

# Лабораторная работа №1

---

Абронина А. К.

Российский университет дружбы народов, Москва, Россия

## Докладчик

- Абронина Алиса Кирилловна
- НКАбд-01-24, с/б 1132246717
- Российский университет дружбы народов
- [https://github.com/akabronina/study\\_2024-2025\\_os-intro](https://github.com/akabronina/study_2024-2025_os-intro)

... ..

# Выполнение лабораторной работы №1

## Цель

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

## Задание

1. Создание виртуальной машины
2. Установка операционной системы
3. Работа с операционной системой после установки.
4. Установка программного обеспечения для создания документации
5. Домашнее задание

## Теоритическое введение

Операционная система - это комплекс взаимосвязанных программ, который действует

## 6. Устанавливаю программы для удобства работы в консоли.

```
[root@vbox ~]# sudo dnf -y install tmux mc
Обновление и загрузка репозитория:
Репозитории загружены.
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:
rpm-libs                             x86_64    1.20.7-48.fc41                       fedora                      27.7 KiB

Сводка транзакции:
Установка:      2 пакетов

Общий размер входящих пакетов составляет 2 MiB. Необходимо загрузить 2 MiB.
После этой операции будут использоваться дополнительные 7 MiB (установка 7 MiB, удаление 0 B).
[1/2] rpm-libs-0:1.20.7-48.fc41.x86_64                                100% | 22.5 KiB/s | 20.2 KiB | 00m01s
[2/2] mc-1:4.8.32-1.fc41.x86_64                                       100% | 862.6 KiB/s | 1.9 MiB | 00m02s
-----
[2/2] Total                                                            100% | 284.5 KiB/s | 2.0 MiB | 00m07s
Выполнение транзакции
[1/4] Проверить файлы пакета                                           100% | 105.0 B/s | 2.0 B | 00m00s
[2/4] Подготовить транзакцию                                           100% | 4.0 B/s | 2.0 B | 00m00s
[3/4] Установка rpm-libs-0:1.20.7-48.fc41.x86_64                    100% | 467.7 KiB/s | 28.5 KiB | 00m00s
[4/4] Установка mc-1:4.8.32-1.fc41.x86_64                            100% [*****] | 30.2 MiB/s | 7.2 MiB | 00m00s
>>> Выполняется trigger-install ckpmtner: glibc-common-0:2.40-3.fc41.x86_64
warning: 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
[4/4] Установка mc-1:4.8.32-1.fc41.x86_64                            100% | 4.1 MiB/s | 7.2 MiB | 00m02s
Завершено!
```

Рис. 4: Установка tmux и mc

## 7. Устанавливаю программы для автоматического обновления.

```
root@vbox ~]# sudo dnf -y install dnf-automatic
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет
```

	Арх.	Версия	Репозиторий	Размер
Установка:				
dnf-plugin-automatic	x86_64	5.2.10.0-2.fc41	updates	178.6 KiB

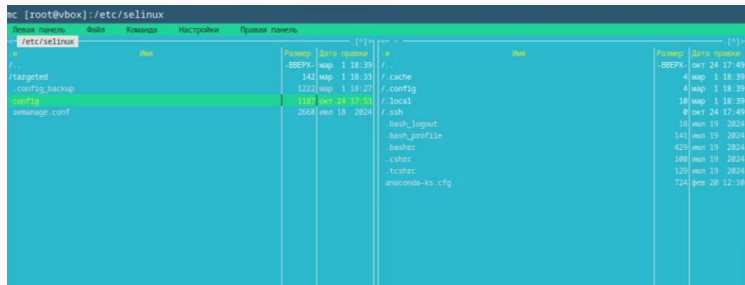
```
Вводка транзакции:
Установка: 1 пакета

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
-----
[1/1] Total
-----
100% | 57.7 KiB/s | 141.3 KiB | 00m02s
100% | 25.3 KiB/s | 141.3 KiB | 00m00s

Выполнение транзакции
1/3] Проверить файлы пакета
2/3] Подготовить транзакцию
3/3] Установка dnf5-plugin-automatic-0:5.2.10.0-2.fc41.x86_64
>>> Выполняется trigger-install скриптов: glibc-common-0:2.40-3.fc41.x86_64
warning: posix.fork(): .fork(), .exec(), .wait() and .redirect2null() are deprecated, use rpm.spawn() or rpm.execute() instead
warning: posix.exec(): .fork(), .exec(), .wait() and .redirect2null() are deprecated, use rpm.spawn() or rpm.execute() instead
3/3] Установка dnf5-plugin-automatic-0:5.2.10.0-2.fc41.x86_64
100% | 60.4 KiB/s | 180.7 KiB | 00m03s
завершено!
```

Рис. 5: Установка программы для автоматического обновления

8. Перехожу в каталог /etc/selinux, открываю тс, ищу нужный файл.

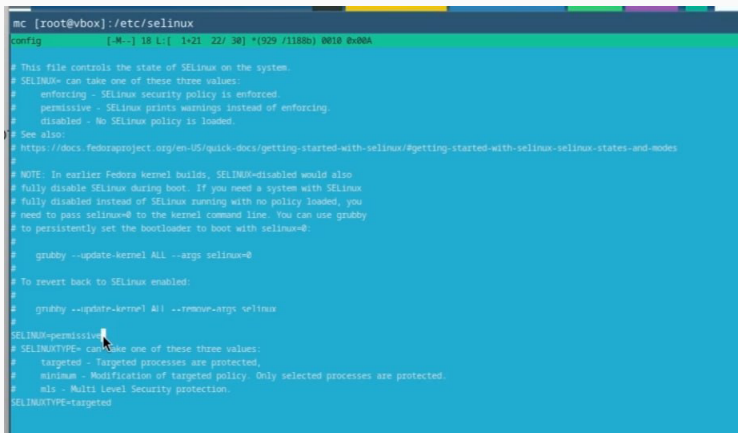


Terminal window showing the command `ls [root@vbox]:/etc/selinux`. The output is a table of files and directories in the `/etc/selinux` directory.

Имя	Размер	Дата правки	Имя	Размер	Дата правки
./	-ВВЕРХ-	мар 1 18:39	./	-ВВЕРХ-	окт 24 17:49
/targeted	142	мар 1 18:33	./cache	4	мар 1 18:39
./config_backup	1222	мар 1 18:27	./config	4	мар 1 18:39
<b>config</b>	<b>1187</b>	<b>окт 24 17:53</b>	./local	16	мар 1 18:39
semanage.conf	2668	июл 18 2024	./ssh	0	окт 24 17:49
			./bash_logout	18	июл 19 2024
			./bash_profile	141	июл 19 2024
			./bashrc	429	июл 19 2024
			./cshrc	100	июл 19 2024
			./tcshrc	129	июл 19 2024
			anaconda-ks.cfg	724	фев 28 12:18

Рис. 6: Поиск файла

## 9. Изменяю открытый файл.

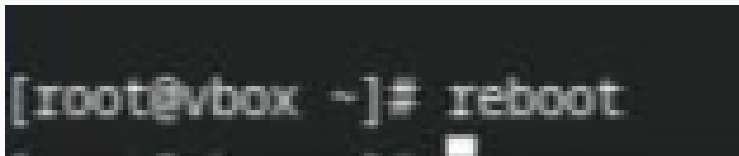


```
mc [root@vbox]:/etc/selinux
config      [-M--] 18 L:[ 1+21 22/ 30] *(929 /1188b) 0010 0x00A

# 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=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. 7: Изменяю файл

10. Перезагружаю виртуальную машину.



```
[root@vbox ~]# reboot
```

**Рис. 8:** Перезагрузка виртуальной машины



11. Перехожу в папку /tc/X11/xorg.conf.d, открываю tc для удобства и открываю файл 00-keyboard.conf.

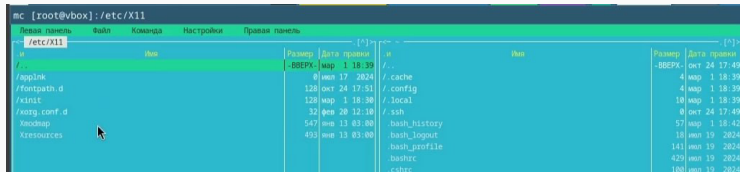


Рис. 9: Поиск файла

## 12. Редактирую файл конфигурации.

```
mc [root@vbox]:/etc/X11/xorg.conf.d
00-keyboard.conf  [-----] 32 L:[ 1+ 7  8/ 12] *(314 / 410b) 0034 0x022
# Written by systemd-localed(8), 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,ru"
    Option "XkbModel" "pc105"
    Option "XkbVariant" ","
    Option "XkbOptions" "grp:alt_shift_toggle"
EndSection
```

**Рис. 10:** Редактирование файла

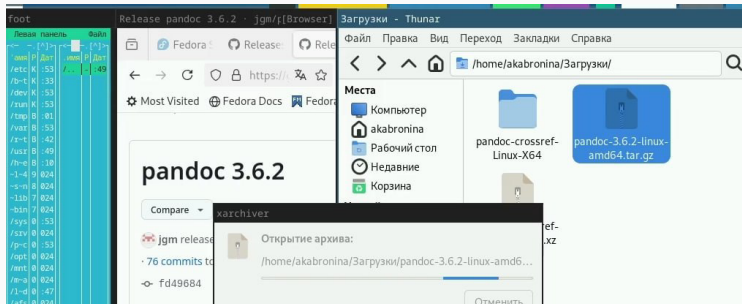
### 13.Перезагрузка виртуальной машины.



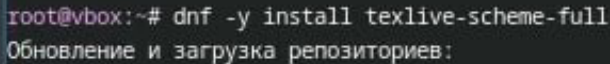
Рис. 11: Перезагрузка виртуальной машины

### Установка программного обеспечения для создания документации

#### 14. Устанавливаю pandoc и pandoc-crossref вручную.



15. Устанавливаю дистрибутив texlive.

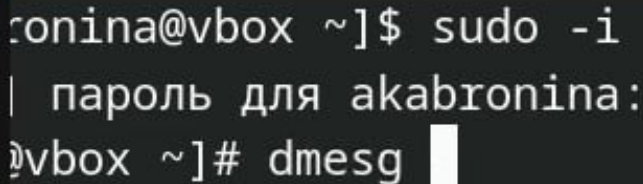


```
root@vbox:~# dnf -y install texlive-scheme-full
Обновление и загрузка репозитория:
```

Рис. 13: Установка texlive

### Выполнение домашнего задания

16. Ввожу в терминале dmesg, чтобы проанализировать последовательность загрузки системы.



```
akabronina@vbox ~]$ sudo -i
| пароль для akabronina:
akabronina@vbox ~]$ dmesg
```

## 18. Ищу частоту процессора.

```
[root@vbox ~]# dmesg | grep -i "processor"
[ 0.00011] tsc: Detected 1899.953 MHz processor
[ 0.517875] smpboot: Total of 1 processors activated (3799.90 BogoMIPS)
[ 0.528952] ACPI: Added _OSI(Processor Device)
[ 0.528956] ACPI: Added _OSI(Processor Aggregator Device)
```

Рис. 16: Поиск частоты процессора

## 19. Ищу модель процессора.

```
[root@vbox ~]# dmesg | grep -i "CPU0"
[ 0.515167] smpboot: CPU0: AMD Ryzen 7 5800U with Radeon Graphics (family:
0x19, model: 0x50, stepping: 0x0)
```

Рис. 17: Поиск модели процессора

## 20. Ищу объем доступной оперативной памяти.

```
/box ~]# dmesg | grep -i "memory"
000000] DMI: Memory slots populated: 0/0
002461] ACPI: Reserving FACP table memory at [mem 0xdfff00f0-0xdfff01e3]
002462] ACPI: Reserving DSDT table memory at [mem 0xdfff0c10-0xdfff0c21]
```

21. Ищу тип обнаруженного гипервизора.

```
[root@vbox ~]# dmesg | grep -i "Hypervisor detected"
[    0.000000] Hypervisor detected: KVM
```

**Рис. 19:** Поиск типа обнаруженного гипервизора

22. Смотрю тип файловой системы.

```
[root@vbox ~]# fdisk -l
Disk /dev/sda: 80 GiB, 85899345920 bytes, 167772160 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 8A82D17D-056C-4266-BC92-0D47C1664BF5

Device            Start      End          Sectors    Size Type
/dev/sda1         2048       4095          2048      1M BIOS boot
/dev/sda2         4096    2101247    2097152      1G Linux extended boot
/dev/sda3    2101248 167770111 165668864    79G Linux filesystem
```

Я приобрела практические навыки установки операционной системы на виртуальную машину, а также настроила минимально необходимые для дальнейшей работы сервисы.