

# **Отчёт по лабораторной работе №1**

**Установка операционной системы Linux, дистрибутив Fedora Sway на виртуальную машину**

Спелов Андрей Николаевич НПИбд-02-23

# Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Домашнее задание	12
4	Выводы	14
	Список литературы	15

## Список иллюстраций

2.1	Начало создания виртуальной машины . . . . .	6
2.2	Первоначальная настройка виртуальной машины . . . . .	7
2.3	Создаем виртуальный жесткий диск . . . . .	7
2.4	Выбираем систему в качестве оптического привода для дальнейшей установки . . . . .	8
2.5	Заходим на машину и вводим необходимые команды для начала установки системы . . . . .	8
2.6	Начинаем выбирать нужную конфигурацию будущей системы . .	9
2.7	Выбираем нужную конфигурацию . . . . .	9
2.8	Удаляем загрузочный диск . . . . .	10
2.9	Снова заходим в систему . . . . .	10
2.10	Используем команду <code>sudo dnf -y update</code> . . . . .	11
2.11	Устанавливаем необходимый софт . . . . .	11
2.12	Устанавливаем язык разметки Markdown . . . . .	11
3.1	Используем команду <code>dmesg   grep -i "linux version"</code> . . . . .	12
3.2	Используем команду <code>dmesg   grep -i "MHz"</code> . . . . .	12
3.3	Используем команду <code>dmesg   grep -i "CPU0"</code> . . . . .	12
3.4	Используем команду <code>free -m</code> . . . . .	12
3.5	Используем команду <code>dmesg   grep -i "hypervisor detected"</code> . . . . .	13
3.6	Используем команду <code>dfindmnt</code> . . . . .	13
3.7	Используем команду <code>dmesg   grep -i "mount"</code> . . . . .	13

## **Список таблиц**

# 1 Цель работы

Приобрести практические навыки установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов.

## 2 Выполнение лабораторной работы

На заранее установленный Virtual Box начинаем процесс установки виртуальной машины(называем машину, ставим тип системы) (рис. 2.1).

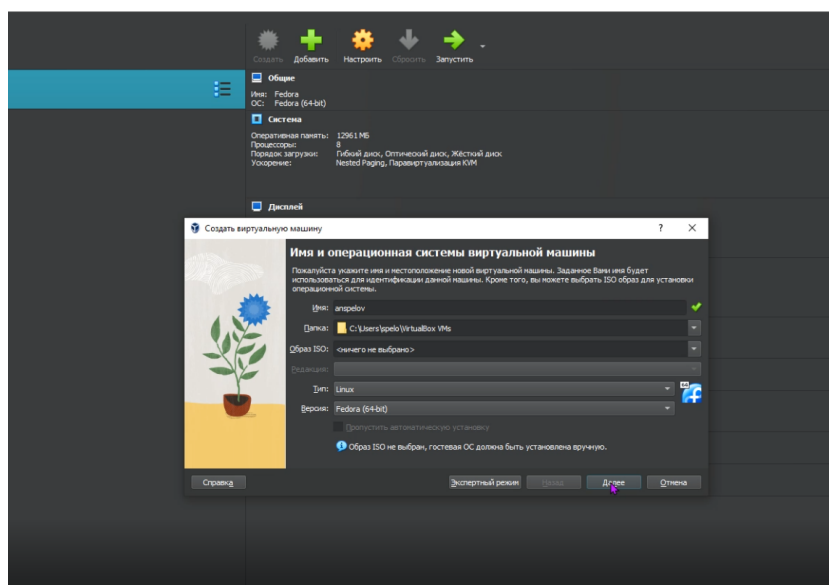


Рис. 2.1: Начало создания виртуальной машины

Ставим необходимое для работы количество памяти и ядер процессора (рис. 2.2).

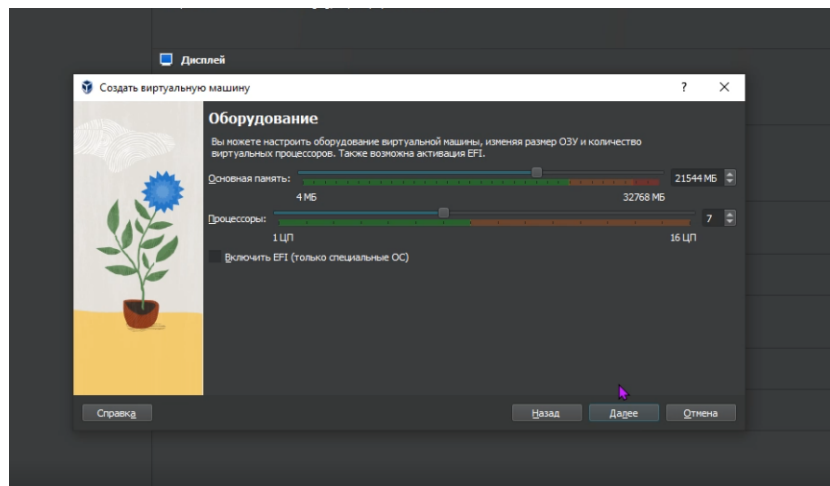


Рис. 2.2: Первоначальная настройка виртуальной машины

Ставим необходимое количество памяти жесткого диска на виртуальную машину(80+ Гб) (рис. 2.3).

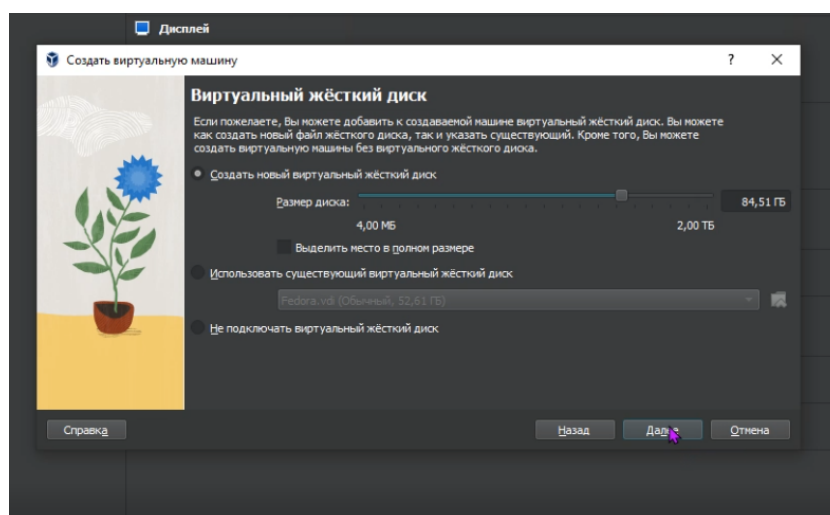


Рис. 2.3: Создаем виртуальный жесткий диск

Закончив настройку виртуальной машины в меню Virtual Box выбираем заранее установленную систему Fedora Sway (рис. 2.4).

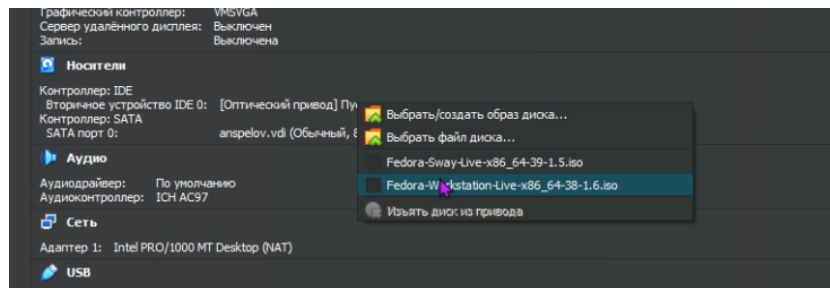


Рис. 2.4: Выбираем систему в качестве оптического привода для дальнейшей установки

Заходим на виртуальную машину (рис. 2.5).

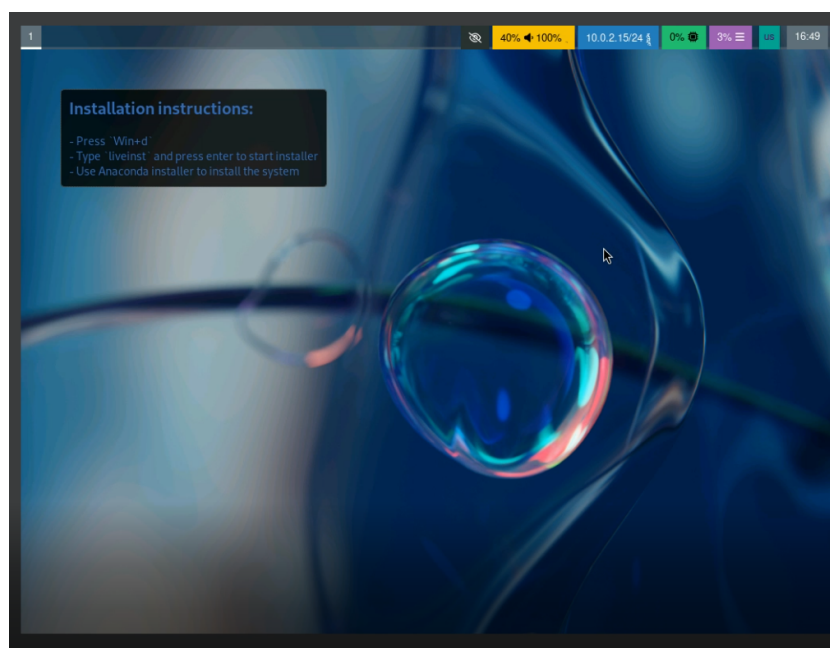


Рис. 2.5: Заходим на машину и вводим необходимые команды для начала установки системы

Заходим в меню установки системы (рис. 2.6).



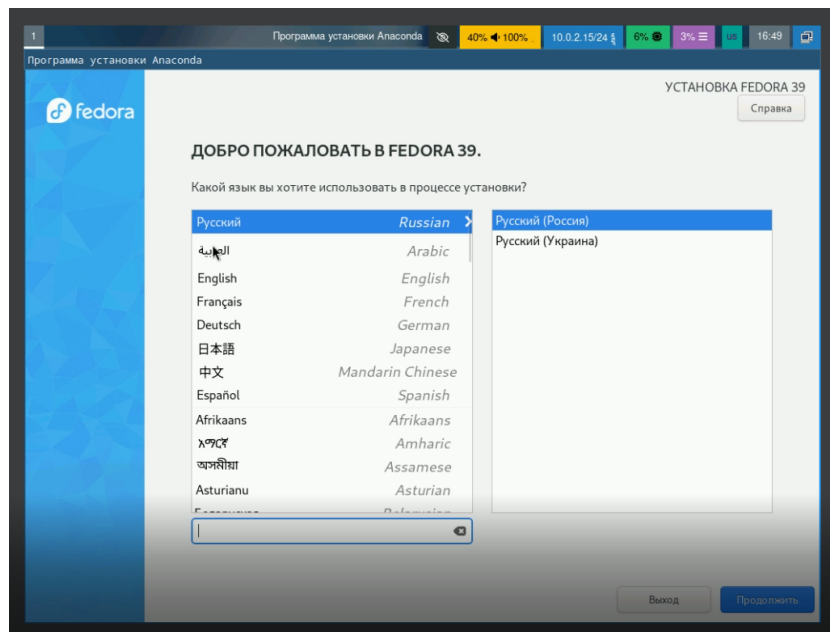


Рис. 2.6: Начинаем выбирать нужную конфигурацию будущей системы

Выбираем все необходимые параметры(раскладку клавиатуры, супер пользователя root) и начинаем установку (рис. 2.7).

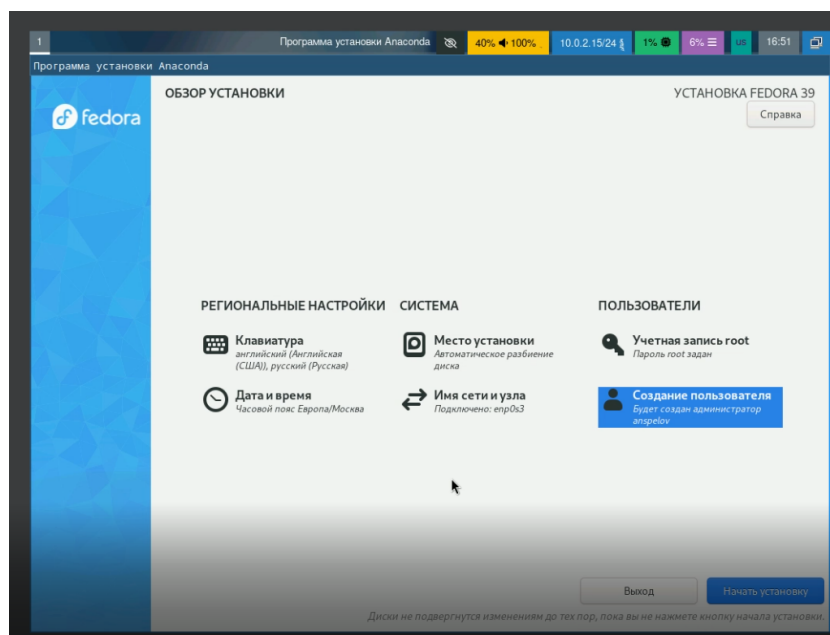


Рис. 2.7: Выбираем нужную конфигурацию

После установки системы закрываем машину и убираем загрузочный диск

(рис. 2.8).

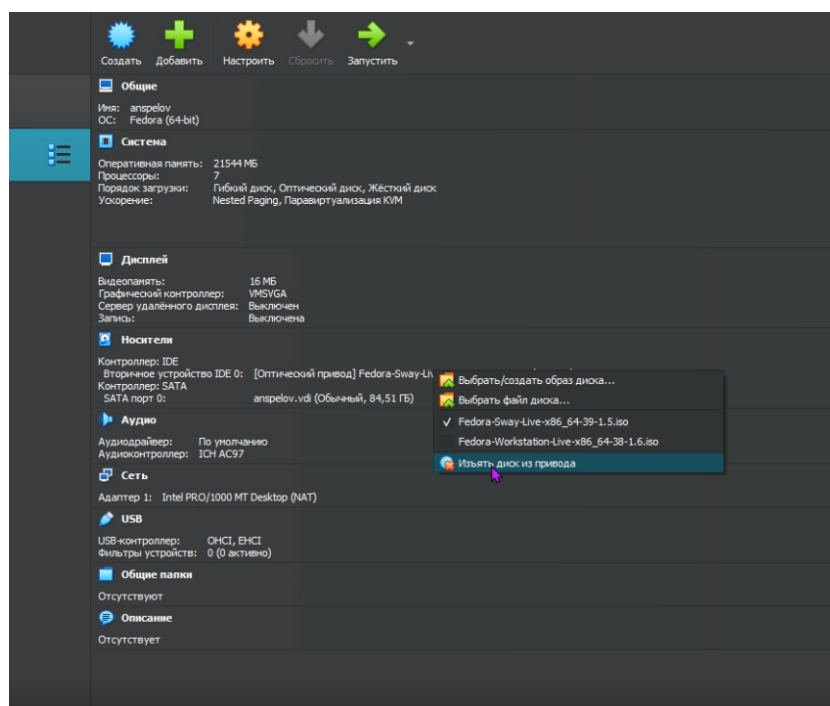


Рис. 2.8: Удаляем загрузочный диск

Запускаем виртуальную машину (рис. 2.9).

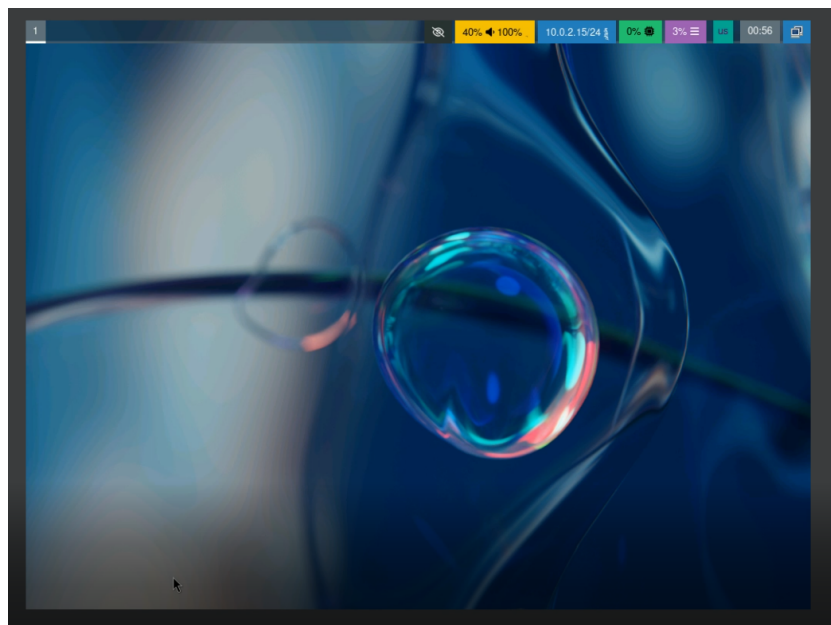
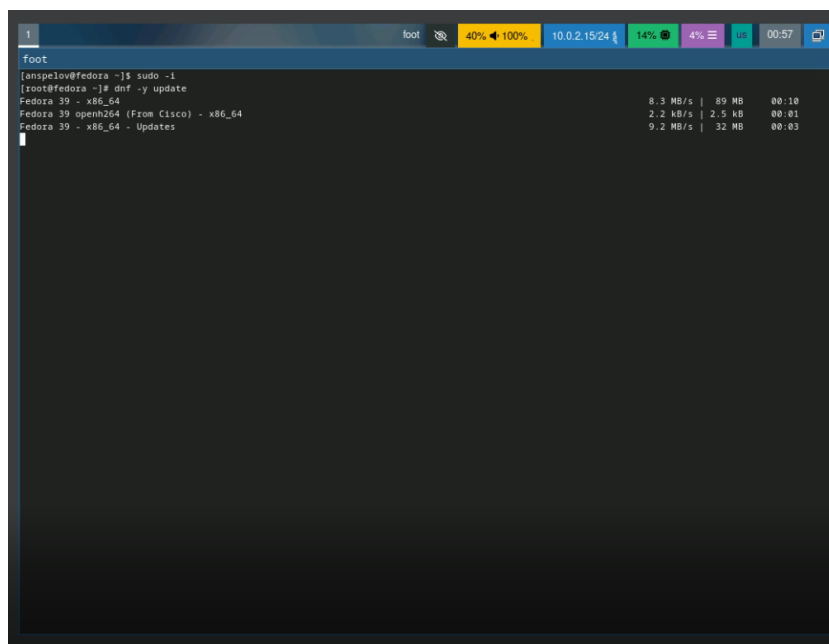


Рис. 2.9: Снова заходим в систему

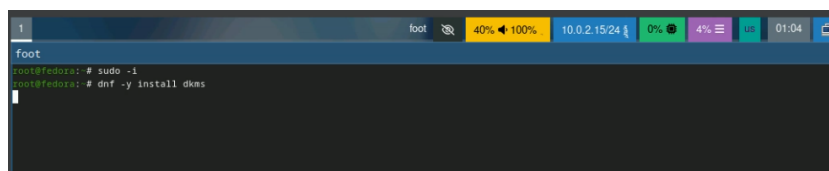
Заходим в терминал и скачиваем все необходимые обновления системы (рис. 2.10).



```
foot
[anspelov@fedora ~]$ sudo -i
[root@fedora ~]# dnf -y update
Fedora 39 - x86_64                                8.3 MB/s | 89 MB | 00:10
Fedora 39 openh264 (From Cisco) - x86_64         2.2 kB/s | 2.5 kB | 00:01
Fedora 39 - x86_64 - Updates                      9.2 MB/s | 32 MB | 00:03
```

Рис. 2.10: Используем команду `sudo dnf -y update`

Устанавливаем tmux командой `dnf -y install tmux mc`, устанавливаем пакет DKMS (рис. 2.11).



```
foot
[root@fedora: ~]# sudo -i
[root@fedora: ~]# dnf -y install dkms
```

Рис. 2.11: Устанавливаем необходимый софт

Так же устанавливаем pandoc командой `dnf -y install pandoc` и TexLive командой `dnf -y install texlive-scheme-full` (рис. 2.12).



```
foot
[root@fedora: ~]# dnf -y install texlive-scheme-full
```

Рис. 2.12: Устанавливаем язык разметки Markdown

### 3 Домашнее задание

Узнаем Версию ядра Linux (рис. 3.1).

```
[anspelov@fedora ~]$ dmesg | grep -i "linux version"
[ 0.000000] Linux version 6.7.4-200.fc39.x86_64 (mockbuild@de0c58eb5f524c20963d3b29334043cc) (gcc (GCC) 13.2.1 20231205 (Red Hat 13.2.1-6), GNU ld version 2.40-14.fc39) #1 SMP PREEMPT_DYNAMIC Mon Feb 5 22:21:14 UTC 2024
[anspelov@fedora ~]$
```

Рис. 3.1: Используем команду `dmesg | grep -i "linux version"`

Узнаем Частоту процессора (рис. 3.2).

```
[anspelov@fedora ~]$ dmesg | grep -i "MHz"
[ 0.000000] tsc: Detected 2495.998 MHz processor
[ 3.541004] e1000 0000:00:03:0 eth0: (PCI:33MHz:32-bit) 08:00:27:e6:af:4d
[anspelov@fedora ~]$
```

Рис. 3.2: Используем команду `dmesg | grep -i "MHz"`

Узнаем Модель процессора (рис. 3.3).

```
[anspelov@fedora ~]$ dmesg | grep -i "CPU0"
[ 0.710656] smpboot: CPU0: 11th Gen Intel(R) Core(TM) i7-11700F @ 2.50GHz (family: 0x6, model: 0xa7, stepping: 0x1)
[anspelov@fedora ~]$
```

Рис. 3.3: Используем команду `dmesg | grep -i "CPU0"`

Узнаем Объём доступной оперативной памяти (рис. 3.4).

```
[anspelov@fedora ~]$ free -m
              total        used         free      shared  buff/cache   available
Mem:           21041         760       19864         24         732       20281
Swap:           8191           0         8191
[anspelov@fedora ~]$
```

Рис. 3.4: Используем команду `free -m`

Узнаем Тип обнаруженного гипервизора (рис. 3.5).

```
[anspelov@fedora ~]$ dmesg | grep -i "hypervisor detected"
[    0.000000] Hypervisor detected: KVM
[anspelov@fedora ~]$
```

Рис. 3.5: Используем команду `dmesg | grep -i "hypervisor detected"`

Узнаем Тип файловой системы корневого раздела (рис. 3.6).

```
[anspelov@fedora ~]$ findmnt
TARGET SOURCE FSTYPE OPTIONS
-/dev /dev/sda3[/root] btrfs rw,relatime,seclabel,compress=zstd:1,space_cache=v2,subvol=257,subvol=/root
-/dev/mqueue devtmpfs devtmpfs rw,nosuid,seclabel,size=4096k,nr_inodes=2688144,mode=755,inode64
-/dev/hugepages hugetlbfs hugetlbfs rw,nosuid,nodev,relatime,seclabel,pagesize=2M
-/dev/shm tmpfs tmpfs rw,nosuid,nodev,seclabel,inode64
-/dev/pts devpts devpts rw,nosuid,exec,relatime,seclabel,gid=5,mode=620,ptmxmode=000
-/sys sysfs sysfs rw,nosuid,nodev,noexec,relatime,seclabel
-/sys/fs/selinux selinuxfs selinuxfs rw,nosuid,noexec,relatime
-/sys/kernel/debug debugfs debugfs rw,nosuid,nodev,noexec,relatime,seclabel
-/sys/kernel/tracing tracefs tracefs rw,nosuid,nodev,noexec,relatime,seclabel
-/sys/fs/fuse/connections fusectl fusectl rw,nosuid,nodev,noexec,relatime
-/sys/kernel/security securityfs securityfs rw,nosuid,nodev,noexec,relatime
-/sys/fs/cgroup cgroup2 cgroup2 rw,nosuid,nodev,noexec,relatime,seclabel,nsdelegate,memory_recursiveprot
-/sys/fs/pstore pstore pstore rw,nosuid,nodev,noexec,relatime,seclabel
-/sys/fs/bpf bpf bpf rw,nosuid,nodev,noexec,relatime,mode=700
-/sys/kernel/config configfs configfs rw,nosuid,nodev,noexec,relatime
-/proc proc proc rw,nosuid,nodev,noexec,relatime
-/proc/sys/fs/binfmt_misc systemd-1 proc rw,relatime,fd=34,pgpr=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=7352
-/run tmpfs tmpfs rw,nosuid,nodev,seclabel,size=4309368k,nr_inodes=819200,mode=755,inode64
  └─/run/user/1000 tmpfs rw,nosuid,nodev,relatime,seclabel,size=2154680k,nr_inodes=538670,mode=700,uid=1000,gid=
    └─/run/user/1000/doc portal fuse.portal rw,nosuid,nodev,relatime,user_id=1000,group_id=1000
-/tmp tmpfs tmpfs rw,nosuid,nodev,seclabel,size=10772410k,nr_inodes=1048576,inode64
-/home /dev/sda3[/home] btrfs rw,relatime,seclabel,compress=zstd:1,space_cache=v2,subvol=256,subvol=/home
-/boot /dev/sda2 ext4 rw,relatime,seclabel
-/var/lib/nfs/rpc_pipefs sunrpc rpc_pipefs rw,relatime
```

Рис. 3.6: Используем команду `dfindmnt`

Узнаем Последовательность монтирования файловых систем (рис. 3.7).

```
[anspelov@fedora ~]$ dmesg | grep -i "mount"
[    0.608612] Mount: cache hash table entries: 65536 (order: 7, 524288 bytes, linear)
[    0.608612] Mountpoint-cache hash table entries: 65536 (order: 7, 524288 bytes, linear)
[    3.264704] BTRFS: device label fedora devid 1 transid 113 /dev/sda3 scanned by mount (529)
[    3.269987] BTRFS info (device sda3): first mount of filesystem 5d127d1b-c0af-4782-9949-d728a635285c
[    5.612843] systemd[1]: Set up automount proc-sys-fs-binfmt_misc.automount - Arbitrary Executable File Formats File System Automount Point.
[    5.633143] systemd[1]: Mounting dev-hugepages.mount - Huge Pages File System.
[    5.635148] systemd[1]: Mounting dev-mqueue.mount - POSIX Message Queue File System.
[    5.636332] systemd[1]: Mounting sys-kernel-debug.mount - Kernel Debug File System.
[    5.637419] systemd[1]: Mounting sys-kernel-tracing.mount - Kernel Trace File System.
[    5.659808] systemd[1]: Starting systemd-remount-fs.service - Remount Root and Kernel File Systems.
[    5.671095] systemd[1]: Mounted dev-hugepages.mount - Huge Pages File System.
[    5.673898] systemd[1]: Mounted dev-mqueue.mount - POSIX Message Queue File System.
[    5.674034] systemd[1]: Mounted sys-kernel-debug.mount - Kernel Debug File System.
[    5.674140] systemd[1]: Mounted sys-kernel-tracing.mount - Kernel Trace File System.
[    6.614054] EXT4-fs (sda2): mounted filesystem 1b32bee9-21a9-4b10-8092-77d96e9f72c5 r/w with ordered data mode. Quota mode: none.
```

Рис. 3.7: Используем команду `dmesg | grep -i "mount"`

## **4 Выводы**

Мы приобрели практические навыки установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов.

## **Список литературы**