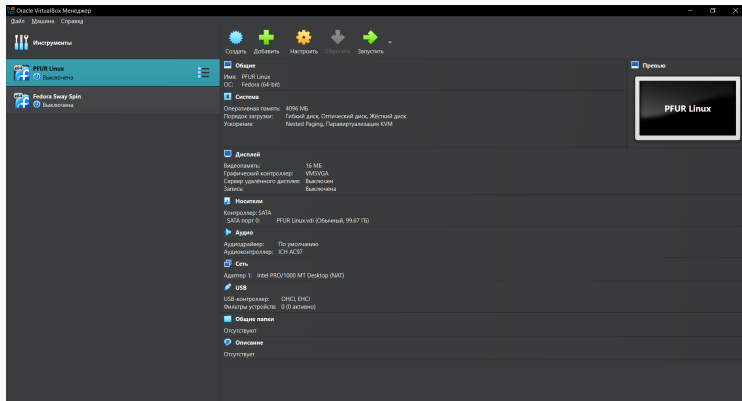
Целью данной работы является приобретение практических навыков установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов.

- 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. Создадим новую виртуальную машину, указав имя, размер основной памяти, размер видеопамяти, размер диска и других параметров на свое усмотрение, выбираем образ системы Fedora.



2. Начнем установку операционной системы, внеся перед этим необходимые для этого данные.

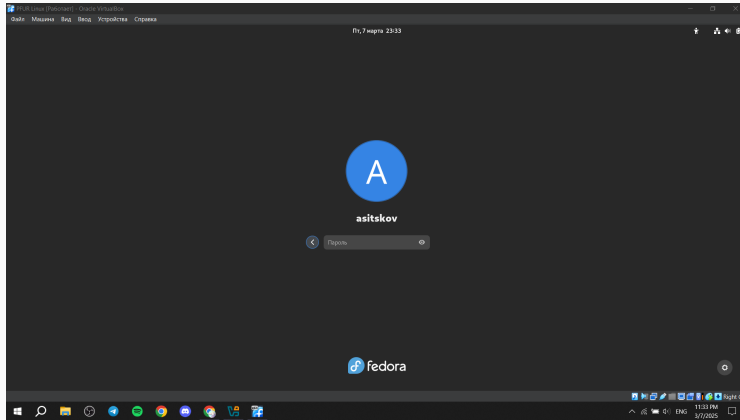
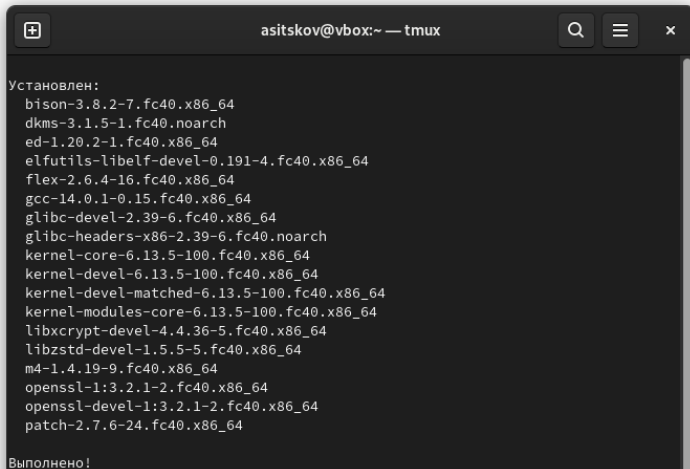


Рис. 2: Установка ОС

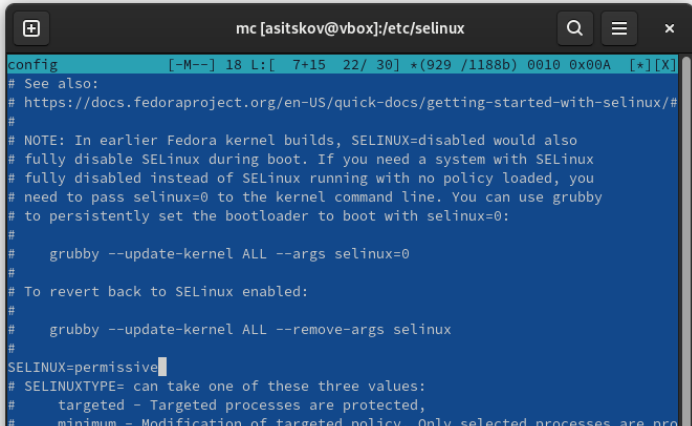
3. Войдем в ОС под своей учетной записью. В терминале через роль суперпользователя производим установку обновлений.



```
asitskov@vbox:~ — tmux
Установлен:
bison-3.8.2-7.fc40.x86_64
dkms-3.1.5-1.fc40.noarch
ed-1.20.2-1.fc40.x86_64
elfutils-libelf-devel-0.191-4.fc40.x86_64
flex-2.6.4-16.fc40.x86_64
gcc-14.0.1-0.15.fc40.x86_64
glibc-devel-2.39-6.fc40.x86_64
glibc-headers-x86-2.39-6.fc40.noarch
kernel-core-6.13.5-100.fc40.x86_64
kernel-devel-6.13.5-100.fc40.x86_64
kernel-devel-matched-6.13.5-100.fc40.x86_64
kernel-modules-core-6.13.5-100.fc40.x86_64
libxcrypt-devel-4.4.36-5.fc40.x86_64
libzstd-devel-1.5.5-5.fc40.x86_64
m4-1.4.19-9.fc40.x86_64
openssl-1:3.2.1-2.fc40.x86_64
openssl-devel-1:3.2.1-2.fc40.x86_64
patch-2.7.6-24.fc40.x86_64
Выполнено!
```

Повышение удобства работы. Отключение SELinux

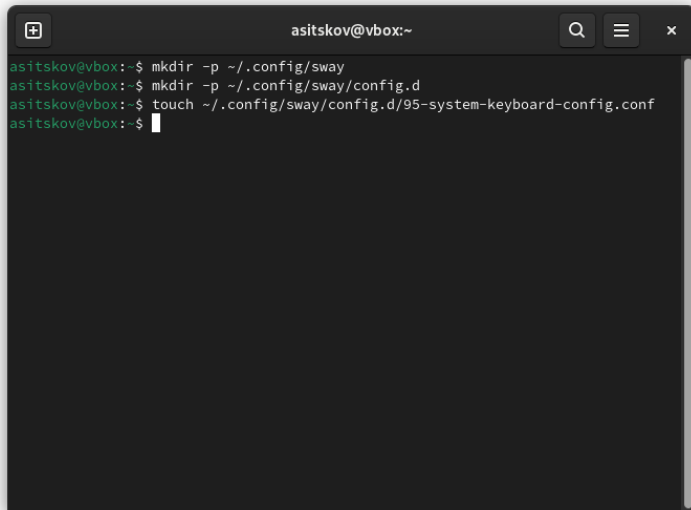
4. Установим программу tmux. Запустим ее, затем через команду mc в терминале заходим в требуемый файл и отключаем SELinux, заменив в файле значение enforcing на permissive. Перезапустим виртуальную машину.



The screenshot shows a terminal window with the title bar "mc [asitskov@vbox]:/etc/selinux". The terminal displays the contents of the `/etc/selinux/config` file, which is being edited with the `mc` (nano) editor. The file content includes instructions on how to disable SELinux by setting `SELINUX=permissive` and using `grubby` to update the kernel arguments. The current line being edited is `SELINUX=permissive`, with the cursor at the end of the line.

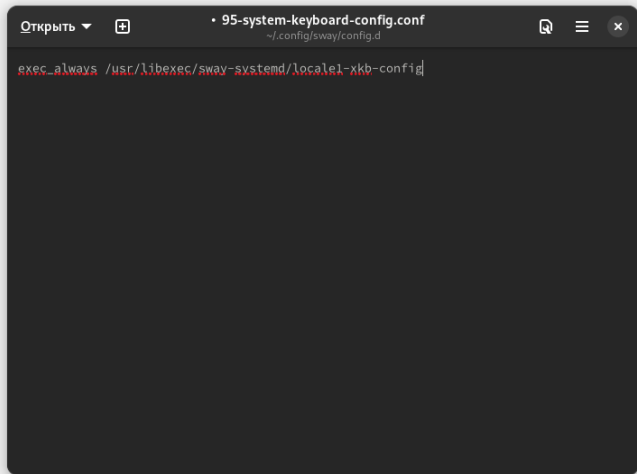
```
mc [asitskov@vbox]:/etc/selinux
config      [-M--] 18 L:[ 7+15 22/ 30] *(929 /1188b) 0010 0x00A  [*][X]
# See also:
# https://docs.fedoraproject.org/en-US/quick-docs/getting-started-with-selinux/#
#
# NOTE: In earlier Fedora kernel builds, SELINUX=disabled would also
# fully disable SELinux during 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
```

5. Создадим конфиг файл.



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

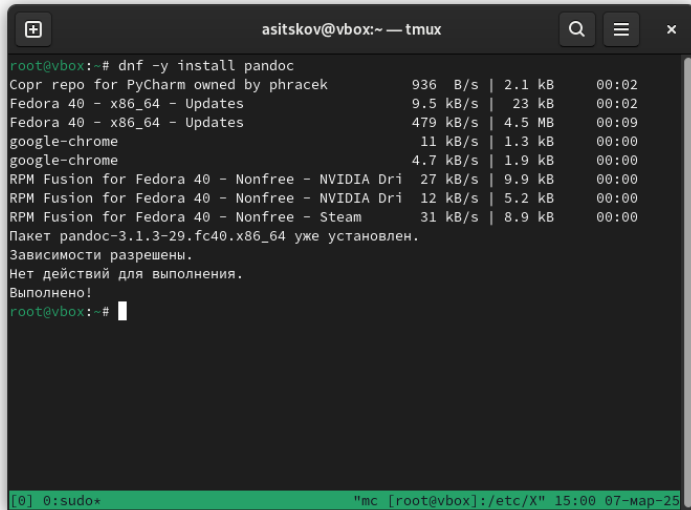
6. Отредактируем этот файл, подбирая значения под себя. Затем отредактируем еще один файл (/etc/X11/xorg.conf.d/00keyboard.conf) и перезагрузим машину.



The image shows a terminal window with a dark background. The title bar at the top reads "95-system-keyboard-config.conf" and " ~/.config/sway/config.d". Below the title bar, the command `exec_always /usr/libexec/sway-systemd/locale1-xkb-config` is entered. The text "exec_always" is highlighted in red, and the rest of the command is in white. The cursor is at the end of the command.

7. Устанавливаем ПО для автообновления. Снова редактируем конфигурационный файл, запускаем таймер.

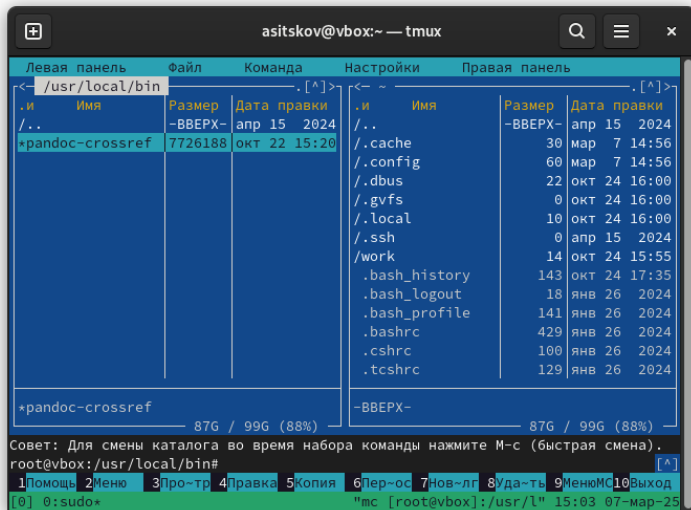
8. Скачаем pandoc и pandoc-crossref из репозитория Гитхаб.



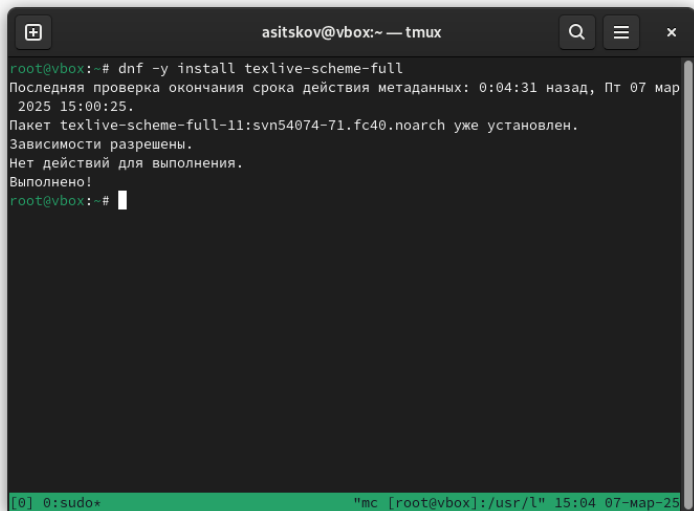
```
asitskov@vbox:~ — tmux
root@vbox:~# dnf -y install pandoc
Copr repo for PyCharm owned by phracek          936 B/s | 2.1 kB      00:02
Fedora 40 - x86_64 - Updates                    9.5 kB/s | 23 kB      00:02
Fedora 40 - x86_64 - Updates                   479 kB/s | 4.5 MB      00:09
google-chrome                                  11 kB/s | 1.3 kB      00:00
google-chrome                                  4.7 kB/s | 1.9 kB      00:00
RPM Fusion for Fedora 40 - Nonfree - NVIDIA Dri 27 kB/s | 9.9 kB      00:00
RPM Fusion for Fedora 40 - Nonfree - NVIDIA Dri 12 kB/s | 5.2 kB      00:00
RPM Fusion for Fedora 40 - Nonfree - Steam      31 kB/s | 8.9 kB      00:00
Пакет pandoc-3.1.3-29.fc40.x86_64 уже установлен.
Зависимости разрешены.
Нет действий для выполнения.
Выполнено!
root@vbox:~#
```

[0] 0:sudo+ "mc [root@vbox]:/etc/X" 15:00 07-мар-25

9. Перенесем необходимые файлы в необходимый каталог.



10. Установим дистрибутив TexLive.

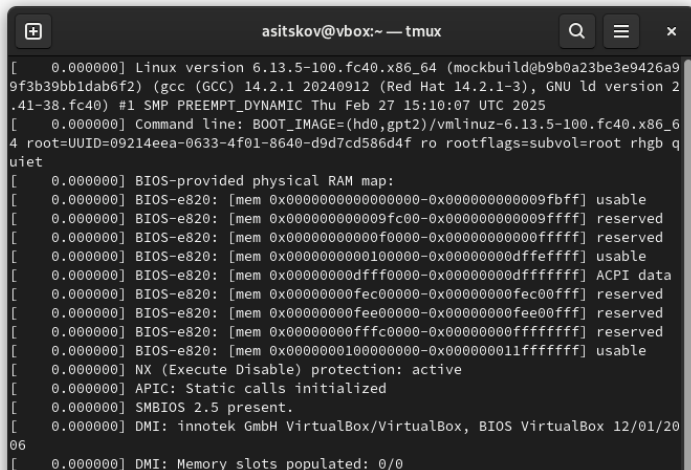


```
asitskov@vbox:~ — tmux
root@vbox:~# dnf -y install texlive-scheme-full
Последняя проверка окончания срока действия метаданных: 0:04:31 назад, Пт 07 мар 2025 15:00:25.
Пакет texlive-scheme-full-11:svn54074-71.fc40.noarch уже установлен.
Зависимости разрешены.
Нет действий для выполнения.
Выполнено!
root@vbox:~#
```

[0] 0:sudo* "mc [root@vbox]:/usr/l" 15:04 07-мар-25

Рис. 10: Установка программы TexLive

- Посмотрим порядок загрузки системы с помощью команды `dmesg`, получим необходимую информацию.



```
asitskov@vbox:~ — tmux
[ 0.000000] Linux version 6.13.5-100.fc40.x86_64 (mockbuild@b9b0a23be3e9426a9
9f3b39bb1dab6f2) (gcc (GCC) 14.2.1 20240912 (Red Hat 14.2.1-3), GNU ld version 2
.41-38.fc40) #1 SMP PREEMPT_DYNAMIC Thu Feb 27 15:10:07 UTC 2025
[ 0.000000] Command line: BOOT_IMAGE=(hd0,gpt2)/vmlinuz-6.13.5-100.fc40.x86_6
4 root=UUID=09214eea-0633-4f01-8640-d9d7cd586d4f ro rootflags=subvol=root rhgb q
uiet
[ 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 0x0000000000100000-0x0000000000dfffff] usable
[ 0.000000] BIOS-e820: [mem 0x0000000000dfff0000-0x0000000000dfffffff] 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 0x0000000100000000-0x000000011fffffffff] 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 12/01/20
06
[ 0.000000] DMI: Memory slots populated: 0/0
```

```
root@vbox:~  
root@vbox:~# dmesg | grep -i "CPU0"  
[ 0.412507] smpboot: CPU0: 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz (family: 0x6, model: 0x8c, stepping: 0x1)  
root@vbox:~# dmesg | grep -i "Memory available"  
root@vbox:~# dmesg | grep -i "Hypervisor detected"  
[ 0.000000] Hypervisor detected: KVM  
root@vbox:~# dmesg | grep -i "processor"  
[ 0.000009] tsc: Detected 2419.200 MHz processor  
[ 0.413935] smpboot: Total of 1 processors activated (4838.40 BogoMIPS)  
[ 0.430764] ACPI: Added _OSI(Processor Device)  
[ 0.430765] ACPI: Added _OSI(Processor Aggregator Device)  
root@vbox:~# dmesg | grep -i "Memory"  
[ 0.000000] DMI: Memory slots populated: 0/0  
[ 0.016287] ACPI: Reserving FACP table memory at [mem 0xdfff00f0-0xdfff01e3]  
[ 0.016289] ACPI: Reserving DSDT table memory at [mem 0xdfff0610-0xdfff2962]  
[ 0.016289] ACPI: Reserving FACS table memory at [mem 0xdfff0200-0xdfff023f]  
[ 0.016290] ACPI: Reserving FACS table memory at [mem 0xdfff0200-0xdfff023f]  
[ 0.016290] ACPI: Reserving APIC table memory at [mem 0xdfff0240-0xdfff0293]  
[ 0.016291] ACPI: Reserving SSDT table memory at [mem 0xdfff02a0-0xdfff060b]  
[ 0.019627] Early memory node ranges  
[ 0.139998] PM: hibernation: Registered nosave memory: [mem 0x00000000-0x00000fff]  
[ 0.140000] PM: hibernation: Registered nosave memory: [mem 0x0009f000-0x0009ffff]
```

Рис. 12: Получение необходимой информации

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. ∴