

# **РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

*Факультет физико-математических и естественных наук*

*Кафедра прикладной информатики и теории вероятностей*

## **ОТЧЕТ**

### **ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1**

**дисциплина:** Архитектура компьютера

**Студент:** Виме Давид Тененте

**Группа:** НКАбд-021-24

**МОСКВА**

2024 г.

## Оглавление

1 Цель работы.....	3
2 Задание.....	4
3 Теоретическое введение .....	5
4 Выполнение лабораторной работы.....	7
4.1 Техническое обеспечение .....	7
4.2 Перемещение по файловой системе .....	7
4.3 Создание пустых каталогов и файлов .....	13
4.4 Перемещение и удаление файлов или каталогов..	20
4.3 Команда cat: вывод содержимого файлов .....	26
5 Выводы.....	28
Список литературы .....	<b>Erro! Indicador não definido.</b>

# **1 Цель работы**

Приобретение практических навыков работы с операционной системой на уровне командной строки (организация файловой системы, навигация по файловой системе, создание и удаление файлов и директорий).

## **2 Задание**

На основе методических указаний провести работу с базовыми командами терминала, выучить применение команд для разных случаев использования, а также ключей для них.

### 3 Теоретическое введение

Операционная система (ОС)— это комплекс взаимосвязанных программ, предназначенных для управления ресурсами компьютера и организации взаимодействия с пользователем. Сегодня наиболее известными операционными системами являются ОС семейства Microsoft Windows и UNIX-подобные системы. 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). Linux-системы распространяются в основном бесплатно в виде различных дистрибутивов. 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. Работу ОС GNU Linux можно представить в виде функционирования множества взаимосвязанных процессов. При загрузке системы сначала запускается ядро, которое, в свою очередь, запускает оболочку ОС (от англ. shell «оболочка»). Взаимодействие пользователя с системой Linux (работа с данными и управление работающими в системе процессами) происходит в интерактивном режиме посредством командного языка. Оболочка операционной системы (или командная оболочка, интерпретатор команд) — интерпретирует (т.е. переводит на машинный язык) вводимые пользователем команды, запускает соответствующие программы (процессы), формирует и выводит ответные сообщения. Кроме того, на языке командной оболочки можно писать небольшие программы для выполнения ряда последовательных операций с файлами и содержащимися в них данными — сценарии (скрипты).

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

Таблица 3.1 Описание некоторых каталогов файловой системы GNU Linux

Команда	Описание	
pwd	<b>P</b> rint <b>W</b> orking <b>D</b> irectory	определение текущего каталога
cd	<b>C</b> hange <b>D</b> irectory	смена каталога
ls	<b>L</b> i <b>S</b> t	вывод списка файлов
mkdir	<b>M</b> a <b>K</b> e <b>D</b> I <b>R</b> ectory	создание пустых каталогов
touch		создание пустых файлов
rm	<b>R</b> e <b>M</b> ove	удаление файлов или каталогов
mv	<b>M</b> o <b>V</b> e	перемещение файлов и каталогов
cp	<b>C</b> o <b>P</b> y	копирование файлов и каталогов
cat		вывод содержимого файлов

Таблица 3.2 Основные команды взаимодействия пользователя с файловой системой

## 4 Выполнение лабораторной работы

### 4.1 Техническое обеспечение

Лабораторная работа была выполнена на домашнем компьютере под управлением операционной системы Fedora Workstation 40.

### 4.2 Перемещение по файловой системе

Я открыл терминал, по умолчанию в нем стоит домашняя директория, убедиться в этом можно, убедившись в наличии тильды ~ в приветствии командной строки. В домашнюю директорию можно перейти также, отправив команду `cd` в терминал. (рис. 4.2.1)

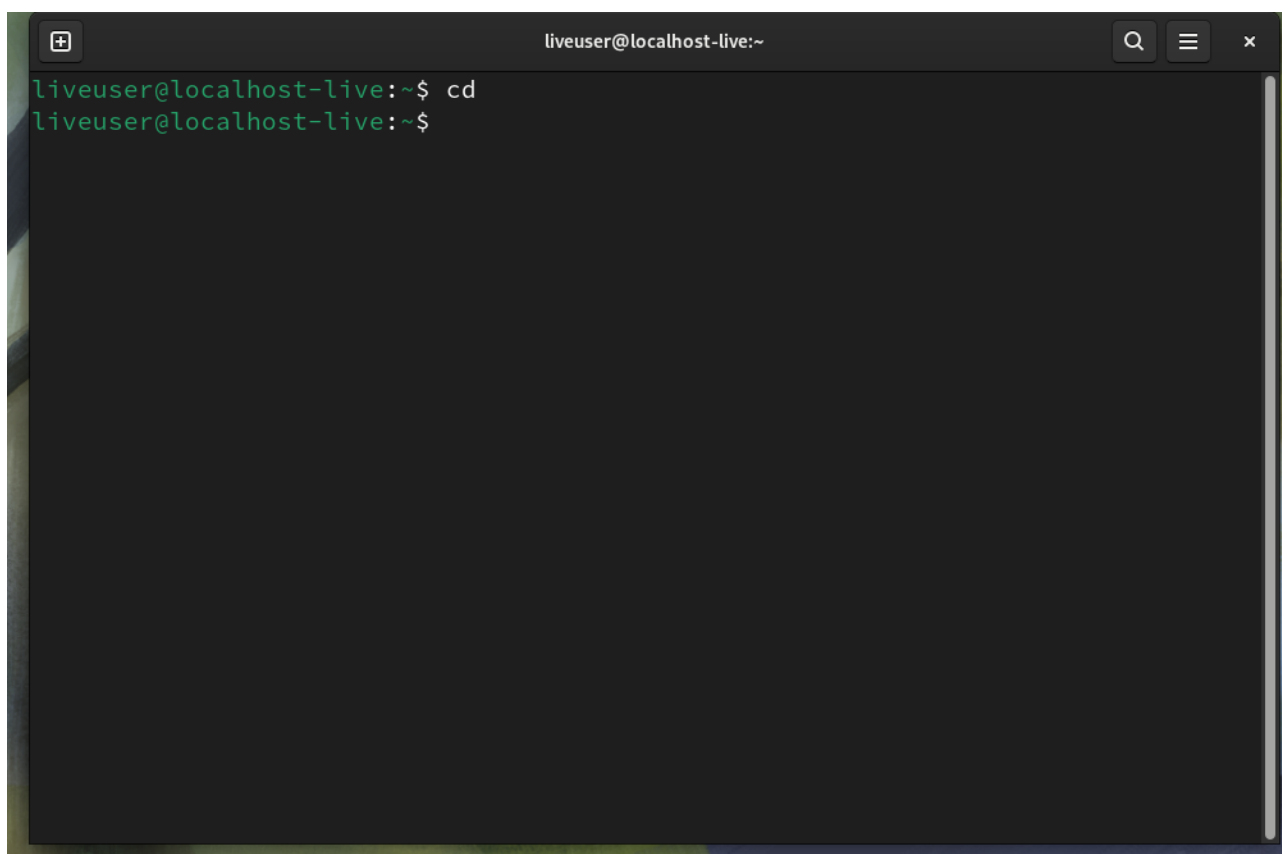
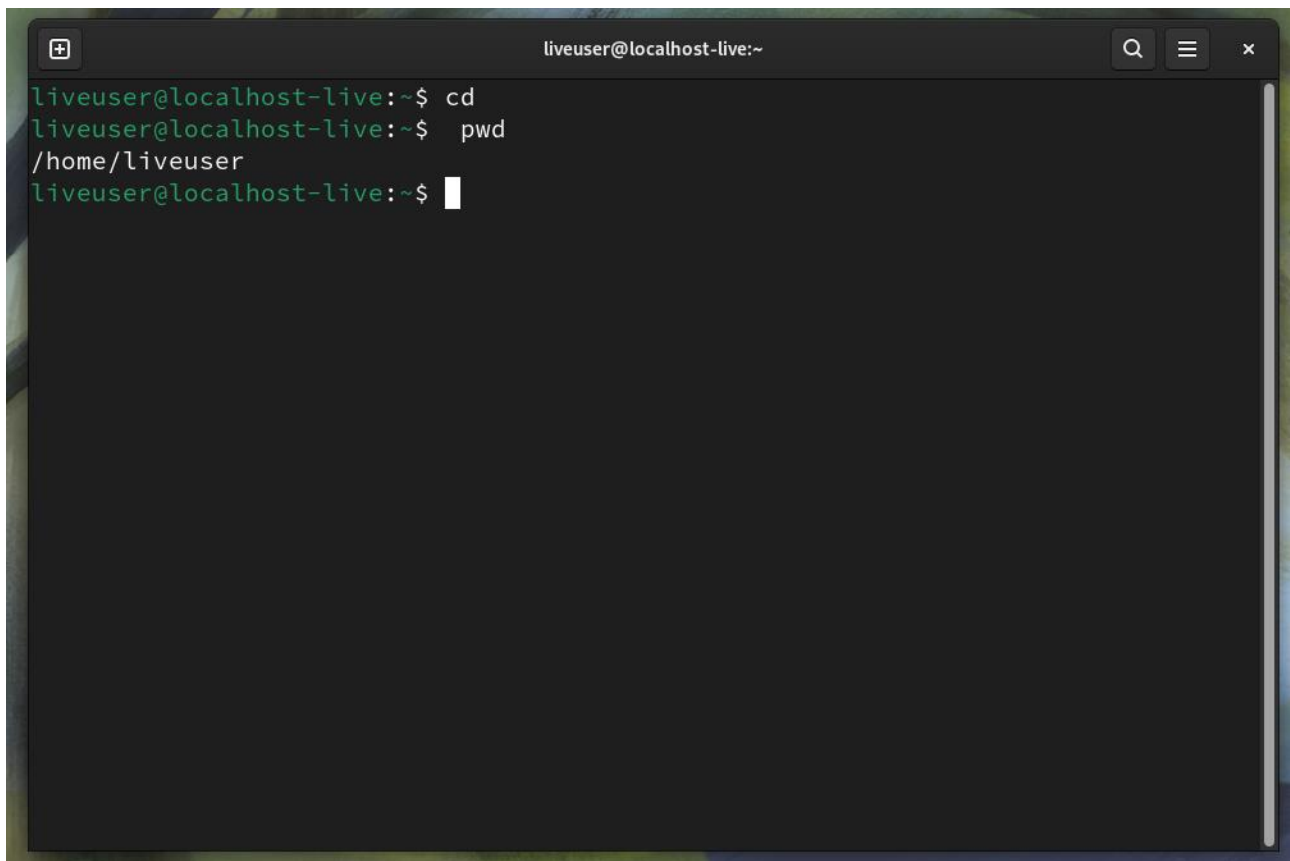


Рис. 4.2.1 Окно терминала с домашней директорией.

С помощью команды `pwd` я могу посмотреть полный путь до текущей – домашней – директории. (рис. 4.2.2)

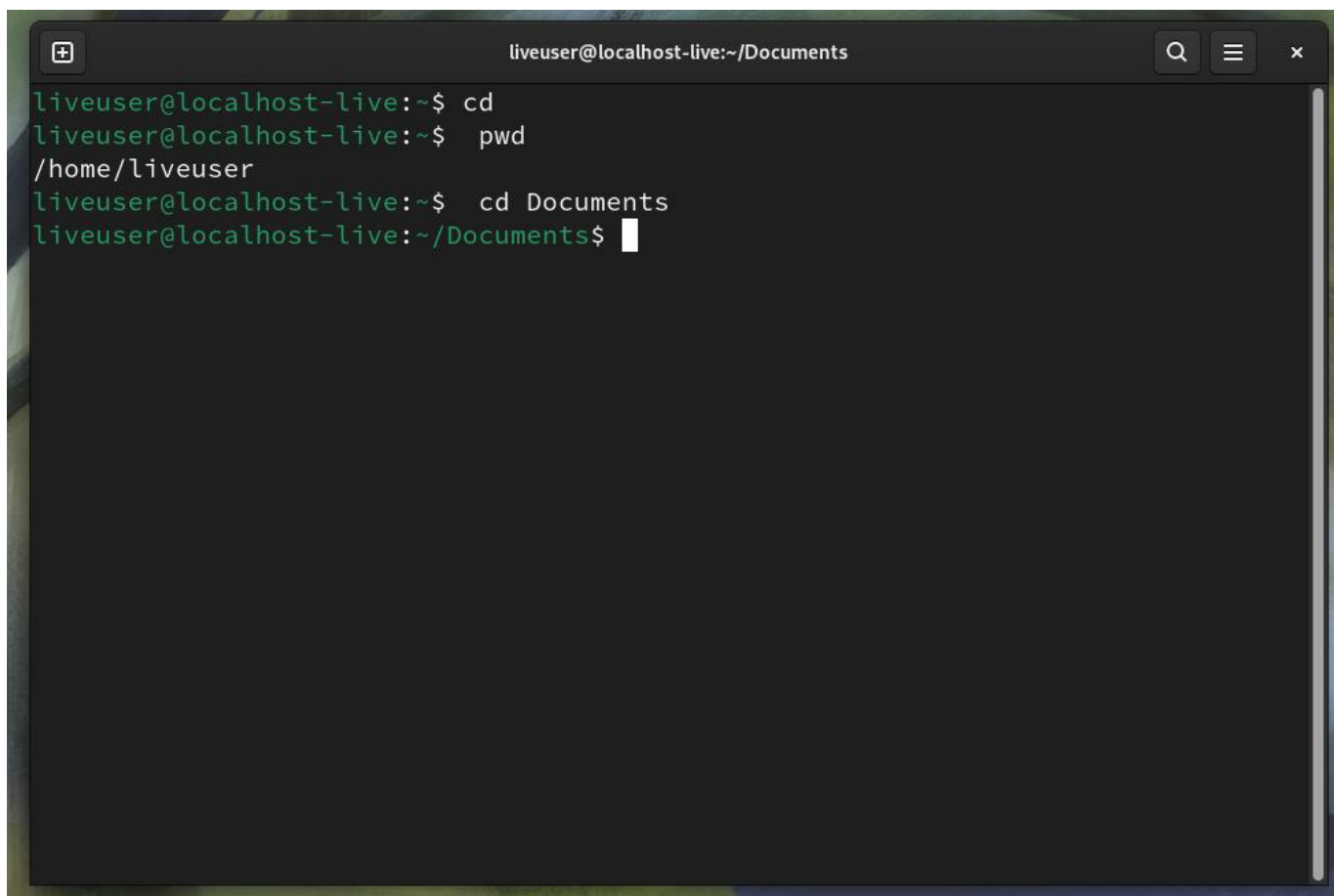
A terminal window with a dark background. The title bar at the top reads 'liveuser@localhost-live:~'. The terminal shows the following sequence of commands and output: 'liveuser@localhost-live:~\$ cd' followed by 'liveuser@localhost-live:~\$ pwd' which outputs '/home/liveuser'. The prompt 'liveuser@localhost-live:~\$' is followed by a white cursor. The window has standard Linux window controls (minimize, maximize, close) and a search icon in the top right corner.

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

Рис. 4.2.2 В терминале выведен полный путь домашней директории.

Далее я перехожу в подкаталог Documents домашней директории, отправив команду терминалу `cd Documents` (рис. 4.2.3)

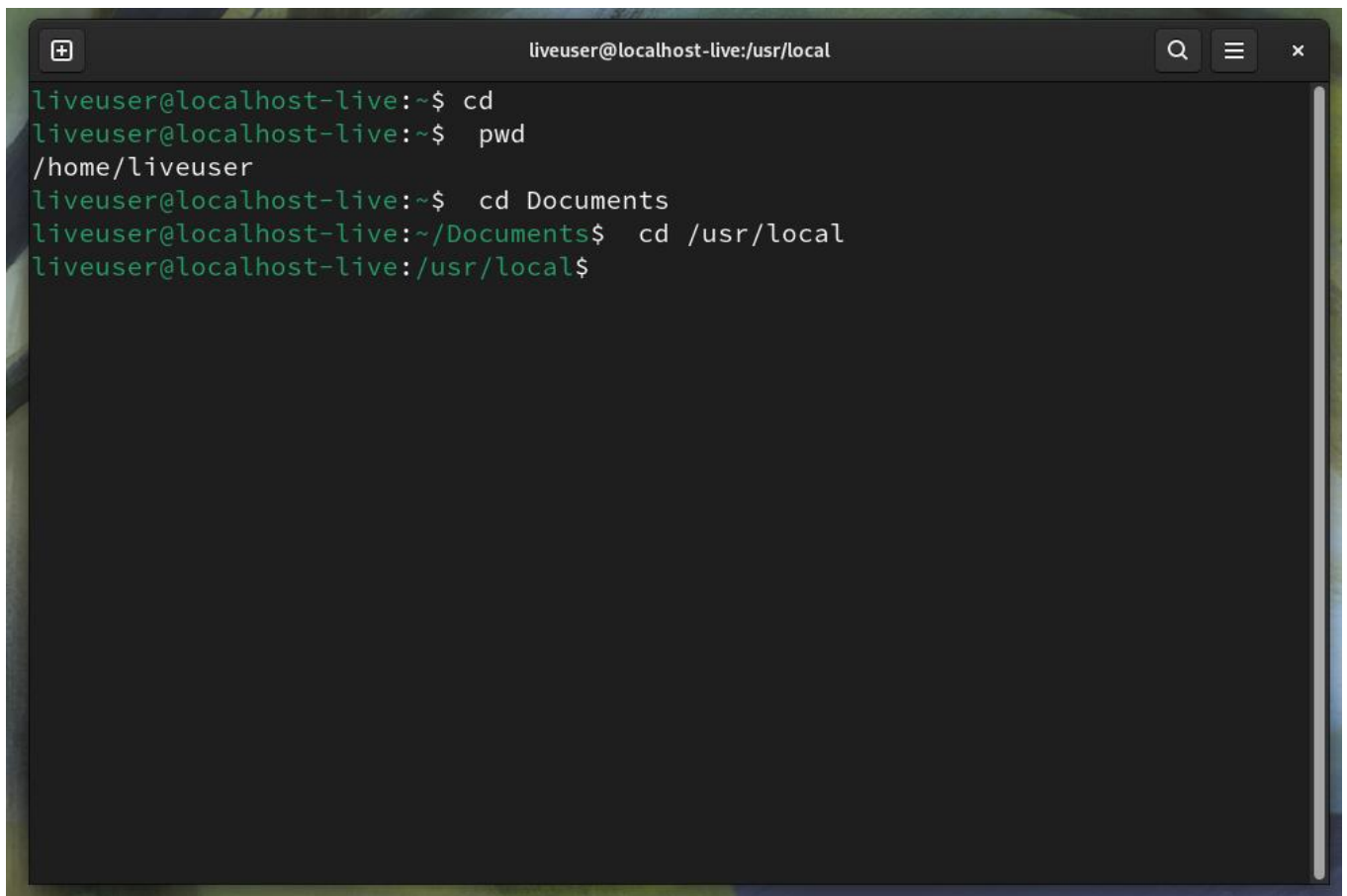


A terminal window with a dark background and light green text. The title bar at the top reads 'liveuser@localhost-live:~/Documents'. The terminal shows a sequence of commands and their outputs: 'cd' is entered, followed by 'pwd' which outputs '/home/liveuser'. Then 'cd Documents' is entered, and the prompt changes to 'liveuser@localhost-live:~/Documents\$'.

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

Рис. 4.2.3 В терминале открыт подкаталог Documents домашней директории.

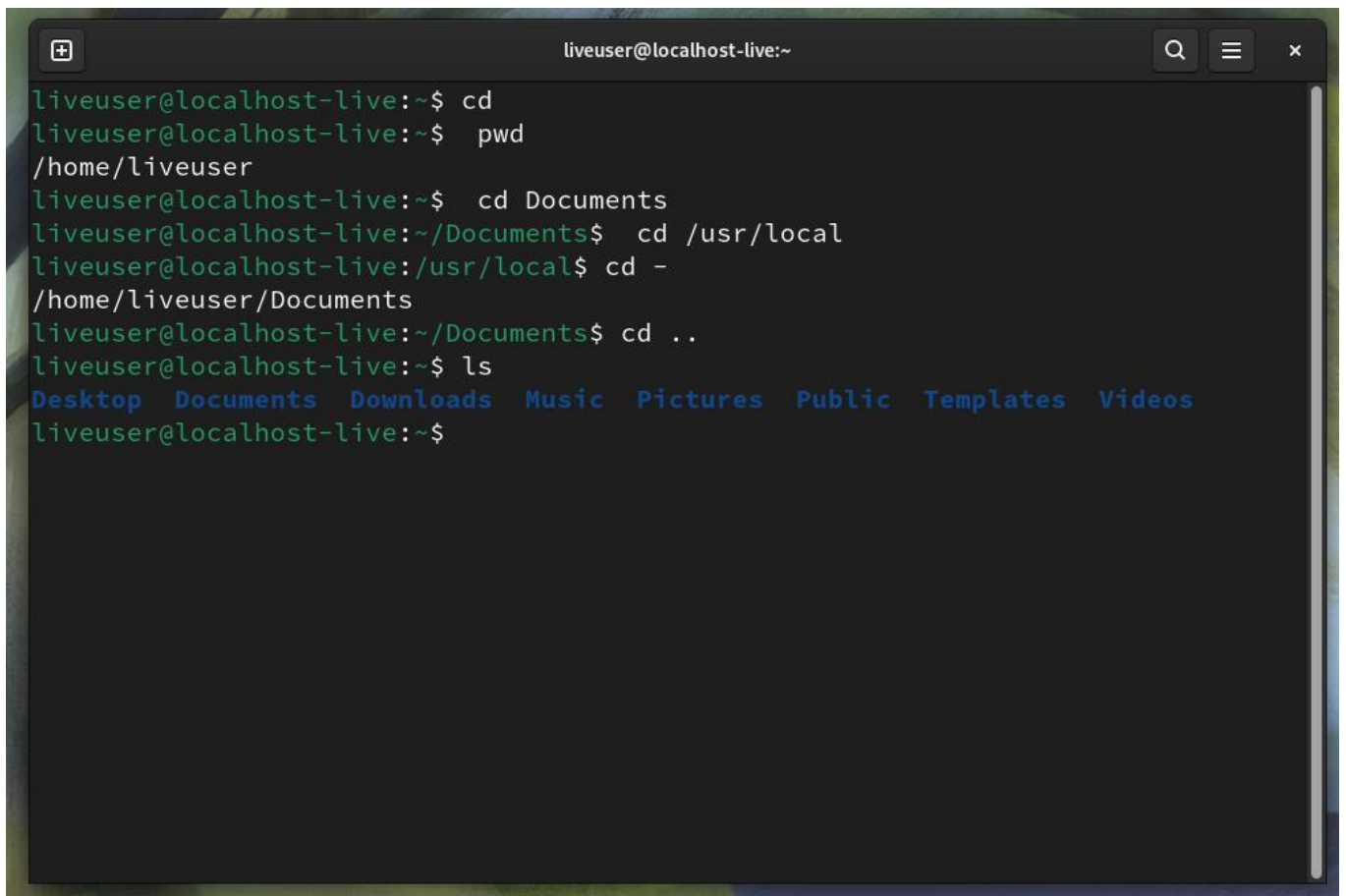
После я перехожу в каталог local – подкаталог usr корневого каталога, указав абсолютный путь к нему (/usr/local). (рис. 4.2.4)

A terminal window with a dark background and light green text. The window title bar shows 'liveuser@localhost-live:/usr/local' and standard window controls. The terminal shows a sequence of commands and their outputs: 'cd' followed by a new prompt, 'pwd' followed by '/home/liveuser', 'cd Documents' followed by a new prompt, and 'cd /usr/local' followed by a new prompt.

```
liveuser@localhost-live:~$ cd
liveuser@localhost-live:~$ pwd
/home/liveuser
liveuser@localhost-live:~$ cd Documents
liveuser@localhost-live:~/Documents$ cd /usr/local
liveuser@localhost-live:/usr/local$
```

Рис. 4.2.4 В терминале выведен каталог local, подкаталог usr корневого каталога.

Вернувшись в домашний каталог, я ввожу команду ls, чтобы вывести список файлов выбранной директории. (рис. 4.2.5)

A terminal window titled 'liveuser@localhost-live:~' with search, menu, and close icons in the title bar. The terminal shows a series of commands and their outputs: 'cd' leads to '/home/liveuser', 'pwd' confirms this, 'cd Documents' moves to the Documents directory, 'cd /usr/local' moves there, 'cd -' returns to Documents, 'cd ..' returns to the home directory, and 'ls' lists the contents: Desktop, Documents, Downloads, Music, Pictures, Public, Templates, and Videos.

```
liveuser@localhost-live:~$ cd
liveuser@localhost-live:~$ pwd
/home/liveuser
liveuser@localhost-live:~$ cd Documents
liveuser@localhost-live:~/Documents$ cd /usr/local
liveuser@localhost-live:/usr/local$ cd -
/home/liveuser/Documents
liveuser@localhost-live:~/Documents$ cd ..
liveuser@localhost-live:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
liveuser@localhost-live:~$
```

Рис. 4.2.5. В терминале выведен список файлов домашней директории.

Чтобы убедиться в правильности отображения, я открываю тот же домашний каталог в файловом менеджере графического интерфейса ОС Fedora Workstation 40, на (рис. 4.2.6) видно, что файлы в выводе интерпретатора совпадают с домашним каталогом в файловом менеджере.

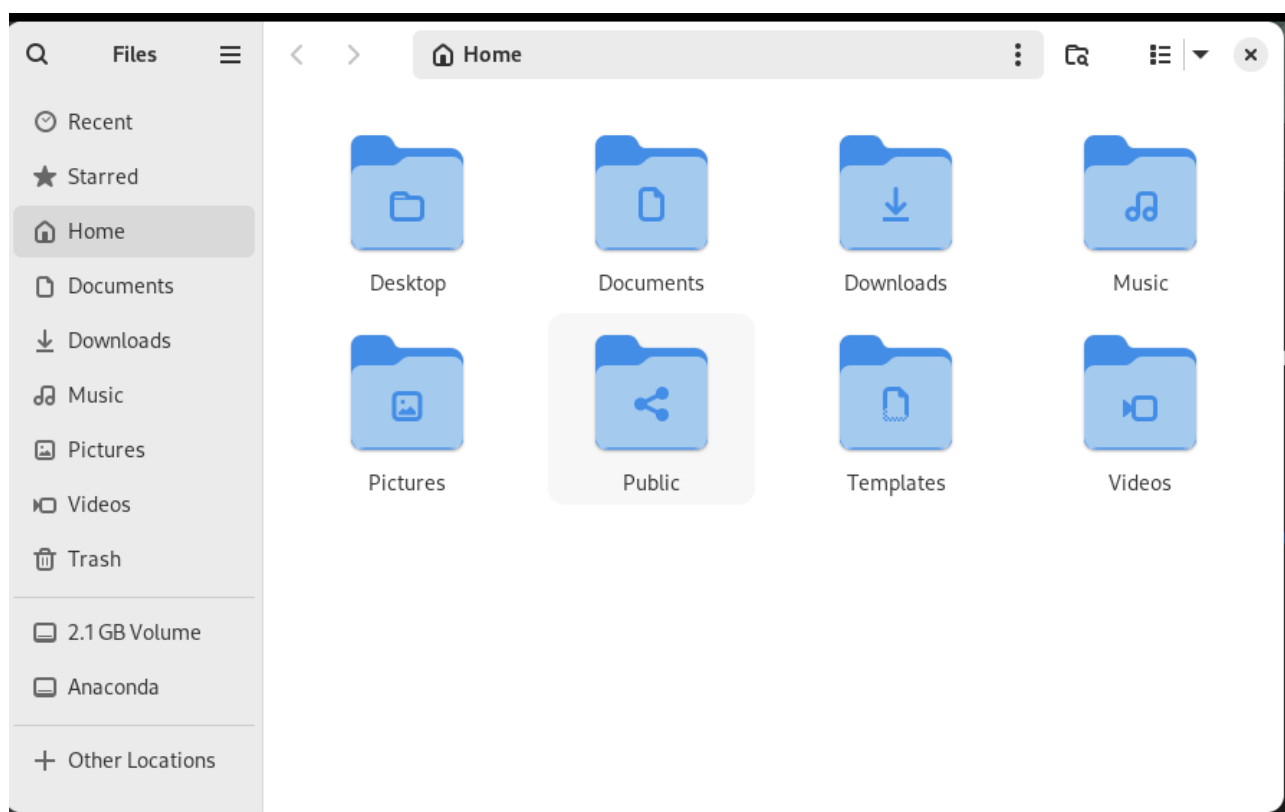
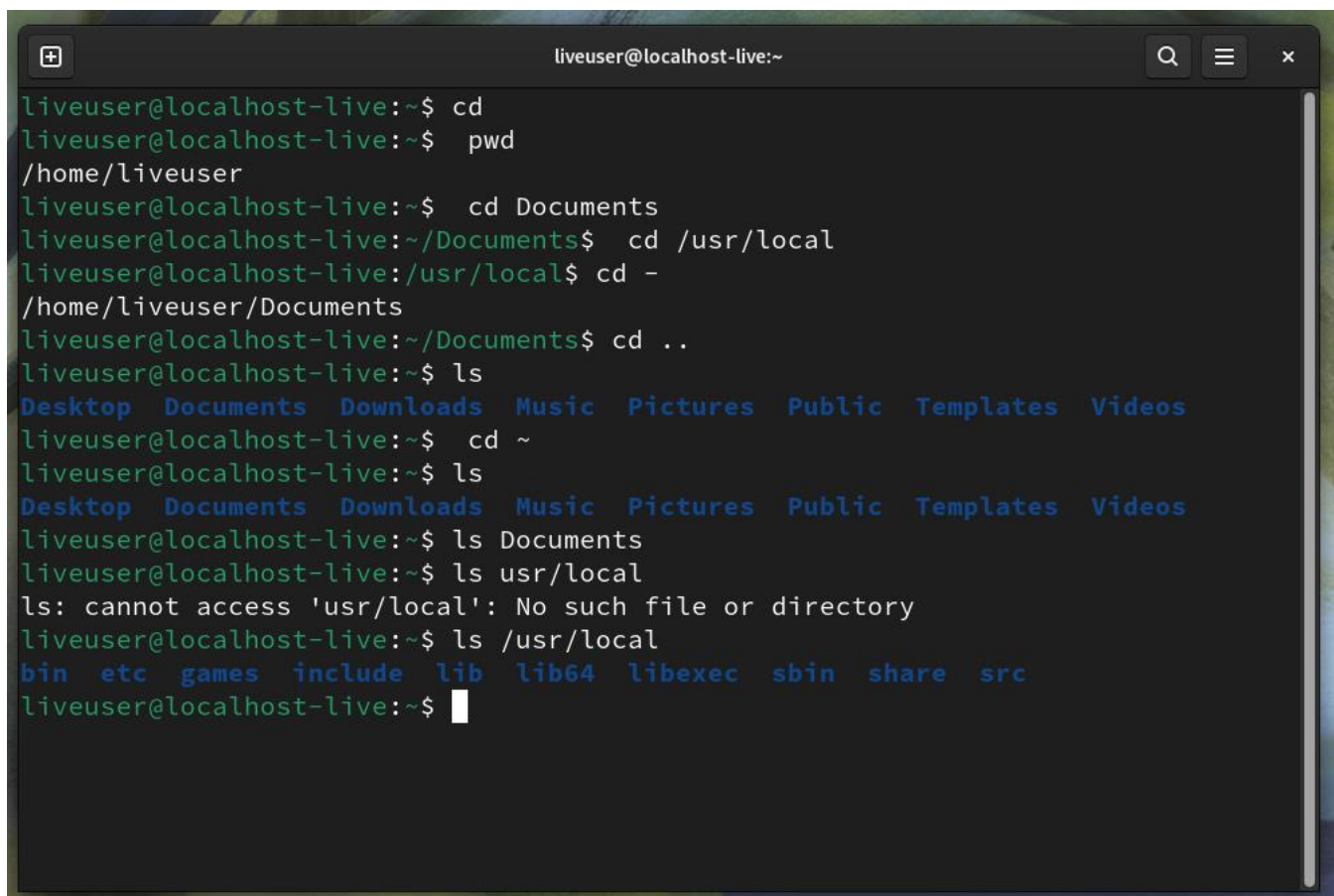


Рис. 4.2.6. Файловый менеджер с открытой домашней директорией.

В процессе работы с терминалом я научился использовать команду `ls` с указанием абсолютных и относительных путей. (рис. 4.2.7)

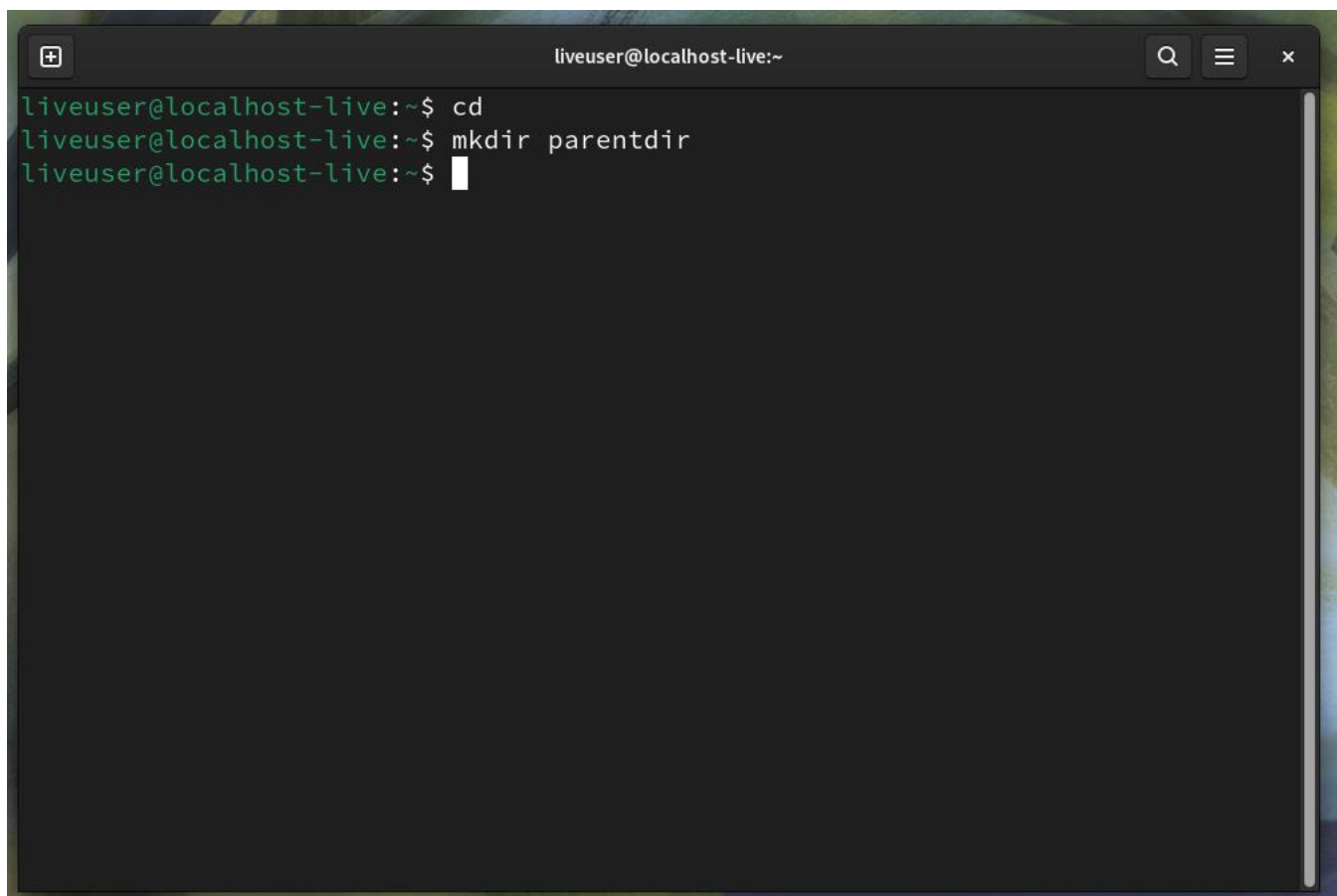
A terminal window titled 'liveuser@localhost-live:~' with search, menu, and close icons in the title bar. The terminal shows a series of commands and their outputs: 'cd' leads to '/home/liveuser'; 'pwd' confirms the current directory; 'cd Documents' moves to the Documents directory; 'cd /usr/local' moves to /usr/local; 'cd -' returns to the previous directory; 'cd ..' returns to the home directory. The 'ls' command is run twice, both times showing the same output: Desktop, Documents, Downloads, Music, Pictures, Public, Templates, Videos. Then, 'ls Documents' and 'ls usr/local' are run. The latter results in an error: 'ls: cannot access 'usr/local': No such file or directory'. Finally, 'ls /usr/local' is run, showing the output: bin, etc, games, include, lib, lib64, libexec, sbin, share, src.

```
liveuser@localhost-live:~$ cd
liveuser@localhost-live:~$ pwd
/home/liveuser
liveuser@localhost-live:~$ cd Documents
liveuser@localhost-live:~/Documents$ cd /usr/local
liveuser@localhost-live:/usr/local$ cd -
/home/liveuser/Documents
liveuser@localhost-live:~/Documents$ cd ..
liveuser@localhost-live:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
liveuser@localhost-live:~$ cd ~
liveuser@localhost-live:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
liveuser@localhost-live:~$ ls Documents
liveuser@localhost-live:~$ ls usr/local
ls: cannot access 'usr/local': No such file or directory
liveuser@localhost-live:~$ ls /usr/local
bin  etc  games  include  lib  lib64  libexec  sbin  share  src
liveuser@localhost-live:~$
```

Рис. 4.2.7 В терминале демонстрируется вывод команды ls с разными путями.

## 4.3 Создание пустых каталогов и файлов

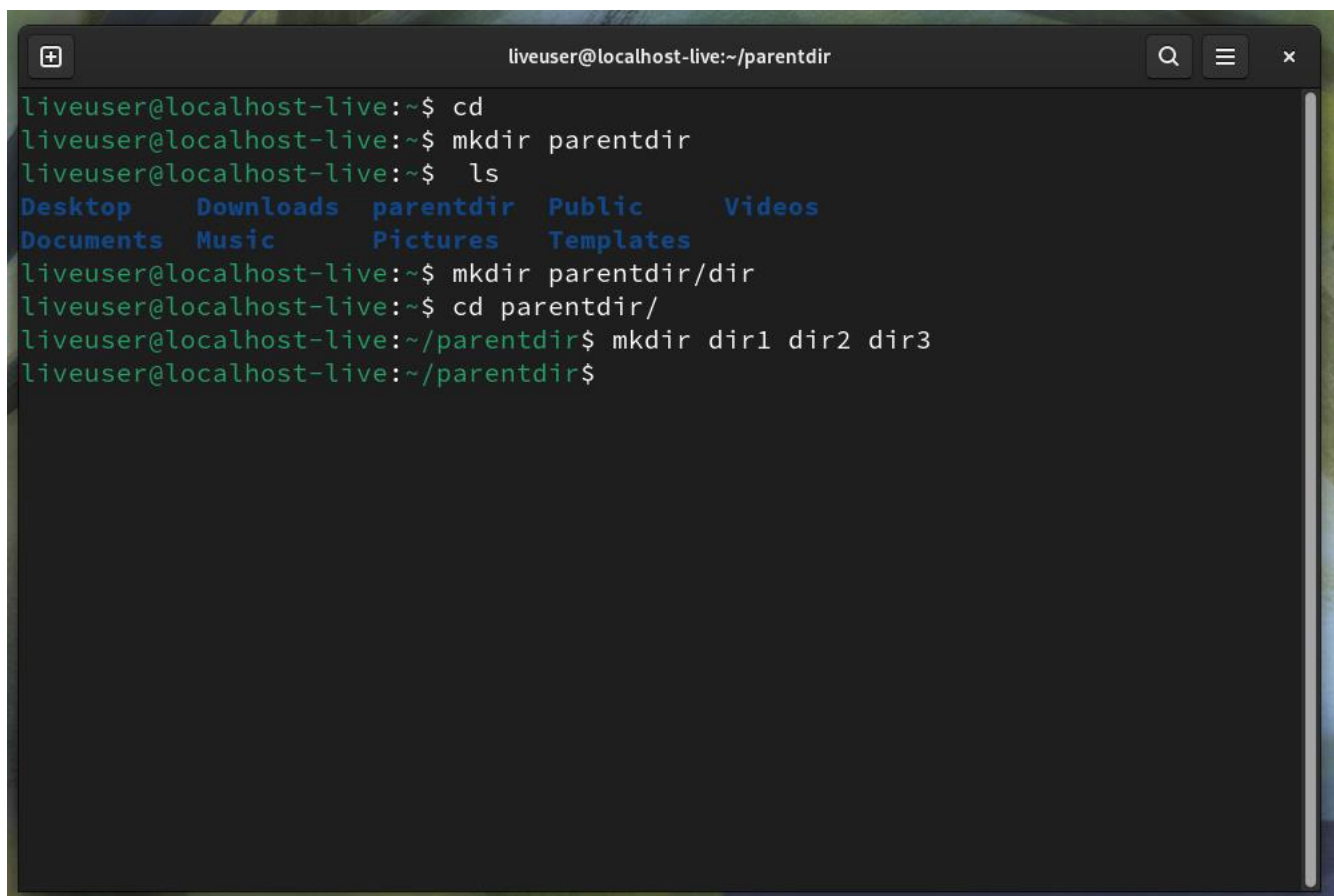
Перейдя обратно в домашнюю директорию, я создал папку командой mkdir (рис. 4.3.1)

A terminal window with a dark background and light green text. The window title is 'liveuser@localhost-live:~'. The terminal shows three lines of commands: 'cd', 'mkdir parentdir', and a blank line with a cursor. The window has standard macOS-style window controls (red, yellow, green buttons) and a search icon in the top right corner.

```
liveuser@localhost-live:~$ cd
liveuser@localhost-live:~$ mkdir parentdir
liveuser@localhost-live:~$
```

Рис. 4.3.1 Демонстрация создания директории в терминале в домашней директории.

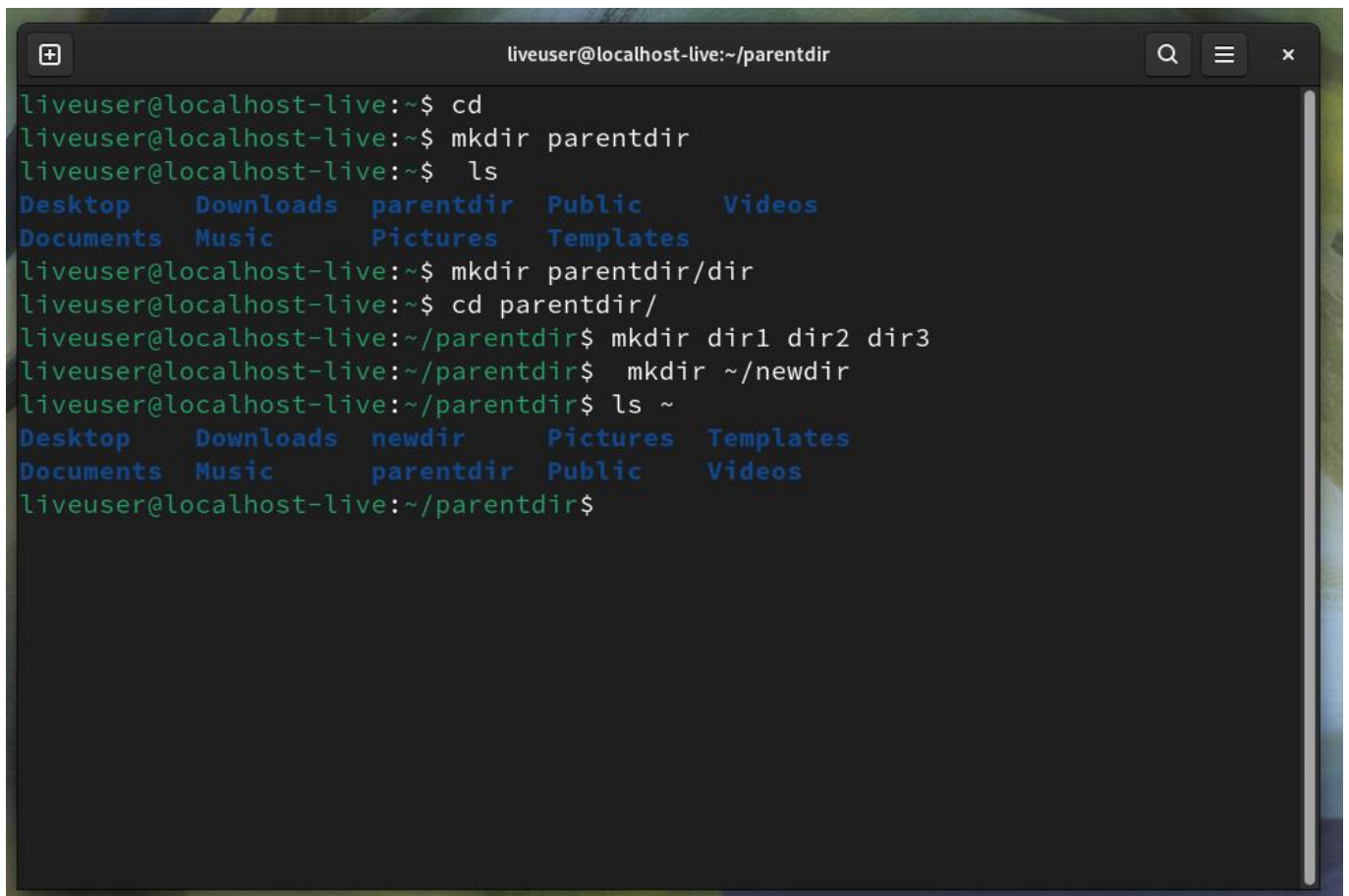
Команда `mkdir` может принимать сразу несколько аргументов (рис. 4.3.2), таким образом я создаю дополнительно еще 3 подкаталога в только что созданной директории.

A terminal window titled 'liveuser@localhost-live:~/parentdir'. The window shows a series of commands and their outputs. The user starts at the root directory, creates a 'parentdir' directory, and lists the contents of the home directory. Then, they create a subdirectory 'dir' inside 'parentdir', move into it, and finally create three more subdirectories: 'dir1', 'dir2', and 'dir3'.

```
liveuser@localhost-live:~$ cd
liveuser@localhost-live:~$ mkdir parentdir
liveuser@localhost-live:~$ ls
Desktop  Downloads  parentdir  Public  Videos
Documents Music      Pictures   Templates
liveuser@localhost-live:~$ mkdir parentdir/dir
liveuser@localhost-live:~$ cd parentdir/
liveuser@localhost-live:~/parentdir$ mkdir dir1 dir2 dir3
liveuser@localhost-live:~/parentdir$
```

Рис. 4.3.2 Использование нескольких аргументов в команде mkdir.

Команду mkdir можно так же использовать с указанием явного пути (рис. 4.3.3), правильность выполнения команду проверяю, отправив команду ls ~ в терминале.

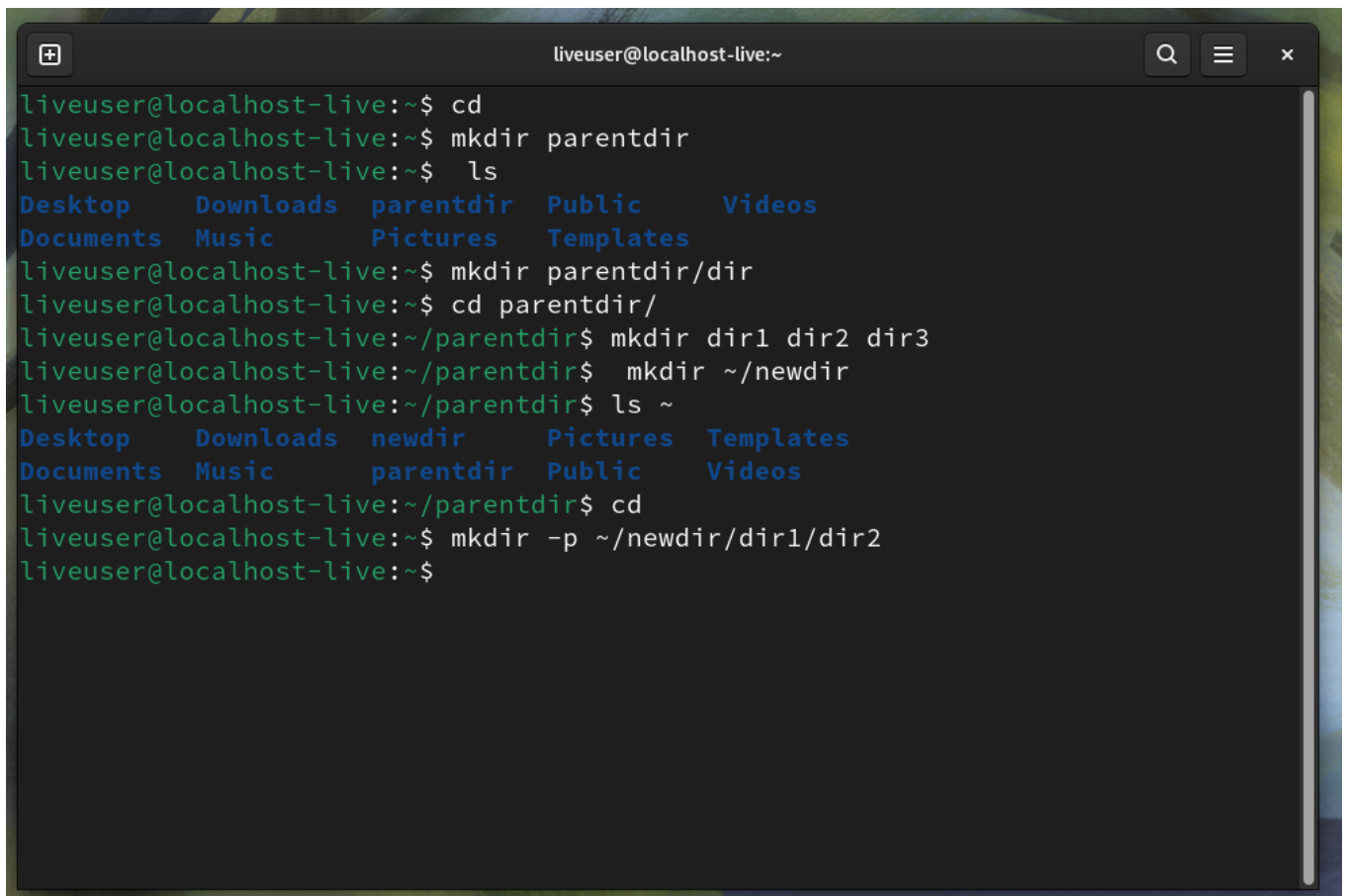
A terminal window titled 'liveuser@localhost-live:~/parentdir'. The terminal shows a sequence of commands: 'cd', 'mkdir parentdir', 'ls', 'mkdir parentdir/dir', 'cd parentdir/', 'mkdir dir1 dir2 dir3', 'mkdir ~/newdir', and 'ls ~'. The output of 'ls ~' shows a list of directories including Desktop, Downloads, parentdir, Public, Videos, Documents, Music, Pictures, and Templates, with 'newdir' added under Downloads. The terminal has a dark background with light green text. The window title bar includes a search icon, a menu icon, and a close button.

```
liveuser@localhost-live:~/parentdir
liveuser@localhost-live:~$ cd
liveuser@localhost-live:~$ mkdir parentdir
liveuser@localhost-live:~$ ls
Desktop    Downloads  parentdir  Public     Videos
Documents  Music      Pictures   Templates
liveuser@localhost-live:~$ mkdir parentdir/dir
liveuser@localhost-live:~$ cd parentdir/
liveuser@localhost-live:~/parentdir$ mkdir dir1 dir2 dir3
liveuser@localhost-live:~/parentdir$ mkdir ~/newdir
liveuser@localhost-live:~/parentdir$ ls ~
Desktop    Downloads  newdir     Pictures   Templates
Documents  Music      parentdir  Public     Videos
liveuser@localhost-live:~/parentdir$
```

Рис. 4.3.3 В терминале выведены файлы домашней директории с новым созданным каталогом.

Опция или аргумент -p (сокращенно от parents) позволяет создавать иерархическую цепочку подкаталогов, создаю такую. (рис 4.3.4)

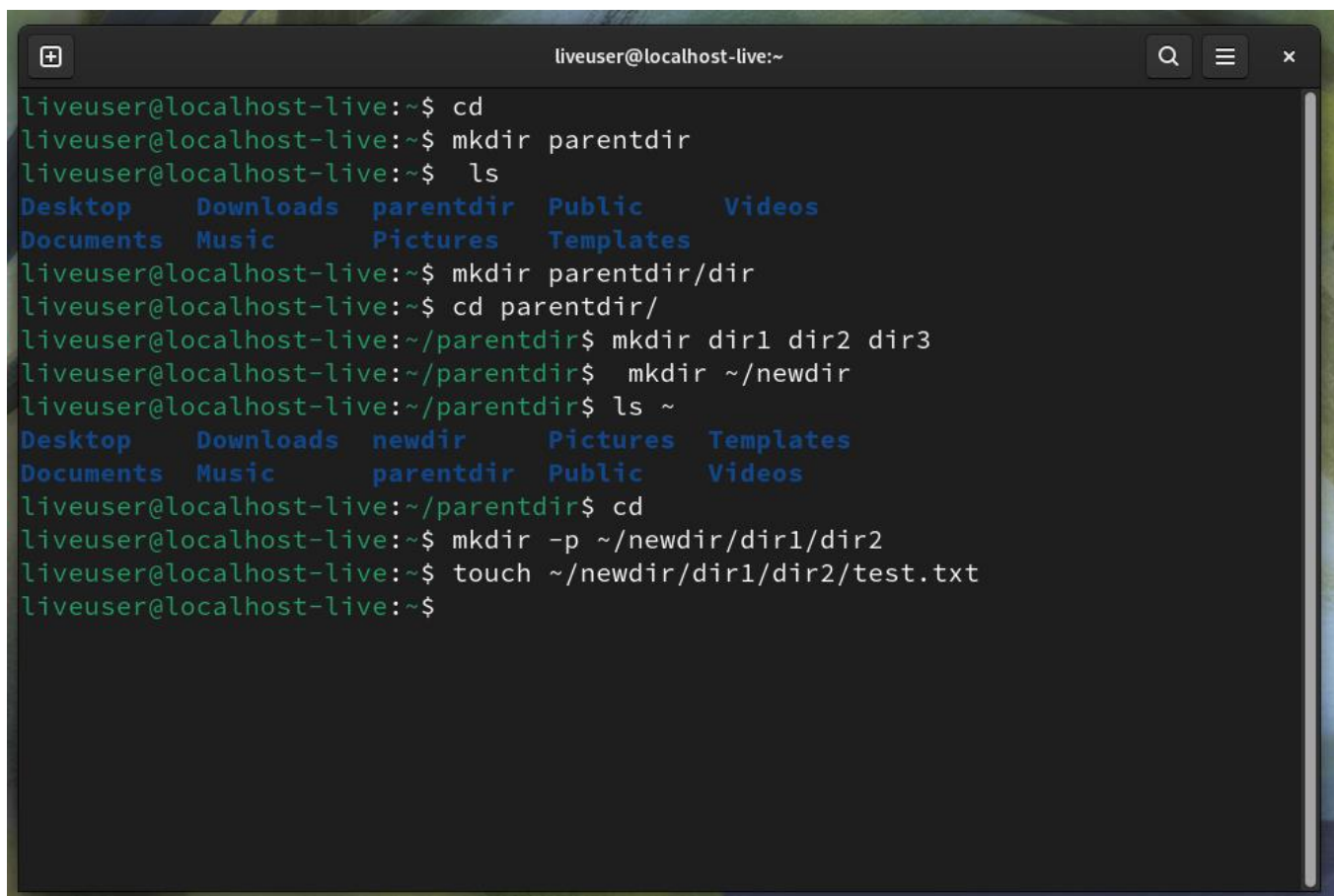


A terminal window titled 'liveuser@localhost-live:~' with search, menu, and close buttons in the title bar. The terminal shows a series of commands to create a directory hierarchy. The user starts in the home directory, creates 'parentdir', lists the contents, creates 'parentdir/dir', moves into it, creates 'dir1', 'dir2', and 'dir3', moves back to the parent directory, creates 'newdir' in the home directory, lists the home directory to show both 'parentdir' and 'newdir', moves into 'parentdir', and finally creates a nested directory structure '~/newdir/dir1/dir2' using the '-p' flag. The window has a vertical scrollbar on the right side.

```
liveuser@localhost-live:~$ cd
liveuser@localhost-live:~$ mkdir parentdir
liveuser@localhost-live:~$ ls
Desktop  Downloads  parentdir  Public  Videos
Documents Music      Pictures  Templates
liveuser@localhost-live:~$ mkdir parentdir/dir
liveuser@localhost-live:~$ cd parentdir/
liveuser@localhost-live:~/parentdir$ mkdir dir1 dir2 dir3
liveuser@localhost-live:~/parentdir$ mkdir ~/newdir
liveuser@localhost-live:~/parentdir$ ls ~
Desktop  Downloads  newdir  Pictures  Templates
Documents Music      parentdir  Public  Videos
liveuser@localhost-live:~/parentdir$ cd
liveuser@localhost-live:~$ mkdir -p ~/newdir/dir1/dir2
liveuser@localhost-live:~$
```

Рис. 4.3.4 Создание иерархической цепочки подкаталогов.

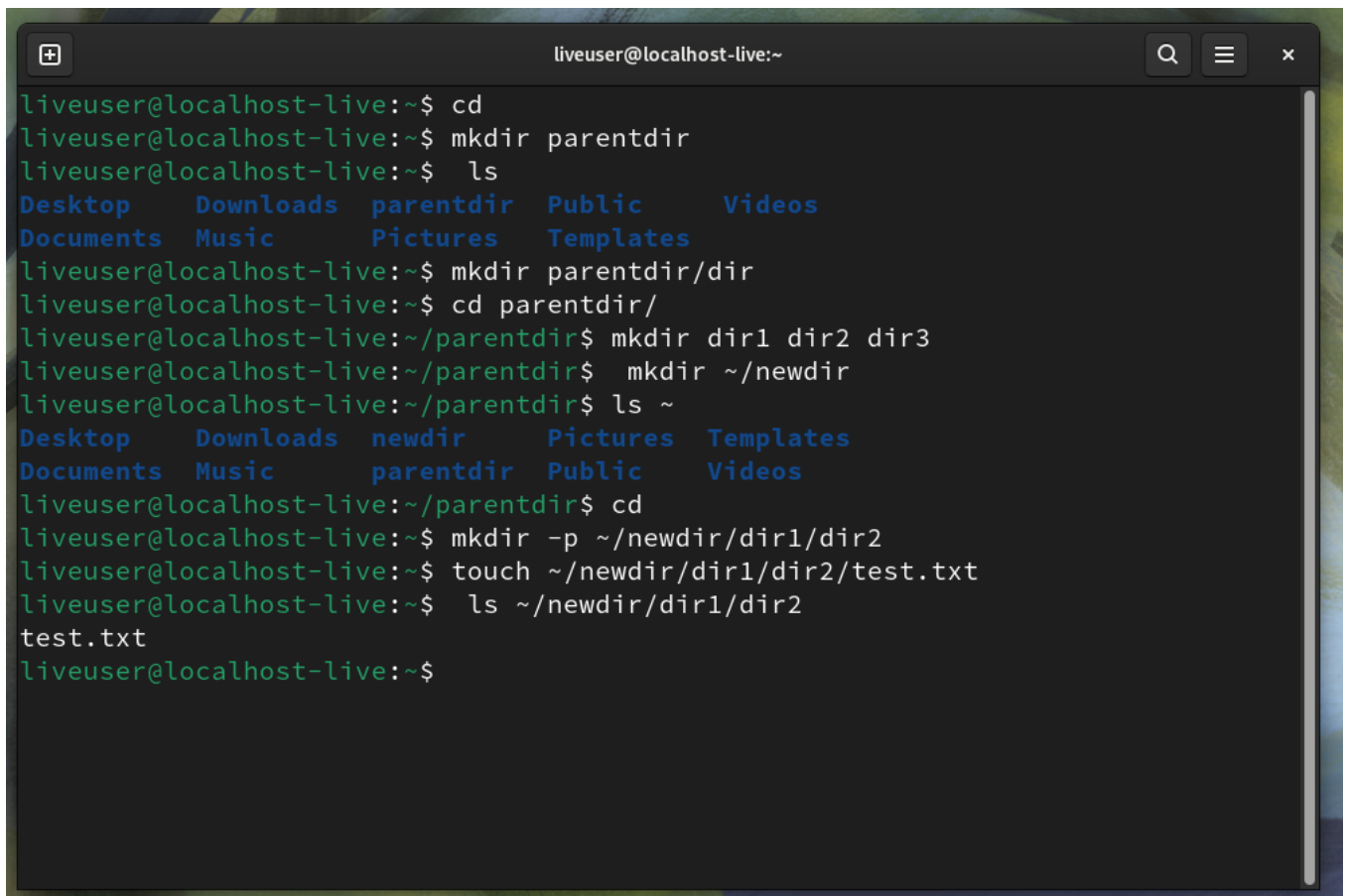
Создание файлов осуществляется командой `touch`, создаю файл `test.txt` в только что созданном каталоге. (рис. 4.3.5)

A terminal window titled 'liveuser@localhost-live:~' with search, menu, and close icons in the title bar. The terminal shows a series of commands and their outputs. The user starts in the home directory, creates a 'parentdir', and lists its contents. Then, they create a subdirectory 'dir' inside 'parentdir', move into it, and create three more subdirectories: 'dir1', 'dir2', and 'dir3'. They then create a 'newdir' in the home directory and list its contents, showing 'parentdir' and 'newdir'. Finally, they create a nested directory structure '~/newdir/dir1/dir2' and create a file 'test.txt' inside it using the 'touch' command.

```
liveuser@localhost-live:~$ cd
liveuser@localhost-live:~$ mkdir parentdir
liveuser@localhost-live:~$ ls
Desktop  Downloads  parentdir  Public  Videos
Documents Music      Pictures  Templates
liveuser@localhost-live:~$ mkdir parentdir/dir
liveuser@localhost-live:~$ cd parentdir/
liveuser@localhost-live:~/parentdir$ mkdir dir1 dir2 dir3
liveuser@localhost-live:~/parentdir$ mkdir ~/newdir
liveuser@localhost-live:~/parentdir$ ls ~
Desktop  Downloads  newdir  Pictures  Templates
Documents Music      parentdir  Public  Videos
liveuser@localhost-live:~/parentdir$ cd
liveuser@localhost-live:~$ mkdir -p ~/newdir/dir1/dir2
liveuser@localhost-live:~$ touch ~/newdir/dir1/dir2/test.txt
liveuser@localhost-live:~$
```

Рис. 4.3.5 Создание файлов с помощью команды touch.

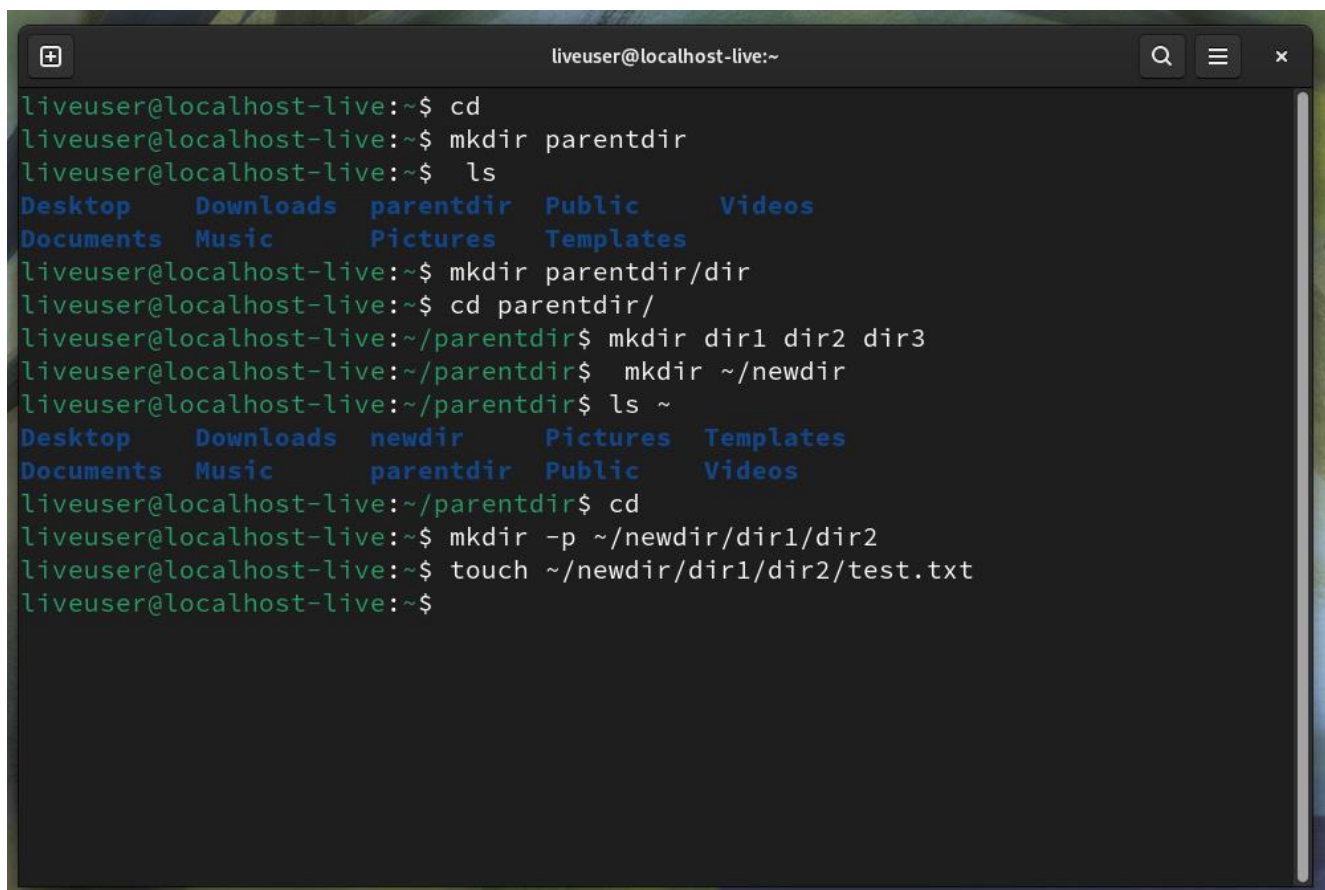
В корректности исполнения команды убеждаюсь, отправив команду ls интерпретатору (рис. 4.3.6)



```
liveuser@localhost-live:~$ cd
liveuser@localhost-live:~$ mkdir parentdir
liveuser@localhost-live:~$ ls
Desktop  Downloads  parentdir  Public     Videos
Documents Music      Pictures   Templates
liveuser@localhost-live:~$ mkdir parentdir/dir
liveuser@localhost-live:~$ cd parentdir/
liveuser@localhost-live:~/parentdir$ mkdir dir1 dir2 dir3
liveuser@localhost-live:~/parentdir$ mkdir ~/newdir
liveuser@localhost-live:~/parentdir$ ls ~
Desktop  Downloads  newdir     Pictures   Templates
Documents Music      parentdir  Public     Videos
liveuser@localhost-live:~/parentdir$ cd
liveuser@localhost-live:~$ mkdir -p ~/newdir/dir1/dir2
liveuser@localhost-live:~$ touch ~/newdir/dir1/dir2/test.txt
liveuser@localhost-live:~$ ls ~/newdir/dir1/dir2
test.txt
liveuser@localhost-live:~$
```

Рис. 4.3.6 Демонстрация корректности исполнения команды touch с помощью ls.

Далее я хочу продемонстрировать работу команд `cp` и `mv`, для этого создаю следующие файлы и подкаталоги. (рис. 4.3.7)

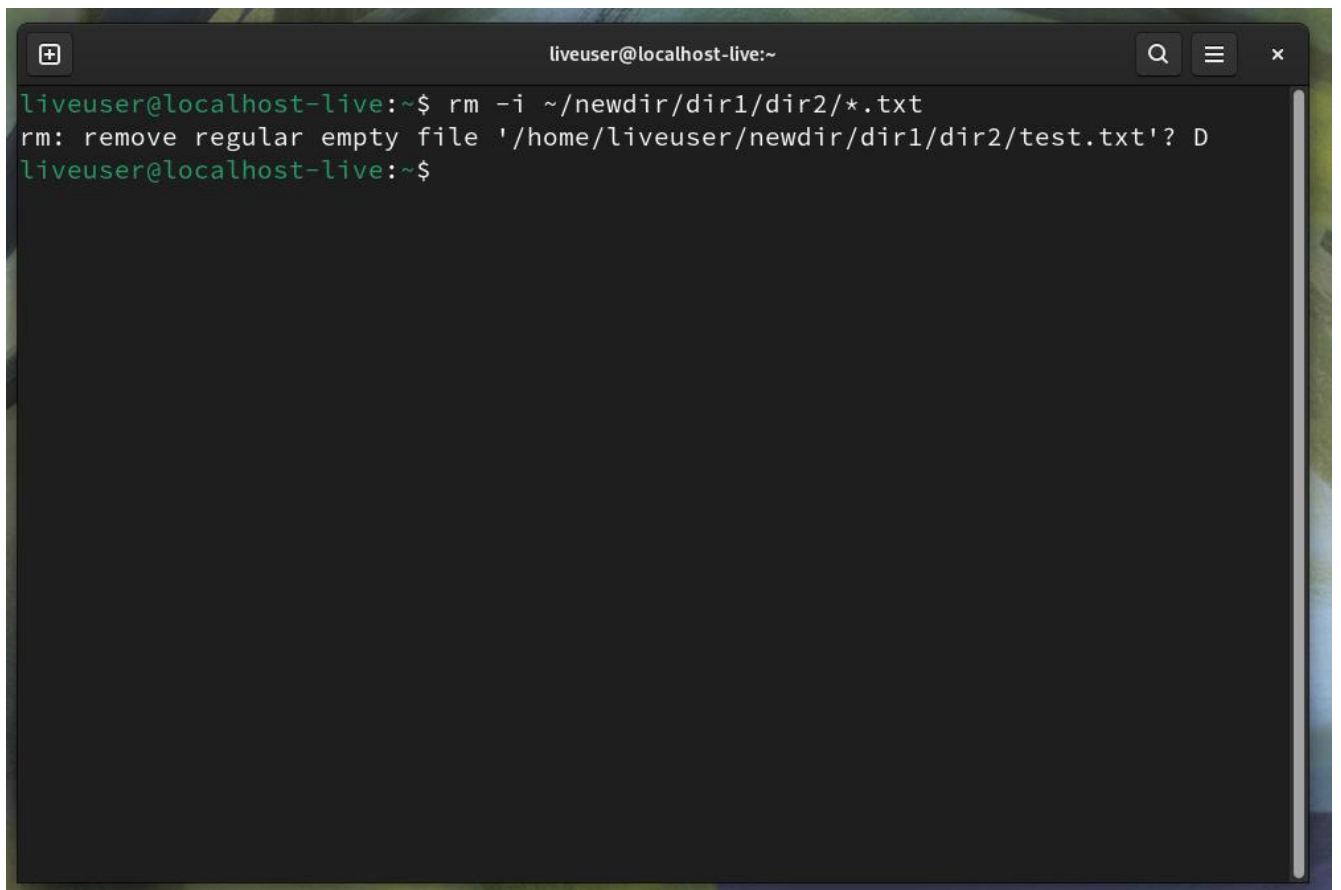
A terminal window titled 'liveuser@localhost-live:~' with search, menu, and close icons in the title bar. The terminal shows a sequence of commands: 'cd', 'mkdir parentdir', 'ls' (listing Desktop, Downloads, parentdir, Public, Videos, Documents, Music, Pictures, Templates), 'mkdir parentdir/dir', 'cd parentdir/', 'mkdir dir1 dir2 dir3', 'mkdir ~/newdir', 'ls ~' (listing Desktop, Downloads, newdir, Pictures, Templates, Documents, Music, parentdir, Public, Videos), 'cd', 'mkdir -p ~/newdir/dir1/dir2', 'touch ~/newdir/dir1/dir2/test.txt', and finally '\$'.

```
liveuser@localhost-live:~$ cd
liveuser@localhost-live:~$ mkdir parentdir
liveuser@localhost-live:~$ ls
Desktop  Downloads  parentdir  Public    Videos
Documents Music      Pictures   Templates
liveuser@localhost-live:~$ mkdir parentdir/dir
liveuser@localhost-live:~$ cd parentdir/
liveuser@localhost-live:~/parentdir$ mkdir dir1 dir2 dir3
liveuser@localhost-live:~/parentdir$ mkdir ~/newdir
liveuser@localhost-live:~/parentdir$ ls ~
Desktop  Downloads  newdir     Pictures  Templates
Documents Music      parentdir  Public    Videos
liveuser@localhost-live:~/parentdir$ cd
liveuser@localhost-live:~$ mkdir -p ~/newdir/dir1/dir2
liveuser@localhost-live:~$ touch ~/newdir/dir1/dir2/test.txt
liveuser@localhost-live:~$
```

Рис. 4.3.7. Создание подкаталогов и файла в конечном.

## 4.4 Перемещение и удаление файлов или каталогов

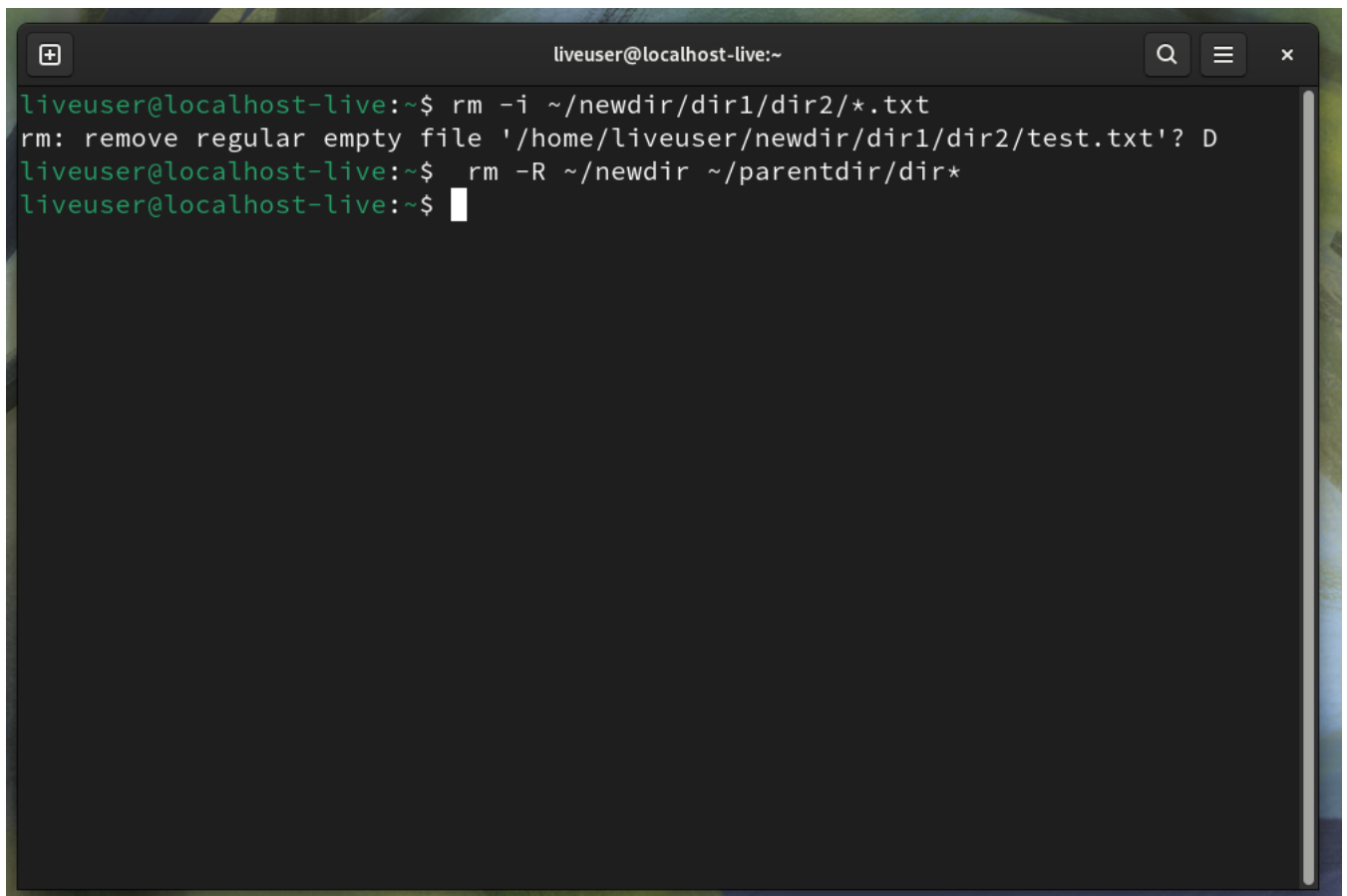
Команда `rm` удаляет файлы и/или каталоги. Для удаления пустых каталогов используется команда `rmdir`. Запросив подтверждение на удаление каждого файла, дописав ключ `-i`, я удаляю в подкаталоге все файлы, имеющие расширение `.txt`. (рис 4.4.1)

A terminal window titled 'liveuser@localhost-live:~' with search, menu, and close icons in the title bar. The terminal shows a command prompt where the user enters 'rm -i ~/newdir/dir1/dir2/\*.txt'. The system responds with 'rm: remove regular empty file '/home/liveuser/newdir/dir1/dir2/test.txt'? D'. The user then enters a second prompt.

```
liveuser@localhost-live:~$ rm -i ~/newdir/dir1/dir2/*.txt
rm: remove regular empty file '/home/liveuser/newdir/dir1/dir2/test.txt'? D
liveuser@localhost-live:~$
```

Рис. 4.4.1 Использование команды `rm` с ключом `-i` для удаления файлов.

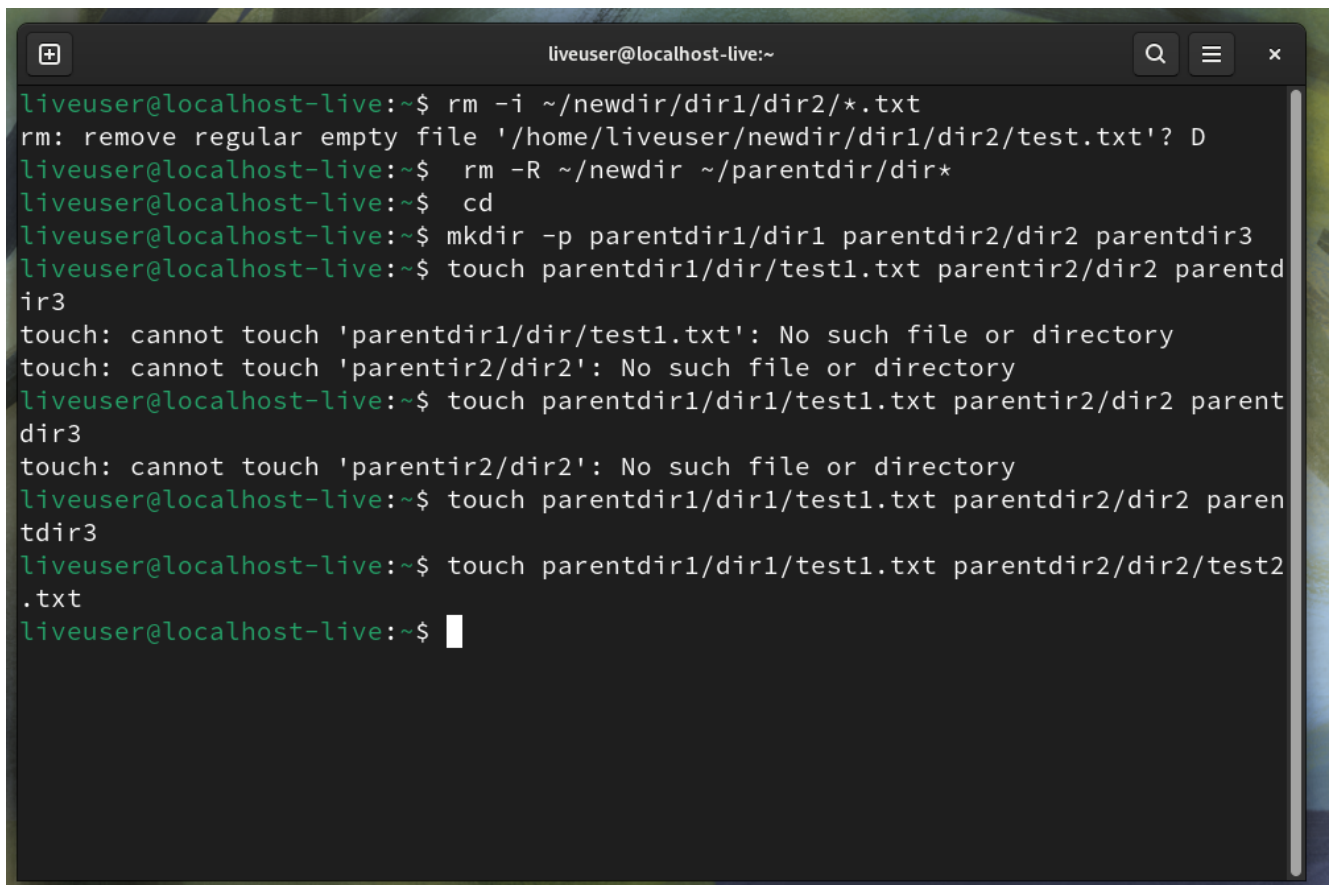
Рекурсивно (с помощью ключа `-R`) удаляю все файлы из каталога `newdir`, а также файлы, чьи имена начинаются с `dir` в каталоге `parentdir`. (рис. 4.2.2)

A terminal window titled 'liveuser@localhost-live:~' with search, menu, and close icons in the title bar. The terminal shows three lines of text: a command to remove a specific file with the -i flag, a confirmation message from 'rm', and a command to recursively remove a directory and its contents with the -R flag.

```
liveuser@localhost-live:~$ rm -i ~/newdir/dir1/dir2/*.txt
rm: remove regular empty file '/home/liveuser/newdir/dir1/dir2/test.txt'? D
liveuser@localhost-live:~$ rm -R ~/newdir ~/parentdir/dir*
liveuser@localhost-live:~$
```

Рис. 4.2.2 Использование команды `rm` с ключом `-R` для рекурсивного удаления файлов.

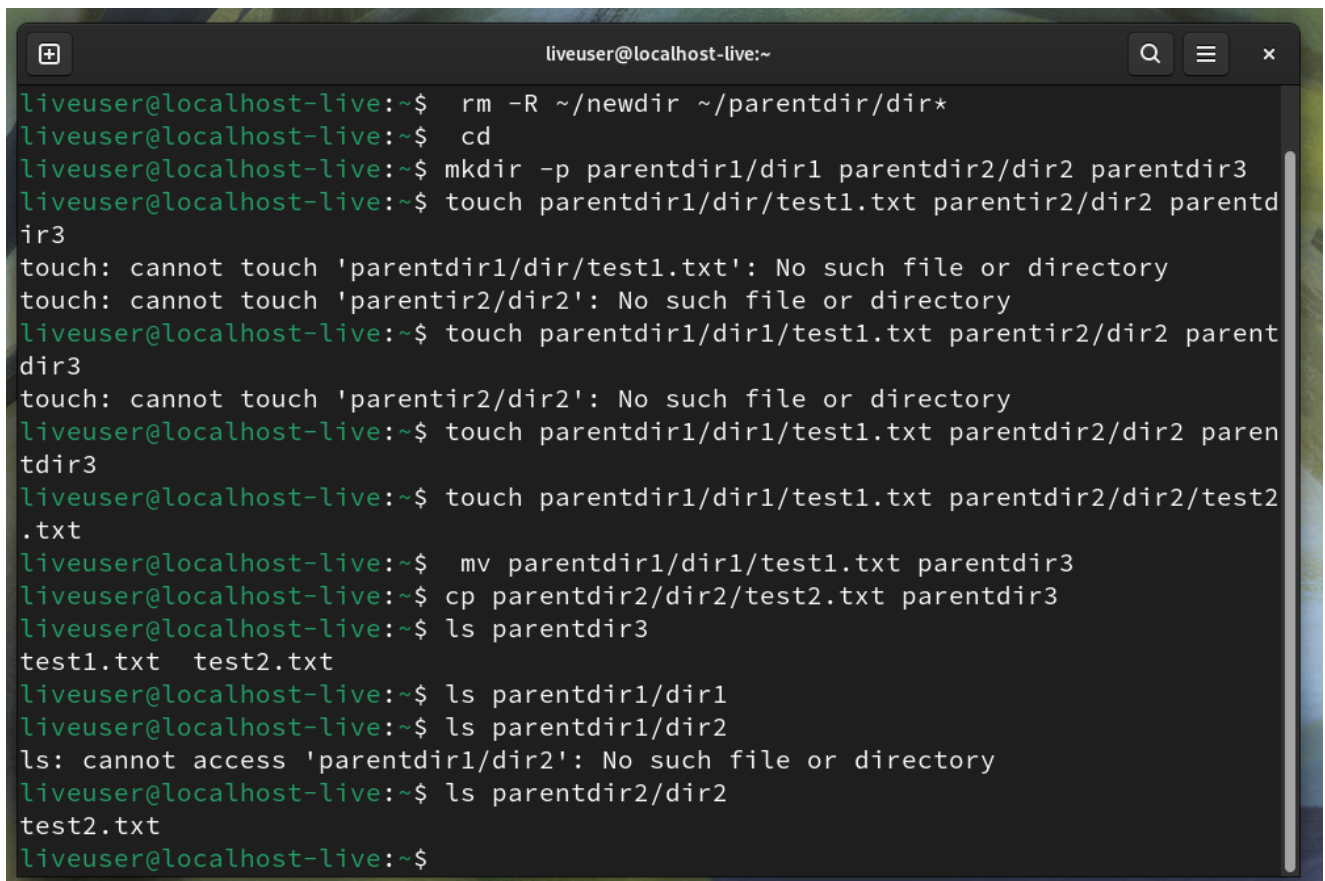
Далее я проверяю команды `mv` и `cp`, для этого создаю следующие файлы и подкаталоги в домашней директории. (рис. 4.2.3)

A terminal window titled 'liveuser@localhost-live:~' with search, menu, and close buttons in the title bar. The terminal shows a series of commands and their outputs. The user starts by removing a file with 'rm -i ~/newdir/dir1/dir2/\*.txt', which prompts for confirmation. Then they remove a directory with 'rm -R ~/newdir ~/parentdir/dir\*'. They change the directory with 'cd'. Next, they create three directories with 'mkdir -p parentdir1/dir1 parentdir2/dir2 parentdir3'. They then attempt to create files with 'touch parentdir1/dir/test1.txt parentdir2/dir2 parentdir3', but receive errors because the directories 'parentdir1/dir' and 'parentdir2/dir2' do not exist. They then try 'touch parentdir1/dir1/test1.txt parentdir2/dir2 parentdir3', which also fails for the same reason. Finally, they successfully create files with 'touch parentdir1/dir1/test1.txt parentdir2/dir2/test2.txt'.

```
liveuser@localhost-live:~$ rm -i ~/newdir/dir1/dir2/*.txt
rm: remove regular empty file '/home/liveuser/newdir/dir1/dir2/test.txt'? D
liveuser@localhost-live:~$ rm -R ~/newdir ~/parentdir/dir*
liveuser@localhost-live:~$ cd
liveuser@localhost-live:~$ mkdir -p parentdir1/dir1 parentdir2/dir2 parentdir3
liveuser@localhost-live:~$ touch parentdir1/dir/test1.txt parentdir2/dir2 parentdir3
touch: cannot touch 'parentdir1/dir/test1.txt': No such file or directory
touch: cannot touch 'parentdir2/dir2': No such file or directory
liveuser@localhost-live:~$ touch parentdir1/dir1/test1.txt parentdir2/dir2 parentdir3
touch: cannot touch 'parentdir2/dir2': No such file or directory
liveuser@localhost-live:~$ touch parentdir1/dir1/test1.txt parentdir2/dir2 parentdir3
touch: cannot touch 'parentdir2/dir2': No such file or directory
liveuser@localhost-live:~$ touch parentdir1/dir1/test1.txt parentdir2/dir2/test2.txt
liveuser@localhost-live:~$
```

Рис. 4.2.3 Создание подкаталогов и файлов в нем.

Используя команды `cp` и `mv`, я копирую и перемещаю только что созданные файлы в соседние подкаталоги, с помощью команды `ls` убеждаюсь в том, что выполнил задание верно. (рис 4.2.4)

A terminal window titled 'liveuser@localhost-live:~' with search, menu, and close icons in the title bar. The terminal shows a series of commands and their outputs. The user starts by removing a directory, then creates a hierarchy of directories. They attempt to create files with 'touch' but get errors for non-existent paths. After correcting the paths, they create two files. Then, they move one file and copy another. Finally, they use 'ls' to verify the file locations.

```
liveuser@localhost-live:~$ rm -R ~/newdir ~/parentdir/dir*
liveuser@localhost-live:~$ cd
liveuser@localhost-live:~$ mkdir -p parentdir1/dir1 parentdir2/dir2 parentdir3
liveuser@localhost-live:~$ touch parentdir1/dir/test1.txt parentir2/dir2 parentd
ir3
touch: cannot touch 'parentdir1/dir/test1.txt': No such file or directory
touch: cannot touch 'parentir2/dir2': No such file or directory
liveuser@localhost-live:~$ touch parentdir1/dir1/test1.txt parentir2/dir2 paren
tdir3
touch: cannot touch 'parentir2/dir2': No such file or directory
liveuser@localhost-live:~$ touch parentdir1/dir1/test1.txt parentdir2/dir2 paren
tdir3
liveuser@localhost-live:~$ touch parentdir1/dir1/test1.txt parentdir2/dir2/test2
.txt
liveuser@localhost-live:~$ mv parentdir1/dir1/test1.txt parentdir3
liveuser@localhost-live:~$ cp parentdir2/dir2/test2.txt parentdir3
liveuser@localhost-live:~$ ls parentdir3
test1.txt  test2.txt
liveuser@localhost-live:~$ ls parentdir1/dir1
liveuser@localhost-live:~$ ls parentdir1/dir2
ls: cannot access 'parentdir1/dir2': No such file or directory
liveuser@localhost-live:~$ ls parentdir2/dir2
test2.txt
liveuser@localhost-live:~$
```

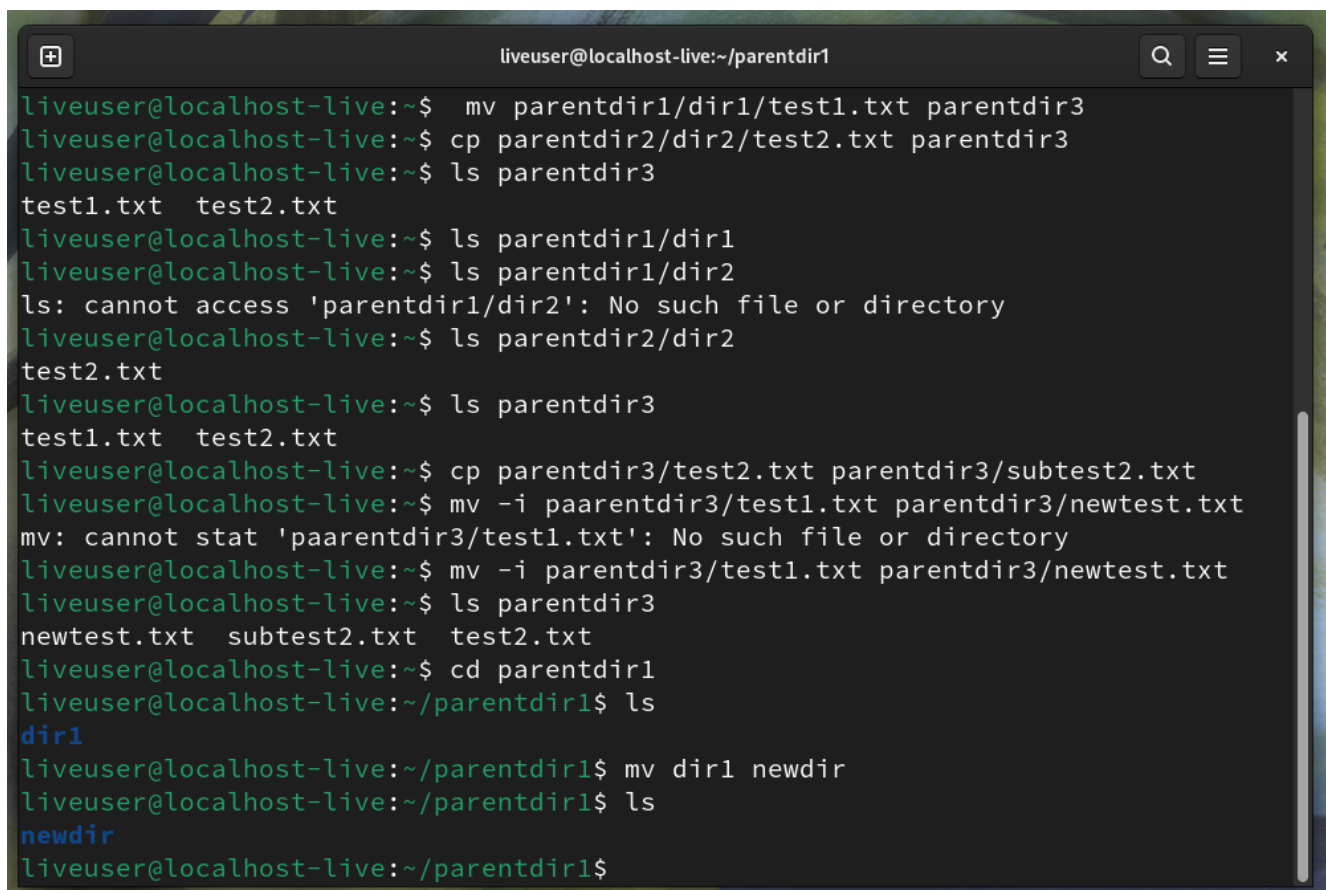
Рис. 4.2.4 Демонстрация работы команд `cp` и `mv`.

Также команда `mv` может быть использована для переименования файлов и каталогов, а команда `cp` позволяет сделать копию файла с новым именем. Переименовываю файл `test1.txt` из каталога `parentdir3` в `newtest.txt`, запрашивая подтверждение перед перезаписью (рис. 4.2.5)



```
liveuser@localhost-live:~  
dir3  
touch: cannot touch 'parentir2/dir2': No such file or directory  
liveuser@localhost-live:~$ touch parentdir1/dir1/test1.txt parentdir2/dir2 paren  
tdir3  
liveuser@localhost-live:~$ touch parentdir1/dir1/test1.txt parentdir2/dir2/test2  
.txt  
liveuser@localhost-live:~$ mv parentdir1/dir1/test1.txt parentdir3  
liveuser@localhost-live:~$ cp parentdir2/dir2/test2.txt parentdir3  
liveuser@localhost-live:~$ ls parentdir3  
test1.txt  test2.txt  
liveuser@localhost-live:~$ ls parentdir1/dir1  
liveuser@localhost-live:~$ ls parentdir1/dir2  
ls: cannot access 'parentdir1/dir2': No such file or directory  
liveuser@localhost-live:~$ ls parentdir2/dir2  
test2.txt  
liveuser@localhost-live:~$ ls parentdir3  
test1.txt  test2.txt  
liveuser@localhost-live:~$ cp parentdir3/test2.txt parentdir3/subtest2.txt  
liveuser@localhost-live:~$ mv -i paarentdir3/test1.txt parentdir3/newtest.txt  
mv: cannot stat 'paarentdir3/test1.txt': No such file or directory  
liveuser@localhost-live:~$ mv -i parentdir3/test1.txt parentdir3/newtest.txt  
liveuser@localhost-live:~$ ls parentdir3  
newtest.txt  subtest2.txt  test2.txt  
liveuser@localhost-live:~$
```

Рис. 4.2.5 Демонстрация работы команд `cp` и `mv` для создания копии файла с новым именем и смены имени файла соответственно. Переименовываю каталог `dir1` в каталоге `parentdir1` в `newdir` (рис. 4.2.6)

A terminal window titled 'liveuser@localhost-live:~/parentdir1'. The window shows a series of commands and their outputs. The commands include moving a file, copying a file, listing files, and renaming a directory. The outputs show the results of these operations, including file names and error messages.

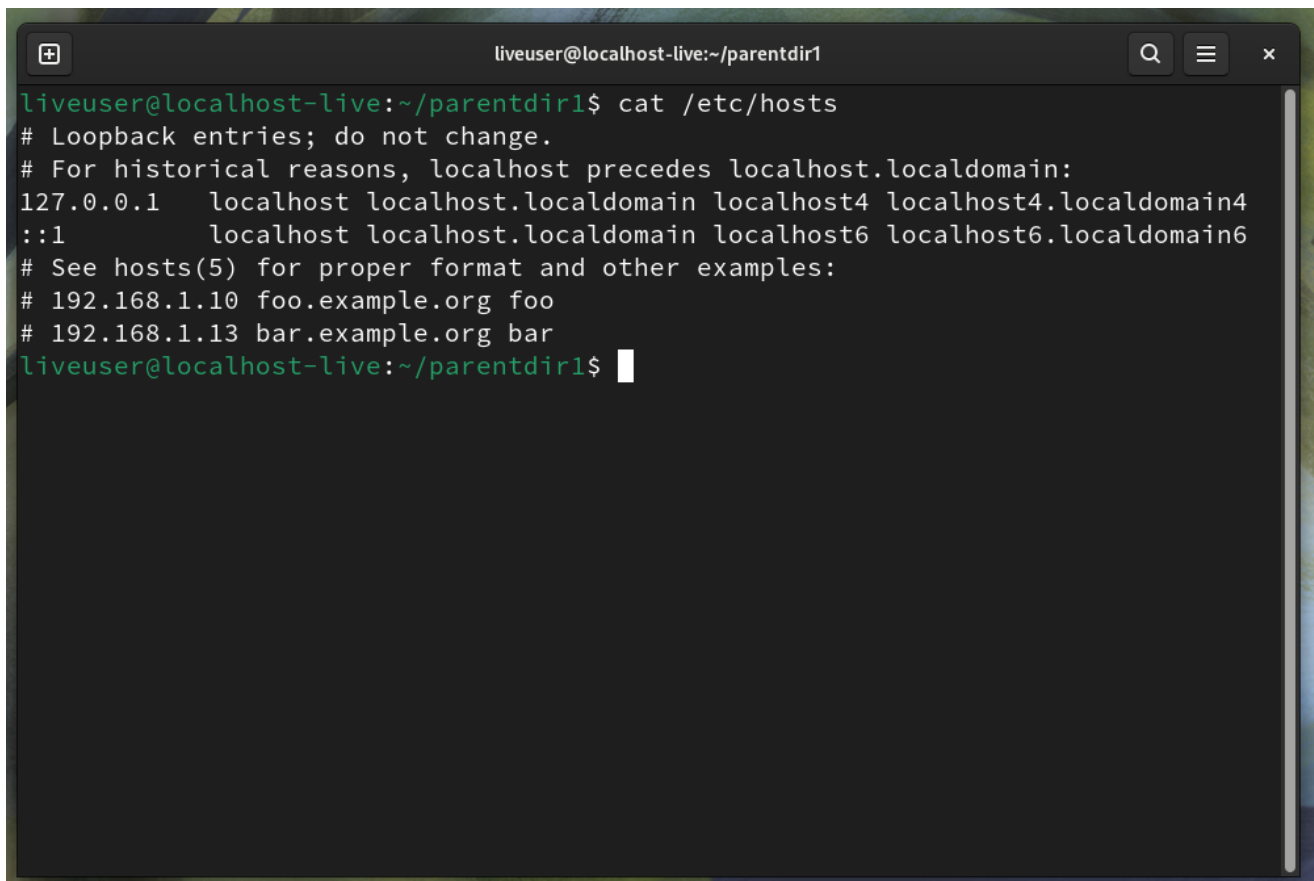
```
liveuser@localhost-live:~$ mv parentdir1/dir1/test1.txt parentdir3
liveuser@localhost-live:~$ cp parentdir2/dir2/test2.txt parentdir3
liveuser@localhost-live:~$ ls parentdir3
test1.txt  test2.txt
liveuser@localhost-live:~$ ls parentdir1/dir1
liveuser@localhost-live:~$ ls parentdir1/dir2
ls: cannot access 'parentdir1/dir2': No such file or directory
liveuser@localhost-live:~$ ls parentdir2/dir2
test2.txt
liveuser@localhost-live:~$ ls parentdir3
test1.txt  test2.txt
liveuser@localhost-live:~$ cp parentdir3/test2.txt parentdir3/subtest2.txt
liveuser@localhost-live:~$ mv -i paarentdir3/test1.txt parentdir3/newtest.txt
mv: cannot stat 'paarentdir3/test1.txt': No such file or directory
liveuser@localhost-live:~$ mv -i parentdir3/test1.txt parentdir3/newtest.txt
liveuser@localhost-live:~$ ls parentdir3
newtest.txt  subtest2.txt  test2.txt
liveuser@localhost-live:~$ cd parentdir1
liveuser@localhost-live:~/parentdir1$ ls
dir1
liveuser@localhost-live:~/parentdir1$ mv dir1 newdir
liveuser@localhost-live:~/parentdir1$ ls
newdir
liveuser@localhost-live:~/parentdir1$
```

Рис. 4.2.6 Использование команды mv для смены имени каталога.

### 4.3 Команда cat: вывод содержимого файлов

Команда cat объединяет файлы и выводит их на стандартный вывод – экран.

(рис. 4.3.1)

A terminal window with a dark background and light green text. The window title is 'liveuser@localhost-live:~/parentdir1'. The command 'cat /etc/hosts' has been executed, displaying the contents of the file. The output includes loopback entries for 127.0.0.1 and ::1, and two example entries for 192.168.1.10 and 192.168.1.13. The prompt 'liveuser@localhost-live:~/parentdir1\$' is visible at the bottom.

```
liveuser@localhost-live:~/parentdir1$ 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:~/parentdir1$
```

Рис. 4.3.1 Просмотр файла hosts с помощью команды cat.

## **5 Выводы**

На данной лабораторной работе я научился базовому набору команд для работы с терминалом, мне эти навыки пригодятся для дальнейшей более углубленной и продуктивной работы с ним