

# **Отчет по лабораторной работе №2**

**Дисциплина: архитектура компьютера**

Сокирка Анна Константиновна

# Содержание

<b>1</b>	<b>Содержание</b>	<b>6</b>
<b>2</b>	<b>Цель работы</b>	<b>7</b>
<b>3</b>	<b>Задание</b>	<b>8</b>
<b>4</b>	<b>Теоретическое введение</b>	<b>9</b>
<b>5</b>	<b>Выполнение лабораторной работы</b>	<b>12</b>
<b>6</b>	<b>Выводы</b>	<b>37</b>
<b>7</b>	<b>Список литературы</b>	<b>38</b>

# Список иллюстраций

5.1	Открытие терминала . . . . .	12
5.2	Вывод команды pwd . . . . .	13
5.3	Перемещение по директориям . . . . .	13
5.4	Перемещение по директориям . . . . .	14
5.5	Перемещение по директориям . . . . .	14
5.6	Перемещение по директориям . . . . .	15
5.7	Вывод всех файлов домашнего каталога . . . . .	15
5.8	Домашний каталог . . . . .	16
5.9	Вывод файлов директории Документы . . . . .	16
5.10	Окно графического файлового менеджера . . . . .	17
5.11	Список файлов каталога /usr/local . . . . .	17
5.12	Пример использования ключей . . . . .	18
5.13	Пример использования ключей . . . . .	18
5.14	Создание каталога . . . . .	19
5.15	Каталог dir . . . . .	19
5.16	Перемещение в каталог и создание в нем каталогов . . . . .	20
5.17	Создание каталога из другой директории; Проверка работы команд . . . . .	20
5.18	создание иерархической цепочки подкаталогов . . . . .	21
5.19	Создание файла . . . . .	22
5.20	Удаление файла с запросом подтверждения . . . . .	23
5.21	Рекурсивное удаление директорий . . . . .	24
5.22	Проверка правильности выполнения команд . . . . .	24
5.23	Создание новых директорий . . . . .	25
5.24	Создание файла . . . . .	26
5.25	Создание файла . . . . .	26
5.26	Перемещение файла . . . . .	27
5.27	Копирование файла . . . . .	27
5.28	Проверка работы команды . . . . .	28
5.29	Копирование и перемещение файлов . . . . .	28
5.30	Перемещение по директориям . . . . .	28
5.31	Переименование каталога . . . . .	29
5.32	Чтение файла . . . . .	29
5.33	Путь к домашнему каталогу . . . . .	29
5.34	Ввод команд . . . . .	30
5.35	Содержимое корневого каталога . . . . .	30
5.36	Содержимое домашнего каталога . . . . .	31
5.37	Содержимое каталога /etc . . . . .	31

5.38 Содержимое каталога /usr/local . . . . .	31
5.39 Создание директорий . . . . .	32
5.40 Создание файлов в каталоге temp . . . . .	32
5.41 Файлы в labs . . . . .	32
5.42 Рисунок 5.42 . . . . .	33
5.43 Рисунок . . . . .	33
5.44 Рисунок 5.44 . . . . .	34
5.45 Рисунок 5.45 . . . . .	34
5.46 Создание файлов в каталоге temp . . . . .	35
5.47 Рисунок 5.47 . . . . .	35
5.48 Рисунок 5.48 . . . . .	35
5.49 Рисунок 5.49 . . . . .	35
5.50 Рисунок 5.50 . . . . .	36
5.51 Рисунок 5.51 . . . . .	36

## **Список таблиц**

# 1 Содержание

1. Цель работы
2. Задание
3. Теоретическое введение
4. Выполнение лабораторной работы
5. Выводы
6. Источники

## 2 Цель работы

Целью данной работы является приобретение практических навыков работы с операционной системой на уровне командной строки (организация файловой системы, навигация по файловой системе, создание и удаление файлов и директорий). Цель данного шаблона — максимально упростить подготовку отчётов по лабораторным работам. Модифицируя данный шаблон, студенты смогут без труда подготовить отчёт по лабораторным работам, а также познакомиться с основными возможностями разметки Markdown.

## 3 Задание

1. Перемещение по файловой системе.
2. Создание пустых каталогов и файлов.
3. Перемещение и удаление файлов или каталогов.
4. Команда cat: вывод содержимого файлов.
5. Выполнение заданий для самостоятельной работы



## 4 Теоретическое введение

GNU Linux — семейство переносимых, многозадачных и многопользовательских операционных систем, на базе ядра Linux, включающих тот или иной набор утилит и программ проекта GNU, и, возможно, другие компоненты. Как и ядро Linux, системы на его основе, как правило, создаются и распространяются в соответствии с моделью разработки свободного и открытого программного обеспечения (Open-Source Software). Дистрибутив GNU Linux — общее определение ОС, использующих ядро Linux и набор библиотек и утилит, выпускаемых в рамках проекта GNU, а также графическую оконную подсистему X Window System. Дистрибутив готов для конечной установки на пользовательское оборудование. Кроме ядра и, собственно, операционной системы дистрибутивы обычно содержат широкий набор приложений, таких как редакторы документов и таблиц, мультимедийные проигрыватели, системы для работы с базами данных и т.д. Существуют дистрибутивы, разрабатываемые как при коммерческой поддержке (Red Hat / Fedora, SLED / OpenSUSE, Ubuntu), так и исключительно усилиями добровольцев (Debian, Slackware, Gentoo, ArchLinux). В GNU Linux доступ пользователя к командной оболочке обеспечивается через терминал (или консоль). Запуск терминала можно осуществить через главное меню Приложения Стандартные Терминал (или Консоль) или нажав `Ctrl + Alt + t`. Интерфейс командной оболочки очень прост. Обычно он состоит из приглашения к командной строке (строки, оканчивающейся символом `)`, `: iivanova@dk4n31 :` Это приглашение командной оболочки, которое несёт в себе информацию об имени пользователя `iivanova`, имени компьютера `dk4n31` и текущем каталоге, в котором находится пользова-

тель, в данном случае это домашний каталог пользователя, обозначенный как ~. Команды могут быть использованы с ключами (или опциями) — указаниями, модифицирующими поведение команды. Ключи обычно начинаются с символа (-) или (—) и часто состоят из одной буквы. Кроме ключей после команды могут быть использованы аргументы (параметры) — названия объектов, для которых нужно выполнить команду (например, имена файлов и каталогов). Например, для подробного просмотра содержимого каталога documents может быть использована команда `ls -l`

ключом `-l`: `iivanova@dk4n31:~$ ls -l documents` Иногда в GNU Linux имена программ и команд слишком длинные, однако `bash` может завершать имена при их вводе в терминале. Нажав клавишу `Tab`, можно завершить имя команды, программы или каталога. Например, предположим, что нужно использовать программу `mcedit`. Для этого наберите в командной строке `mc`, затем нажмите один раз клавишу `Tab`. Если ничего не происходит, то это означает, что существует несколько возможных вариантов завершения команды. Нажав клавишу `Tab` ещё раз, можно получить список имён, начинающихся с `mc`: `iivanova@dk4n31:~$ mc`  
`mc mcd mcedit mclasser mcookie mcview` В ОС Linux каталог, который является “вершиной” файловой системы, называется корневым каталогом, обозначается символом `/` и содержит все остальные каталоги и файлы. Каталог Описание / Корневая директория, содержащая всю файловую `/bin` Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям (например: `cat`, `ls`, `cp`) `/etc` Общесистемные конфигурационные файлы и файлы конфигурации установленных программ `/home` Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя `/media` Точки монтирования для сменных носителей, таких как CD-ROM, DVD-ROM, flash `/root` Домашняя директория пользователя `root` `/tmp` Временные файлы Демидова А. В. 3 Архитектура ЭВМ `/usr` Вторичная иерархия для данных пользователя; содержит большинство пользовательских приложений и утилит, используемых в многопользователь-

ском режиме; может быть смонтирована по сети только для чтения и быть общей для нескольких машин. Существует несколько видов путей к файлу: • полный или абсолютный путь — начинается от корня (/), образуется перечислением всех каталогов, разделённых

прямым слешем (/), и завершается именем файла (напри мер, полный путь к файлу addition.txt из каталога user в каталоге home, находящемся в корневом каталоге, будет иметь вид: /home/user/documents/addition.txt; • относительный путь — так же как и полный путь, строится перечислением через (/) всех каталогов, но начинается от текущего каталога (каталога, в котором “находится” пользователь), т.е. пользователь, находясь в каталоге user, может обратиться к файлу addition.txt, указав относительный путь documents/addition.txt. Таким образом, в Linux если имя объекта начинается с /,то системой это интерпретируется как полный путь, в любом другом случае — как относительный. В Linux любой пользователь имеет домашний каталог, который, как правило, имеет имя пользователя. В домашних каталогах хранятся документы и настройки пользователя. Для обозначения домашнего каталога используется знак тильды (~). При переходе из домашнего каталога знак тильды будет заменён на имя нового текущего каталога. Основные команды взаимодействия пользователя с файловой системой

Команда	Описание
pwd	Print Working Directory определение текущего каталога
cd	Change Directory смена каталога
ls	LiSt вывод списка файлов
mkdir	MaKe DIRectory создание пустых каталогов
touch	создание пустых файлов
rm	ReMove удаление файлов или каталогов
mv	MoVe перемещение файлов и каталогов
cp	CoPy копирование файлов и каталогов
cat	вывод содержимого файлов

## 5 Выполнение лабораторной работы

### 1. Перемещение по файловой системе

Открываю терминал (рис. 5.1):

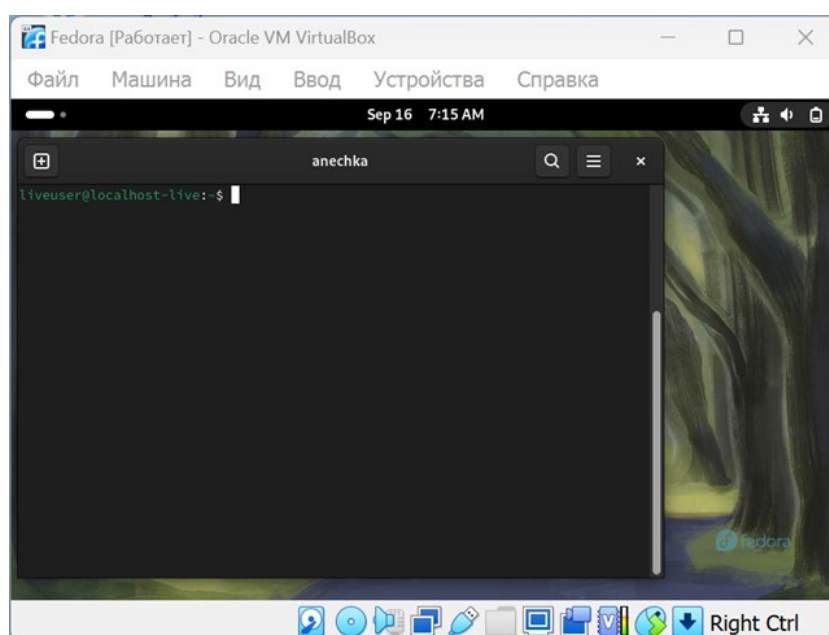


Рис. 5.1: Открытие терминала

Убедилась, что нахожусь в домашней директории, так как появился значок тильда около имени пользователя. Ввожу в терминале команду `pwd`, чтобы узнать полный путь к домашнему каталогу (рис. 5.2):

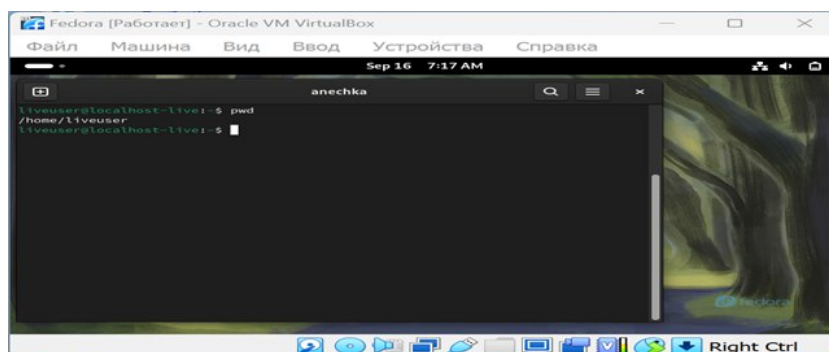


Рис. 5.2: Вывод команды pwd

С помощью команды `cd` указываю относительный путь к каталогу Documents и перемещаюсь в указанную директорию, т. к. Документы – директория внутри домашнего каталога (рис. 5.3):

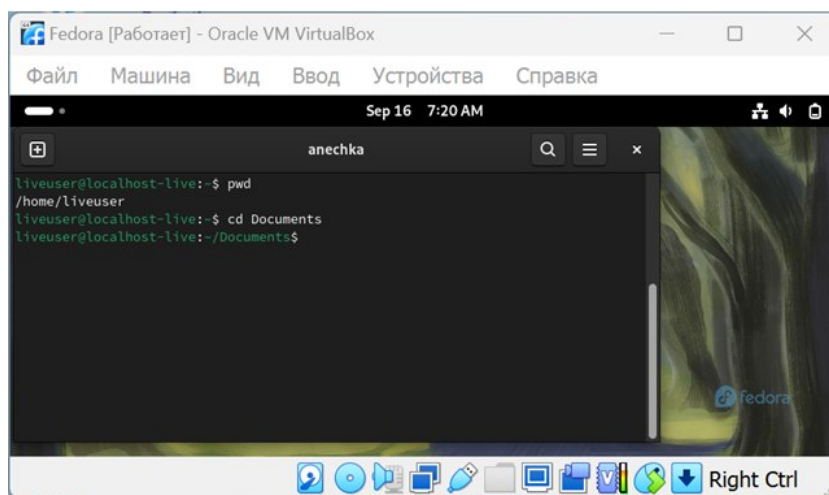


Рис. 5.3: Перемещение по директориям

Перейдя в каталог `local`, который является подкаталогом директории `usr`, находящийся в корневом каталоге. Для этого при написании команды указываю после утилиты `cd` абсолютный путь к нужному каталогу, начинающийся с корневого каталога «`/`» (рис. 5.4):

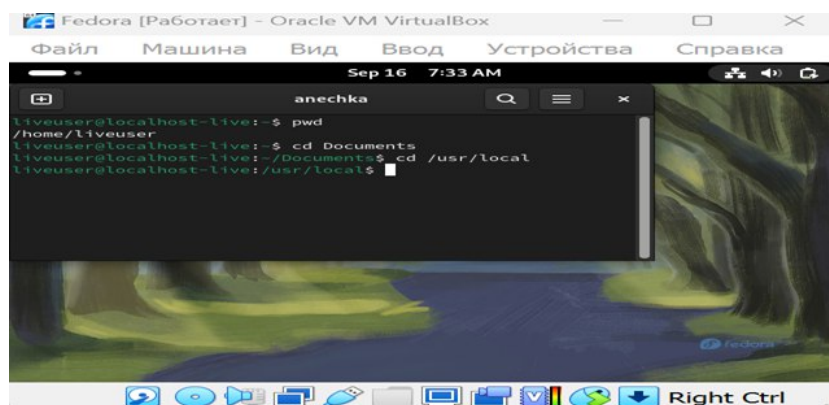


Рис. 5.4: Перемещение по директориям

Перехожу в последний каталог, в котором я была с помощью команды «cd –» (рис. 5.5):

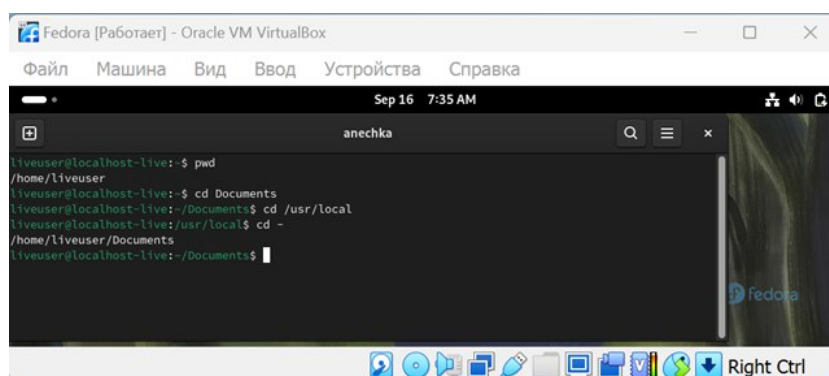


Рис. 5.5: Перемещение по директориям

Перехожу на один каталог выше по иерархии с помощью команды «cd ..» (рис. 5.6):

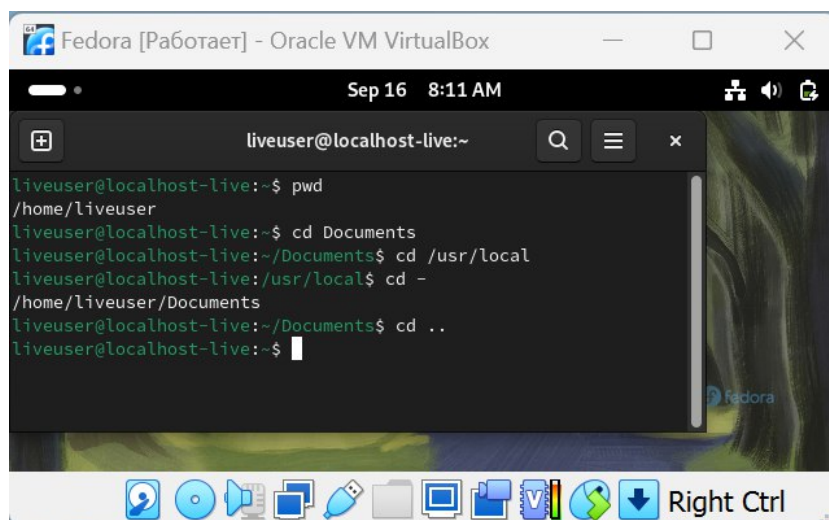


Рис. 5.6: Перемещение по директориям

Находясь в домашнем каталоге, вывожу директории домашнего каталога с помощью команды `ls`, которая выдает список файлов текущего каталога (рис. 5.7):

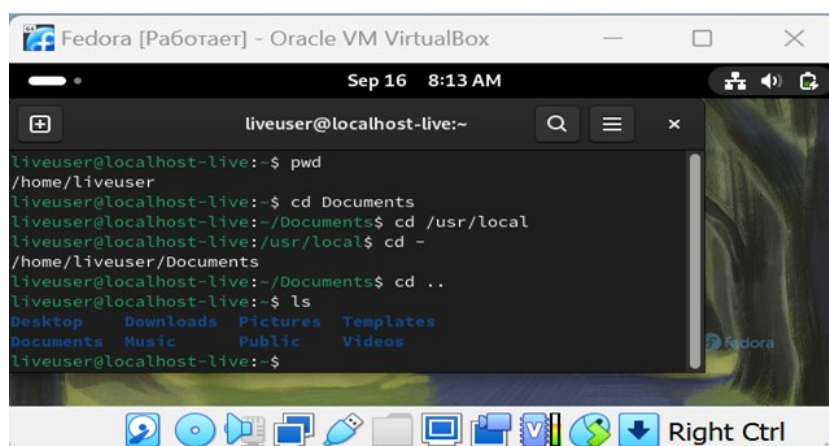


Рис. 5.7: Вывод всех файлов домашнего каталога

Открываю домашний каталог с помощью файлового менеджера графического окружения моей ОС (рис. 5.8):

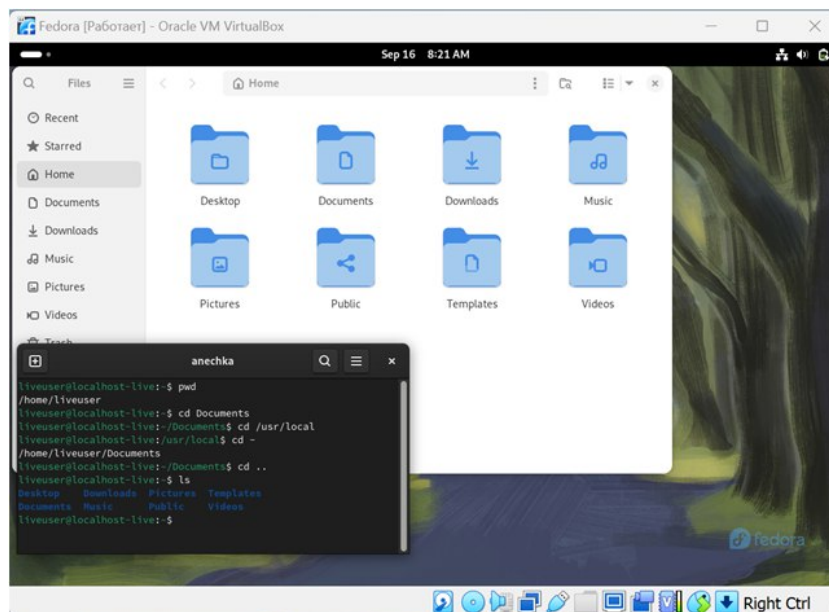


Рис. 5.8: Домашний каталог

Убеждаюсь, что список файлов, полученных с помощью команды ls, совпадает с файлами, отображающимися в графическом файловом менеджере. Выведу список файлов подкаталога Documents моего домашнего каталога, указав относительный путь (рис. 5.9):

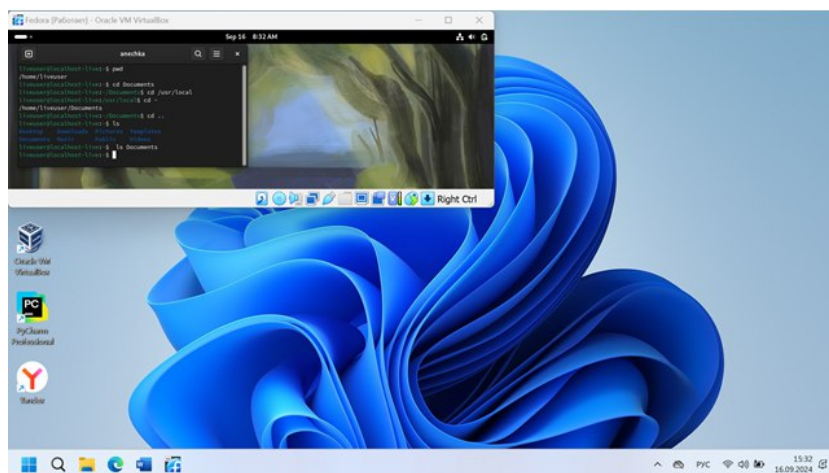


Рис. 5.9: Вывод файлов директории Документы

Так как в каталоге Documents нет файлов, вывод пустой, это мы так же можем проверить через графический файловый менеджер, выбрав в домашнем каталоге



директорию Documents (рис. 5.10):

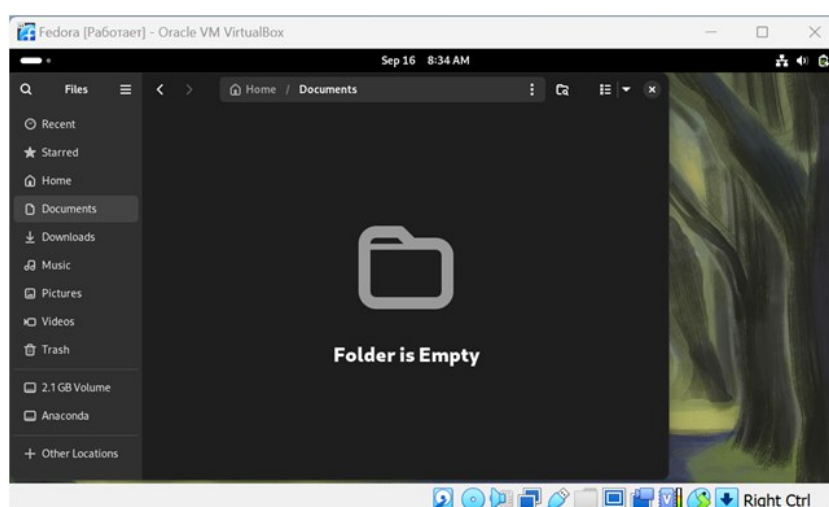


Рис. 5.10: Окно графического файлового менеджера

Вывожу список файлов каталога `/usr/local`, указав абсолютный путь к нему (рис. 5.11):

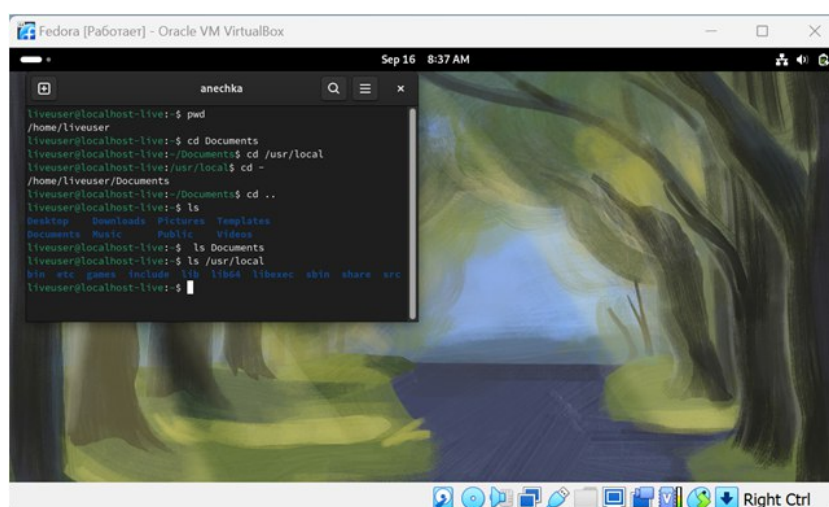


Рис. 5.11: Список файлов каталога `/usr/local`

Выведу список файлов каталога `/usr/local`, используя ключи `-la`, где `-l` – вывод дополнительной информации о файлах (права доступа, владельцы и группы, размеры файлов и время последнего доступа), `-a` – вывод списка файлов, включая скрытые файлы, в данном случае добавились директории `«.»` и `«..»`,

как скрытые (рис. 5.12):

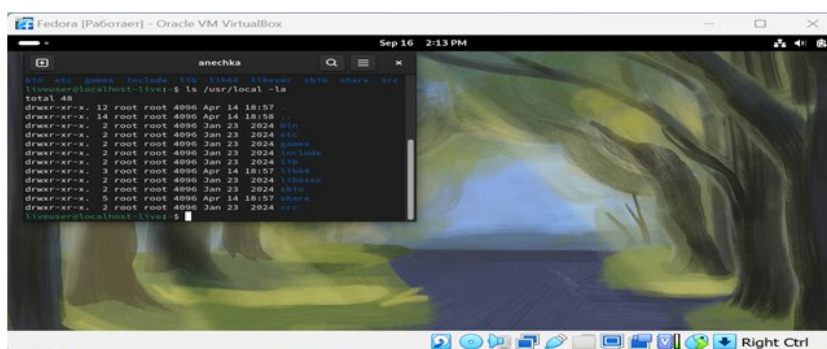


Рис. 5.12: Пример использования ключей

Используя ключ `-i`, благодаря которому осуществляется вывод уникального номера файла в файловой системе перед каждым файлом (рис. 5.13):

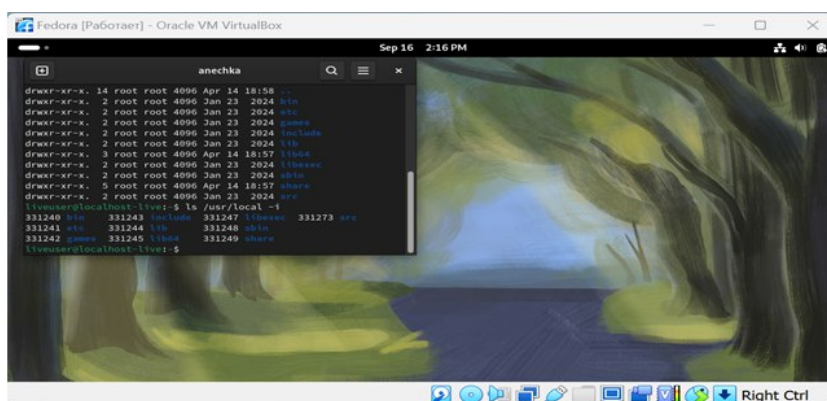


Рис. 5.13: Пример использования ключей

## 2.Создание пустых каталогов и файлов

Создам в домашнем каталоге подкаталог с именем `parentdir` с помощью команды `mkdir`. Благодаря команде `ls` проверяю, создан ли каталог. Да, директория `parentdir` находится в домашнем каталоге (рис. 5.14):

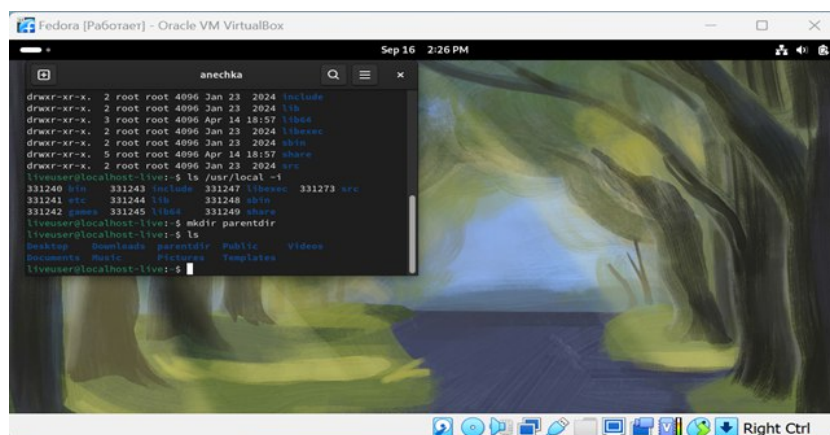


Рис. 5.14: Создание каталога

Создам подкаталог dir в только что созданном каталоге parentdir (рис. 5.15):

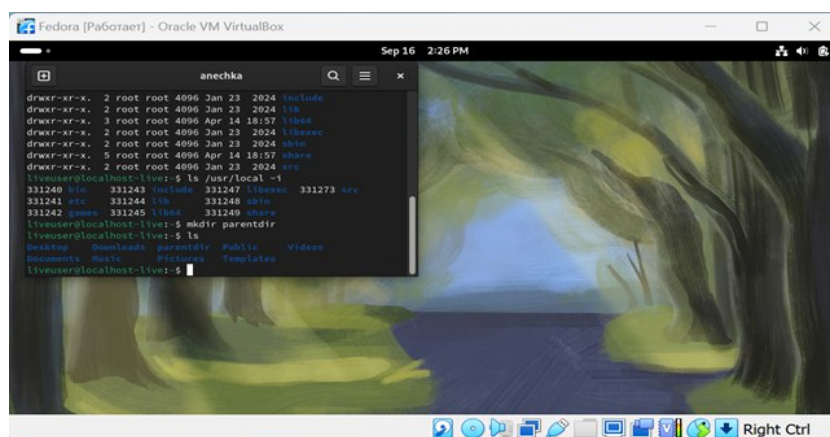


Рис. 5.15: Каталог dir

Перехожу в директорию parentdir, создаю в ней подкаталоги dir1, dir2, dir3, введя несколько аргументов для команды mkdir (рис. 5.16):

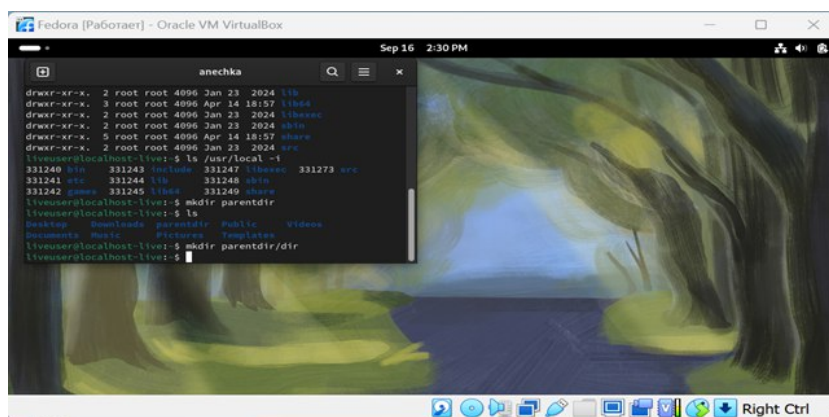


Рис. 5.16: Перемещение в каталог и создание в нем каталогов

Создам подкаталог в каталоге, отличном от текущего (сейчас я нахожусь в директории parentdir, а создавать подкаталог буду в домашней директории), для этого необходимо указать путь к месту создания подкаталога: `mkdir ~/newdir`, т. е. сначала домашнюю директорию, в которой буду создавать подкаталог, потом название создаваемого подкаталога. Командой «`ls ~`» проверяю, получилось ли создать подкаталог в домашнем каталоге (рис. 5.17):

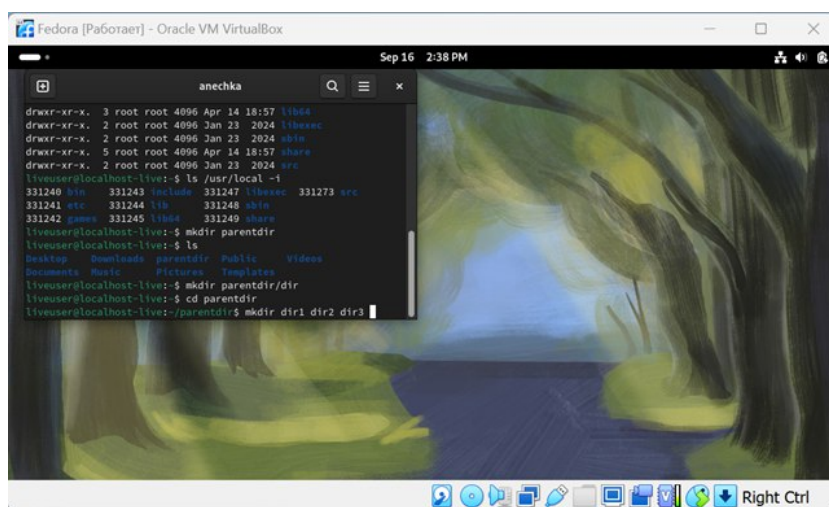


Рис. 5.17: Создание каталога из другой директории; Проверка работы команд

Создам иерархическую цепочку подкаталогов `newdir/dir1/dir2`, создавая все промежуточные каталоги, выбрав у команды `mkdir` опцию `-p`, позволяющую создавать последовательность вложенных каталогов (рис. 5.18):

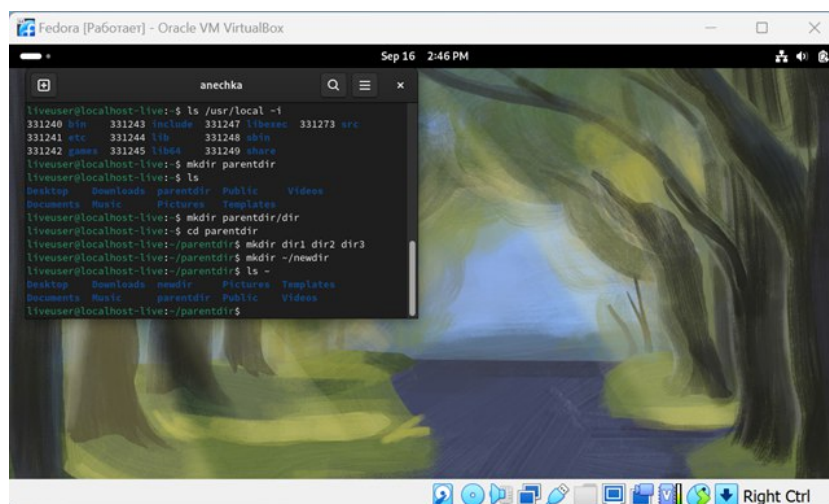


Рис. 5.18: создание иерархической цепочки подкаталогов

Создаю файл `text.txt` в каталоге `~/newdir/dir1/dir2`, с помощью команды `touch`, прописывая путь к месту создания файла, в конце которого добавляю имя создаваемого файла `~/newdir/dir1/dir2/text.txt`, также проверяю наличие файла с помощью команды `ls ~/newdir/dir1/dir2`, снова указывая путь от домашней директории (рис. 5.19):



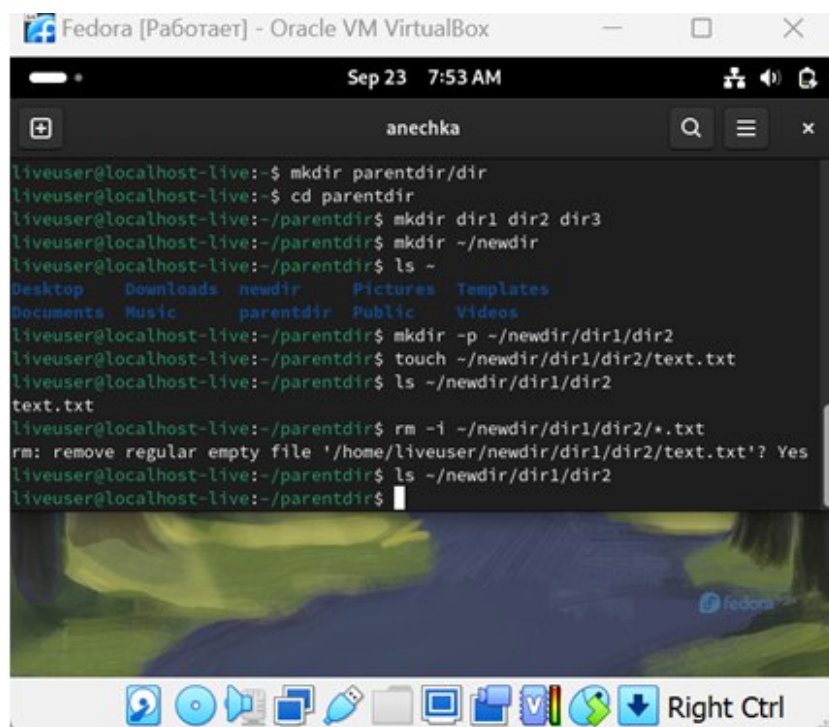


Рис. 5.19: Создание файла

### 3. Перемещение и удаление файлов и каталогов

Для удаления пустых каталогов можно воспользоваться командой `rmdir`. Запросив подтверждение на удаление каждого файла в текущем каталоге, удалю в подкаталоге `/newdir/dir1/dir2/` все файлы с именами, заканчивающимися на `.txt` (рис. 5.20):

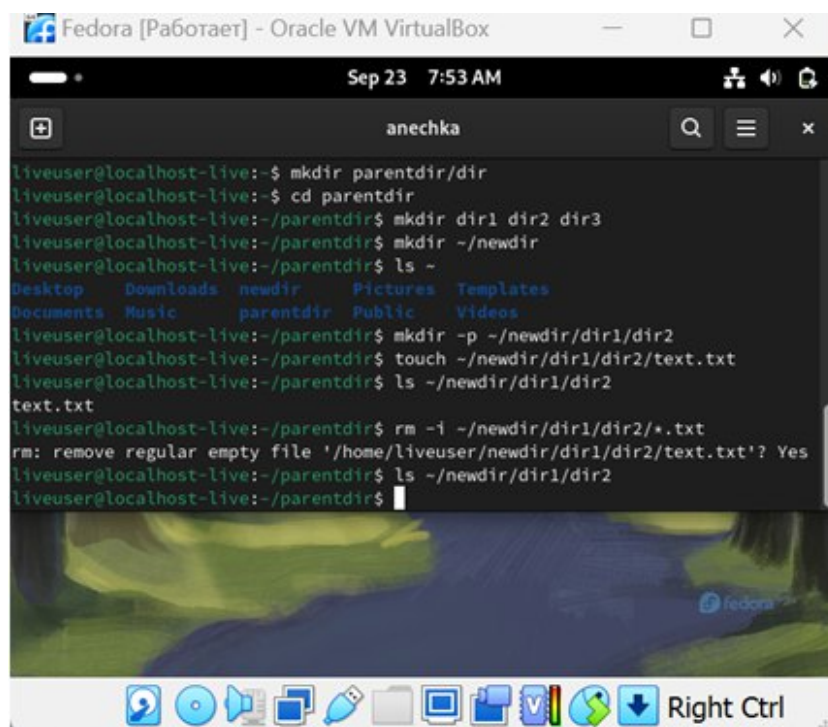


Рис. 5.20: Удаление файла с запросом подтверждения

Рекурсивно удаляю из текущего каталога без запроса подтверждения на удаление каталог newdir с помощью ключа -R, а также файлы, чьи имена начинаются с dir в каталоге parentdir (рис. 5.21):

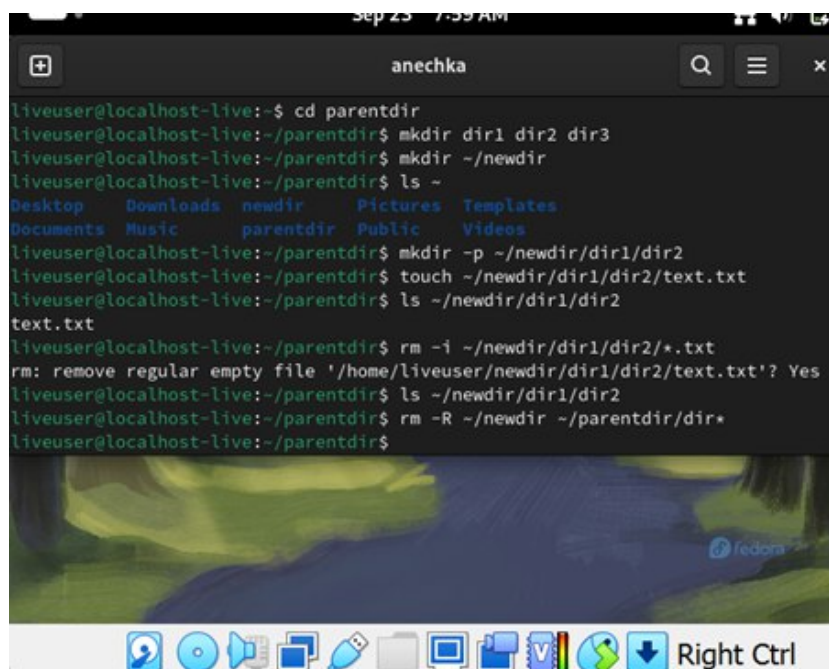


Рис. 5.21: Рекурсивное удаление директорий

С помощью `ls` и `ls ~` проверяю правильность выполнения команды (рис. 5.22):

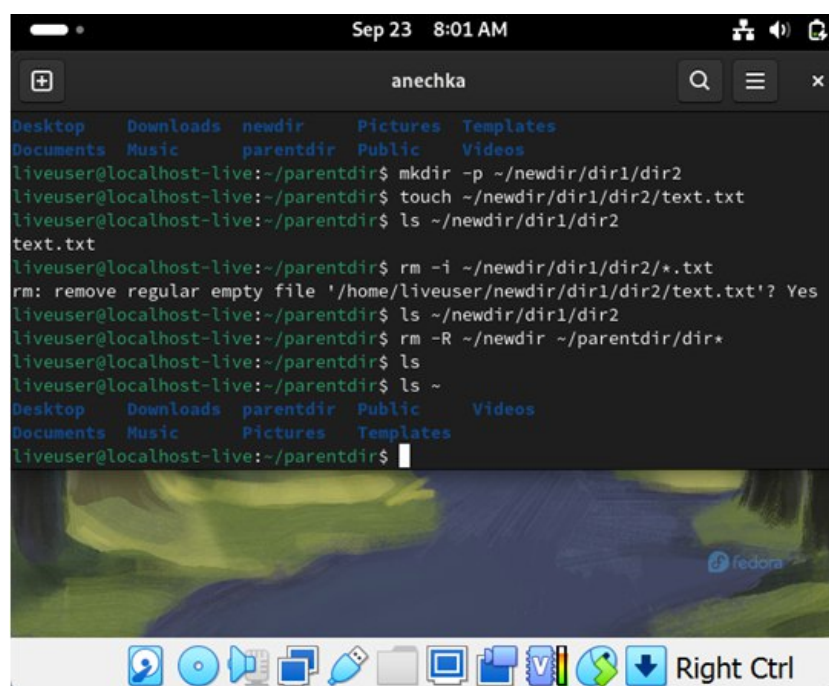
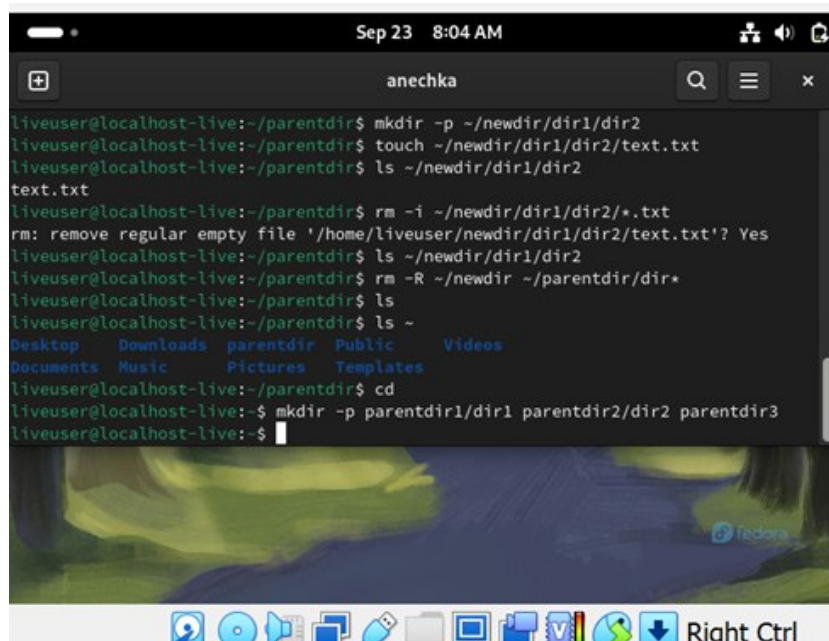


Рис. 5.22: Проверка правильности выполнения команд

Перемещаюсь в домашний каталог, создаю последовательности вложенных



каталогов parentdir1/dir1 parentdir2/dir2 с помощью ключа -p команды mkdir и каталог parentdir3, передаю команде три аргумента (рис. 5.23):



```
liveuser@localhost-live:~/parentdir$ mkdir -p ~/newdir/dir1/dir2
liveuser@localhost-live:~/parentdir$ touch ~/newdir/dir1/dir2/text.txt
liveuser@localhost-live:~/parentdir$ ls ~/newdir/dir1/dir2
text.txt
liveuser@localhost-live:~/parentdir$ rm -i ~/newdir/dir1/dir2/*.txt
rm: remove regular empty file '/home/liveuser/newdir/dir1/dir2/text.txt'? Yes
liveuser@localhost-live:~/parentdir$ ls ~/newdir/dir1/dir2
liveuser@localhost-live:~/parentdir$ rm -R ~/newdir ~/parentdir/dir*
liveuser@localhost-live:~/parentdir$ ls
liveuser@localhost-live:~/parentdir$ ls ~
Desktop  Downloads  parentdir  Public  Videos
Documents  Music  Pictures  Templates
liveuser@localhost-live:~/parentdir$ cd
liveuser@localhost-live:~$ mkdir -p parentdir1/dir1 parentdir2/dir2 parentdir3
liveuser@localhost-live:~$
```

Рис. 5.23: Создание новых директорий

Создаю файл text1.txt в директории parentdir1/dir1/ с помощью команды touch. Проверяю на наличие созданного файла в директории (рис. 5.24):

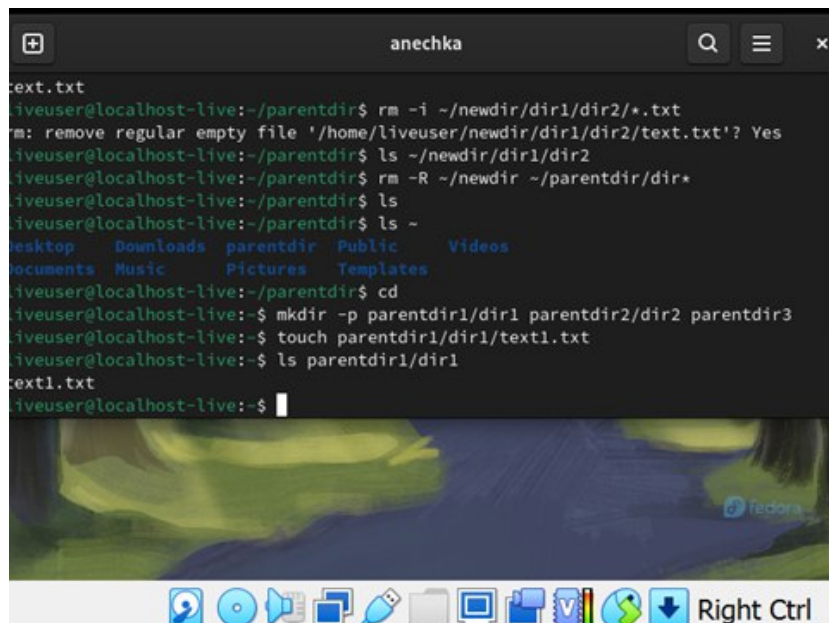


Рис. 5.24: Создание файла

Аналогично действую для создания файла text2.txt (рис. 5.25):

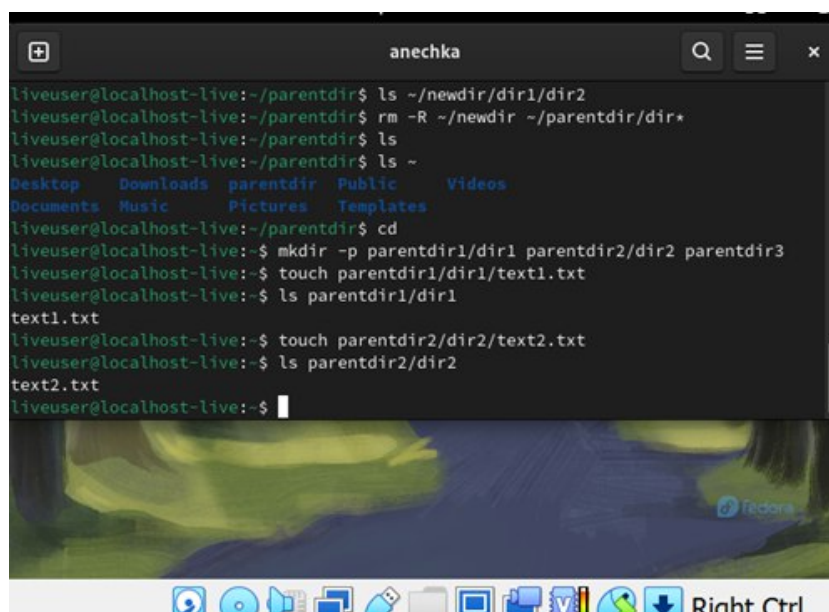


Рис. 5.25: Создание файла

Использую команду mv, перемещаю файл text1.txt, указывая путь к нему, в директорию parentdir3 (рис. 5.26):

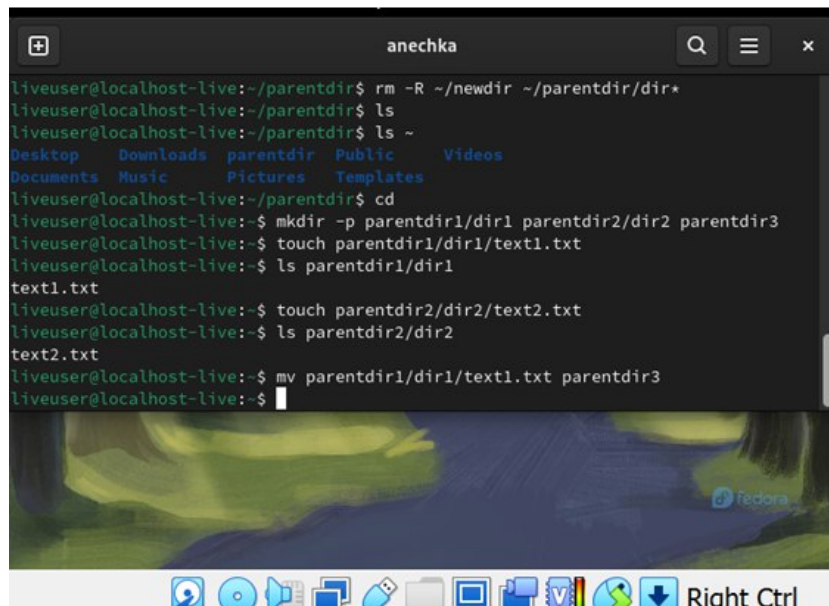


Рис. 5.26: Перемещение файла

Используя команду `cp`, копирую файл `text2.txt` в каталог `parentdir3`, также указывая путь к файлу, который нужно скопировать (рис. 5.27):

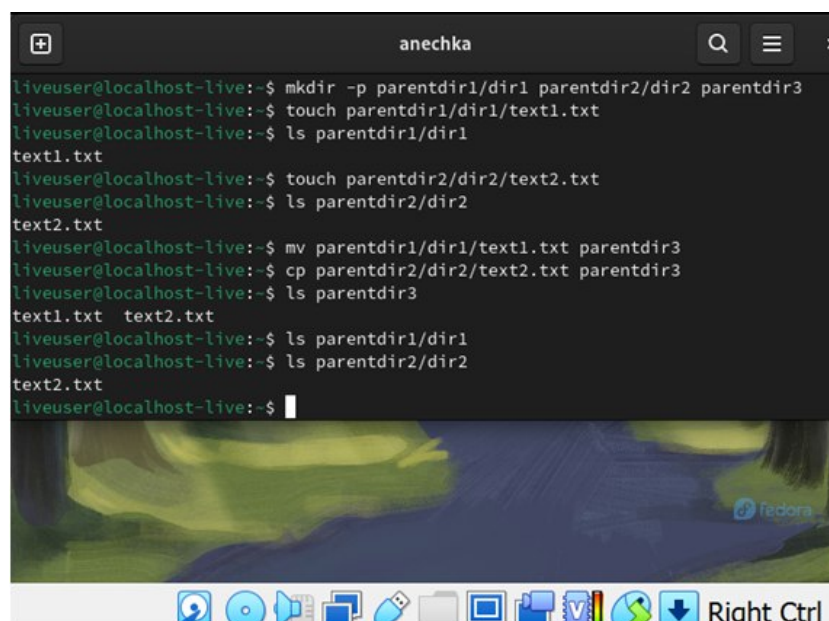
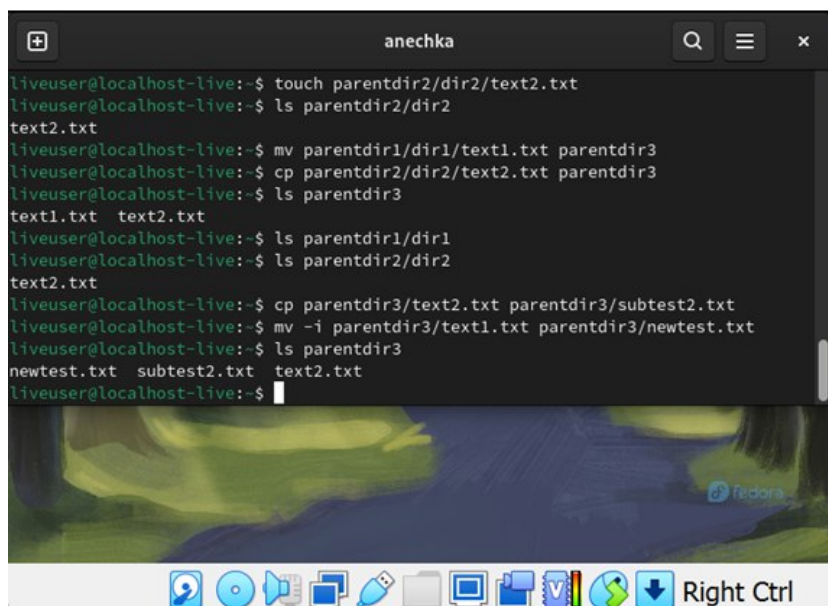


Рис. 5.27: Копирование файла

Проверяю, что в каталоге `parentdir3` действительно два файла, файла `text1.txt` теперь нет в каталоге `parentdir1/dir1`, `text2.txt` все еще находится в `parentdir2/dir2`

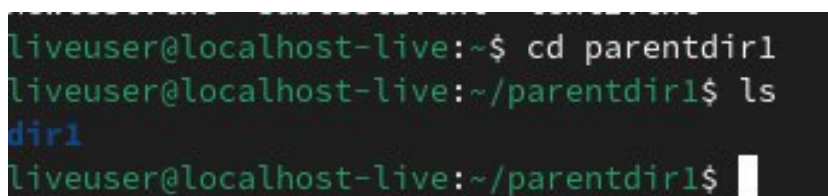
(рис. 5.28):



```
liveuser@localhost-live:~$ touch parentdir2/dir2/text2.txt
liveuser@localhost-live:~$ ls parentdir2/dir2
text2.txt
liveuser@localhost-live:~$ mv parentdir1/dir1/text1.txt parentdir3
liveuser@localhost-live:~$ cp parentdir2/dir2/text2.txt parentdir3
liveuser@localhost-live:~$ ls parentdir3
text1.txt  text2.txt
liveuser@localhost-live:~$ ls parentdir1/dir1
text2.txt
liveuser@localhost-live:~$ ls parentdir2/dir2
text2.txt
liveuser@localhost-live:~$ cp parentdir3/text2.txt parentdir3/subtest2.txt
liveuser@localhost-live:~$ mv -i parentdir3/text1.txt parentdir3/newtest.txt
liveuser@localhost-live:~$ ls parentdir3
newtest.txt  subtest2.txt  text2.txt
liveuser@localhost-live:~$
```

Рис. 5.28: Проверка работы команды

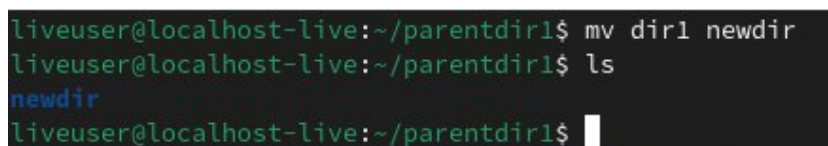
Просмотрим файлы в директории parentdir3 с помощью ls. Создаю копию text2.txt с новым именем subtest2.txt с помощью команды cp. Переименовую файл text1.txt из каталога parentdir3 в newtest.txt с помощью команды mv, а с помощью ее ключа -i запрашиваю подтверждение перед перезаписью. Проверяю правильность выполнения работы с помощью ls (рис. 5.29):



```
liveuser@localhost-live:~$ cd parentdir1
liveuser@localhost-live:~/parentdir1$ ls
dir1
liveuser@localhost-live:~/parentdir1$
```

Рис. 5.29: Копирование и перемещение файлов

Перехожу в директорию parentdir1 с помощью утилиты cd (рис. 5.30):



```
liveuser@localhost-live:~/parentdir1$ mv dir1 newdir
liveuser@localhost-live:~/parentdir1$ ls
newdir
liveuser@localhost-live:~/parentdir1$
```

Рис. 5.30: Перемещение по директориям

Переименовываю каталог dir1 в каталоге parentdir1 в newdir с помощью mv. Я нахожусь в директории, где находится подкаталог dir1, поэтому прописывать путь до подкаталога мне не нужно (рис. 5.31):

```
liveuser@localhost-live:~/parentdir1$ mv dir1 newdir
liveuser@localhost-live:~/parentdir1$ ls
newdir
liveuser@localhost-live:~/parentdir1$
```

Рис. 5.31: Переименование каталога

#### 4. Команда cat: вывод содержимого файлов

Возвращаюсь в домашнюю директорию с помощью команды cd. Команда cat объединяет файлы и выводит их на стандартный вывод: использую команду cat, чтобы прочитать файл hosts в подкаталоге etc корневого каталога, для этого в аргументе к команде указываю абсолютный путь к файлу (рис. 5.32):

```
liveuser@localhost-live:~/parentdir1$ cd
liveuser@localhost-live:~$ cat /etc/hosts
# Loopback entries; do not change.
# For historical reasons, localhost precedes localhost.localdomain:
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1        localhost localhost.localdomain localhost6 localhost6.localdomain6
# See hosts(5) for proper format and other examples:
# 192.168.1.10 foo.example.org foo
# 192.168.1.13 bar.example.org bar
liveuser@localhost-live:~$
```

Рис. 5.32: Чтение файла

#### 5. Задание для самостоятельной работы

Воспользовавшись командой pwd, узнаю полный путь к своей домашней директории (рис. 5.33):

```
liveuser@localhost-live:~$ pwd
/home/liveuser
liveuser@localhost-live:~$
```

Рис. 5.33: Путь к домашнему каталогу

Ввожу последовательность команд (рис. 5.34):



```
liveuser@localhost-live:~$ cd
liveuser@localhost-live:~$ mkdir tmp
liveuser@localhost-live:~$ cd tmp
liveuser@localhost-live:~/tmp$ pwd
/home/liveuser/tmp
liveuser@localhost-live:~/tmp$ cd /tmp
liveuser@localhost-live:/tmp$ pwd
/tmp
liveuser@localhost-live:/tmp$
```

Рис. 5.34: Ввод команд

Сначала я возвращаюсь в домашнюю директорию, затем создаю в ней директорию tmp, перехожу в подкаталог домашнего каталога tmp с помощью команды cd. Использую команду pwd. Получаю путь к директории tmp, начинающийся от корневого каталога, домашнего каталога пользователя, потому что именно в домашнем каталоге я сама создала директорию. Если я использую команду «cd /tmp», где / - корневой каталог, tmp – подкаталог корневого каталога, в котором содержатся временные файлы, эта директория есть в системе по умолчанию и путь к ней отличен от созданной мной директории tmp, поэтому при последующем использовании команды pwd, я получаю вывод /tmp.

Перехожу в корневой каталог с помощью cd /, просматриваю его содержимое с помощью ls, добавляю к утилите ключ -a, чтобы увидеть скрытые файлы «.» и «..» в директории (рис. 5.35):

```
liveuser@localhost-live:/tmp$ cd /
liveuser@localhost-live:/$ ls
afs boot etc lib lost+found mnt proc run srv tmp var
bin dev home lib64 media opt root sbin sys usr
liveuser@localhost-live:/$
liveuser@localhost-live:/$ ls -a
. bin etc lib64 lost+found opt run sys var
.. boot home .liveimg-configured media proc sbin tmp
afs dev lib .liveimg-late-configured mnt root srv usr
liveuser@localhost-live:/$
```

Рис. 5.35: Содержимое корневого каталога

Возвращаюсь в домашнюю директорию с помощью cd, указываю к директории абсолютный путь. Просматриваю с помощью ls содержимое домашнего каталога.

Чтобы просмотреть содержимое со скрытыми файлами снова использую ls -a (рис. 5.36):

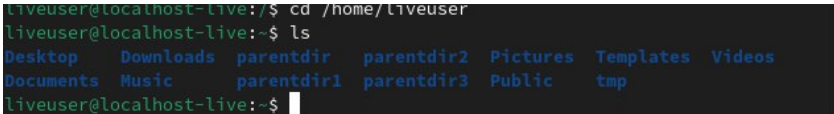


Рис. 5.36: Содержание домашнего каталога

Из домашней директории просматриваю содержимое каталога etc с помощью команды ls, указав абсолютный путь к каталогу (рис. 5.37):

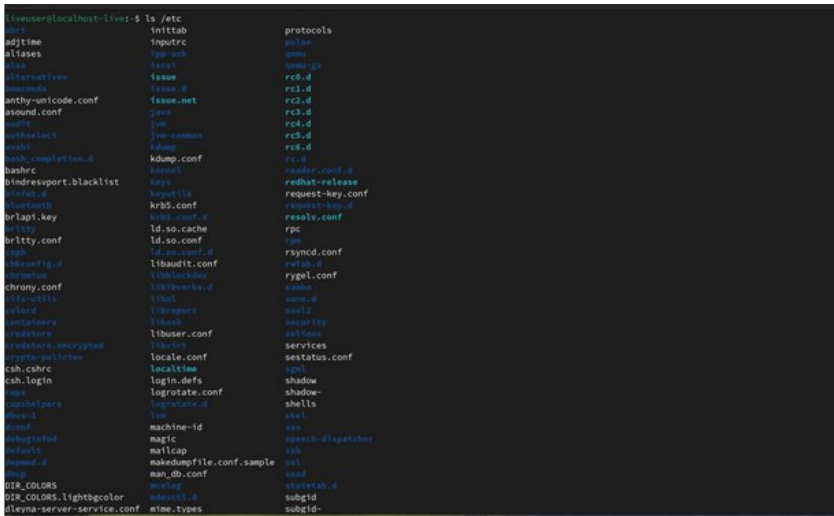


Рис. 5.37: Содержимое каталога /etc

С помощью команды `cd` перехожу в каталог `/usr/local`. С помощью команды `ls` смотрю содержание этого каталога. К команде `ls` добавляю ключ `-a` и просматриваю всё содержимое каталога, включая скрытые файлы (рис. 5.38):

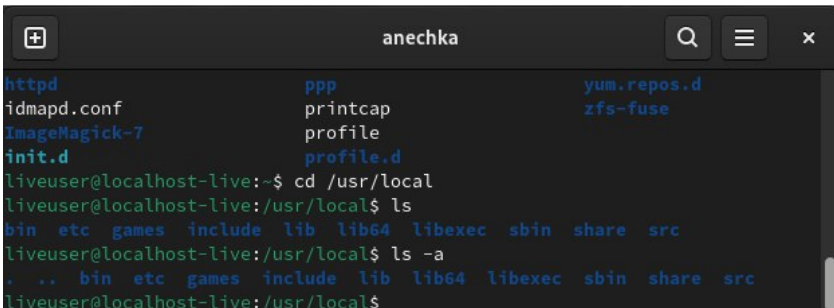


Рис. 5.38: Содержимое каталога /usr/local

Пользуясь изученными консольными командами, в своём домашнем каталоге создаю каталог temp и каталог labs (labs нужно создавать с подкаталогами, для этого использую ключ -p) с подкаталогами lab1, lab2 и lab3 одной командой (рис. 5.39):

```
liveuser@localhost-live:/usr/local$ cd
liveuser@localhost-live:~$ mkdir -p temp labs/labs1 labs/labs2 labs/labs3
liveuser@localhost-live:~$
```

Рис. 5.39: Создание директорий

В каталоге temp создаю файлы text1.txt, text2.txt, text3.txt. Пользуясь командой ls, убеждаюсь, что все действия выполнены успешно (каталоги и файлы созданы) (рис. 5.40):

```
liveuser@localhost-live:~$ cd
liveuser@localhost-live:~$ touch temp/text1.txt temp/text2.txt temp/text3.txt
liveuser@localhost-live:~$ ls temp
text1.txt text2.txt text3.txt
liveuser@localhost-live:~$
```

Рис. 5.40: Создание файлов в каталоге temp

С помощью ls labs проверяю правильность создания подкаталогов в каталоге labs (рис. [-fig. 5.41):

```
liveuser@localhost-live:~$ ls labs
labs1 labs2 labs3
liveuser@localhost-live:~$
```

Рис. 5.41: Файлы в labs

Выбираю во вкладке «файл» пункт «открыть» text.1. Открывается графический файловый менеджер, в нем выбираю путь к нужному файлу: домашний каталог, подкаталог temp. Выбираю нужный файл text1.txt (рис. 5.42):



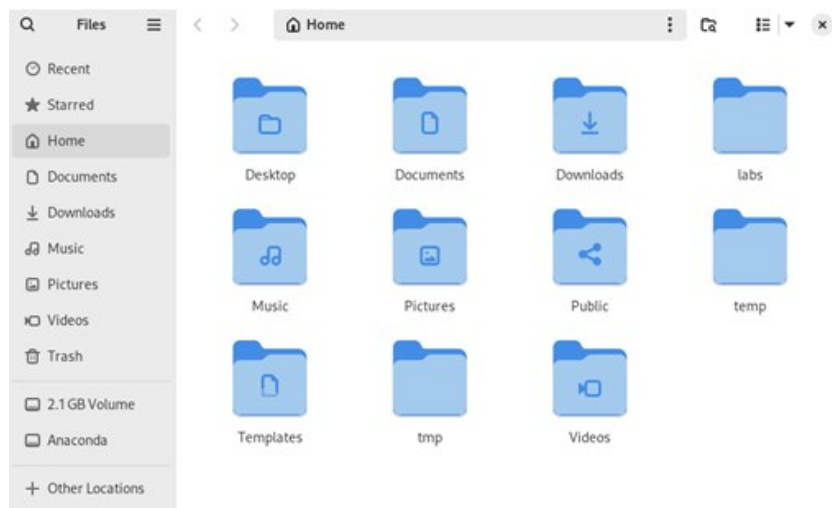


Рис. 5.42: Рисунок 5.42

(рис. 5.43):

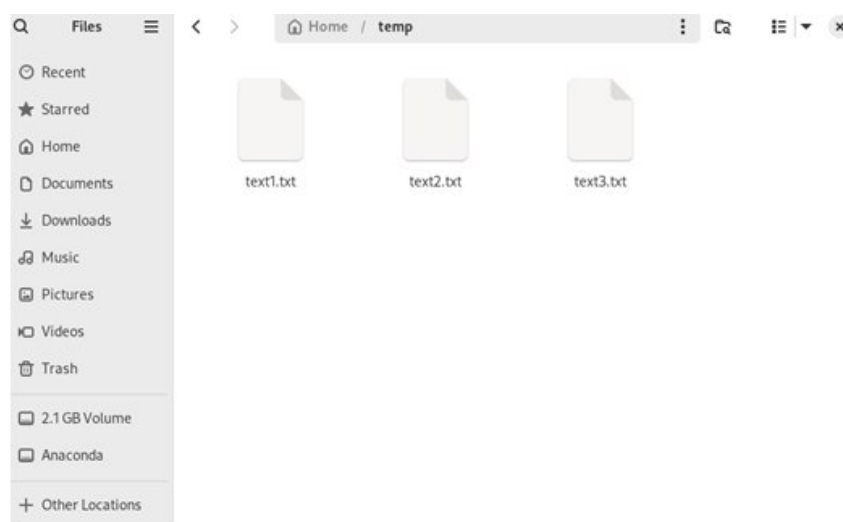


Рис. 5.43: Рисунок

(рис. 5.44):

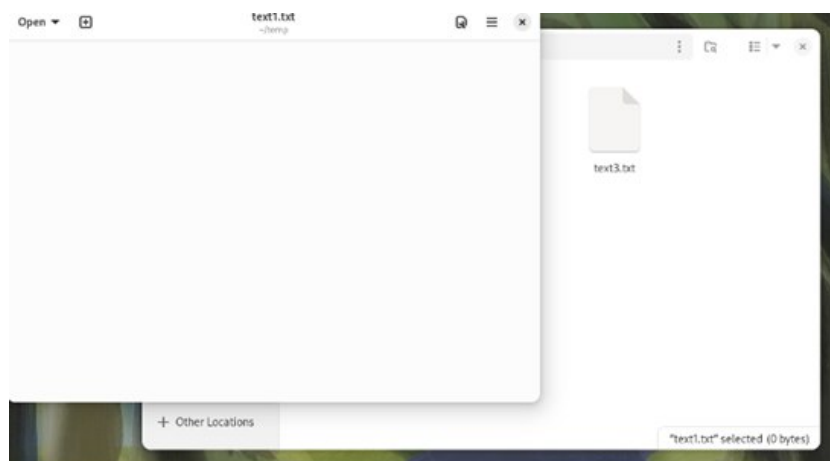


Рис. 5.44: Рисунок 5.44

Записываю в файл свое имя (рис. 5.45):



Рис. 5.45: Рисунок 5.45

Открываю файл text2.txt аналогично тому, как открывала text1.txt. Записываю в файл text2.txt свою фамилию (рис. 5.46):

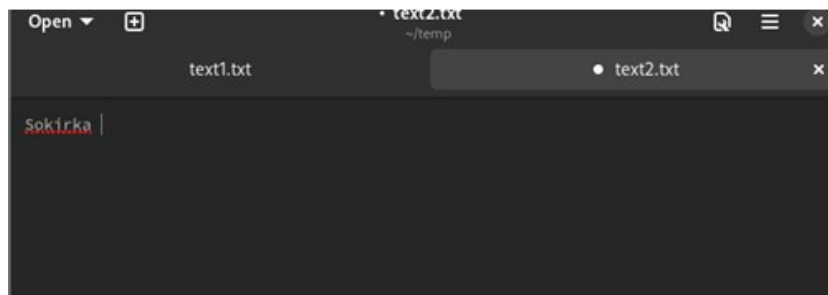


Рис. 5.46: Создание файлов в каталоге temp

С помощью команды `cat` просмотрю содержимое каталога `temp` (рис. 5.47):

```
liveuser@localhost-live:~$ cat temp/text1.txt temp/text2.txt temp/text3.txt
Anna
Sokirka
NKAbd-04-24
```

Рис. 5.47: Рисунок 5.47

Копирую файлы, чьи имена заканчиваются на `.txt`, из каталога `~/temp` в каталог `labs`. Выбираю все файлы с помощью `*`, копирую их с помощью утилиты `cp` (рис. 5.48):

```
liveuser@localhost-live:~$ cd
liveuser@localhost-live:~$ cp ~/temp/*.txt labs
```

Рис. 5.48: Рисунок 5.48

Переименовываю файлы каталога `labs`: `text1.txt` переименовываю в `firstname.txt` и перемещаю в подкаталог `lab1`, `text2.txt` переименовываю в `lastname.txt` и перемещаю в подкаталог `lab2`, `text3.txt` переименовываю в `id-group.txt` и перемещаю в подкаталог `lab3` (рис. 5.49):

```
liveuser@localhost-live:~$ cd
liveuser@localhost-live:~$ cp ~/temp/*.txt labs
```

Рис. 5.49: Рисунок 5.49

Воспользовавшись командой `ls`, я проверяю содержание каталога `lab`. С помощью `ls` проверяю содержание каждого подкаталога каталога `labs` и тут же читаю

с помощью утилиты cat содержимое файла в каталоге, которое выводилось при применении прошлой команды: проверяю, какие файлы есть в директории lab1, читаю содержимое этого файла в командной строке. Аналогично для lab2 и lab3 (рис. 5.50):

```
liveuser@localhost-live:~$ cp ~/temp/text1.txt lab1
liveuser@localhost-live:~$ mv ~/labs/text1.txt ~/labs/lab1/firstname.txt
liveuser@localhost-live:~$ mv ~/labs/text2.txt ~/labs/lab2/lastname.txt
liveuser@localhost-live:~$ mv ~/labs/text3.txt ~/labs/lab3/id-group.txt
liveuser@localhost-live:~$
```

Рис. 5.50: Рисунок 5.50

Использую ls, чтобы проверить содержимое домашнего каталога. С помощью команды rm и ее ключа -R удаляю каталоги labs, temp, tmp. Вместе с их содержанием. Проверяю с помощью ls (рис. 5.51):

```
liveuser@localhost-live:~$ rm -R labs temp tmp parentdir parentdir1 parentdir2 parentdir3
rm: cannot remove 'parentdir': No such file or directory
rm: cannot remove 'parentdir1': No such file or directory
rm: cannot remove 'parentdir2': No such file or directory
rm: cannot remove 'parentdir3': No such file or directory
liveuser@localhost-live:~$ rm -R labs temp tmp
rm: cannot remove 'labs': No such file or directory
rm: cannot remove 'temp': No such file or directory
rm: cannot remove 'tmp': No such file or directory
liveuser@localhost-live:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
```

Рис. 5.51: Рисунок 5.51

## **6 Выводы**

При выполнении данной лабораторной работы я приобрела практические навыки работы с операционной системой на уровне командной строки, изучила организацию файловой системы, научилась создавать и удалять файлы и директории.

## 7 Список литературы

1. [https://esystem.rudn.ru/pluginfile.php/2089080/mod\\_resource/content/0/Лабораторная%20работа%20№1.%20Основы%20интерфейса%20командной%20строки%20ОС%20GNU%20Linux.pdf](https://esystem.rudn.ru/pluginfile.php/2089080/mod_resource/content/0/Лабораторная%20работа%20№1.%20Основы%20интерфейса%20командной%20строки%20ОС%20GNU%20Linux.pdf)