

# 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: Иванова Анастасия

Группа: НКАбд-07-25

МОСКВА

2025 г.

# Содержание

1. Цель работы.....	3
2. Теоретическое введение.....	3
2.1. Введение в GNU Linux.....	3
2.2. Введение в командную строку GNU Linux.....	4
2.3. Файловая структура GNU Linux: каталоги и файлы.....	5
2.4. Базовые команды bash.....	7
2.5. Полезные комбинации клавиш.....	8
3. Порядок выполнения работы.....	8
3.1. Техническое обеспечение.....	8
3.2. Перемещение по файловой системе.....	9
3.3 Создание пустых каталогов и файлов.....	13
3.4 Перемещение и удаление файлов или каталогов.....	15
3.5 Команда cat: вывод содержимого файлов.....	17
4. Задание для самостоятельной работы.....	18
5. Заключение.....	22

## **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. Теоретическое введение**

### **2.1. Введение в GNU Linux**

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).

## 2.2. Введение в командную строку GNU Linux

Работу ОС GNU Linux можно представить в виде функционирования множества взаимосвязанных процессов. При загрузке системы сначала запускается ядро, которое, в свою очередь, 1 Демидова А. В. Архитектура ЭВМ 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). Из командных оболочек GNU Linux наиболее популярны bash, csh, ksh, zsh. Команда `echo $SHELL` позволяет проверить, какая оболочка используется. В качестве предустановленной командной оболочки GNU Linux используется одна из наиболее распространённых разновидностей командной оболочки — bash (Bourne again shell). 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 командной строки (строки, оканчивающейся символом \$), по которому пользователь вводит команды (Рисунок 2.2):

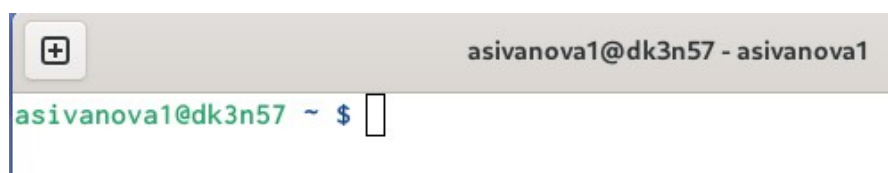


Рисунок 2.2 - Интерфейс командной оболочки

Это приглашение командной оболочки, которое несёт в себе информацию об

имени пользователя `iivanova`, имени компьютера `dk4n31` и текущем каталоге, в котором находится пользователь, 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` с ключом `-l`: **`iivanova@dk4n31:~$ ls -l documents`**.

В данном случае `ls` – это имя команды, `-l`–ключ, `documents` – аргумент. Команды, ключи и аргументы должны быть отделены друг от друга пробелом.

Ввод команды завершается нажатием клавиши **Enter** , после чего команда передаётся оболочке на исполнение. Результатом выполнения команды могут являться сообщения о ходе выполнения команды или об ошибках. Появление приглашения командной строки говорит о том, что выполнение команды завершено.

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 mclasserace mcookie mcview
mcat mcdiff mcheck
iivanova@dk4n31:~$ mc mcomp mcopy
```

### 2.3. Файловая структура GNU Linux: каталоги и файлы

Файловая система определяет способ организации, хранения и именования

данных на носителях информации в компьютерах и представляет собой иерархическую структуру в виде вложенных друг в друга каталогов (директорий), содержащих все файлы. В ОС Linux каталог, который является “вершиной” файловой системы, называется корневым каталогом, обозначается символом / и содержит все остальные каталоги и файлы.

В большинстве Linux-систем поддерживается стандарт иерархии файловой системы (Filesystem Hierarchy Standard, FHS), унифицирующий местонахождение файлов и каталогов. Это означает, что в корневом каталоге находятся только подкаталоги со стандартными именами и типами данных, которые могут попасть в тот или иной каталог. Так, в любой Linux-системе всегда есть каталоги /etc, /home, /usr, /bin и т.п. В табл.1.1 приведено краткое описание нескольких каталогов.

Таблица 1.1 - Описание некоторых каталогов файловой системы GNU Linux

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
/usr	Вторичная иерархия для данных пользователя; содержит большинство пользовательских приложений и утилит, используемых в многопользовательском режиме; может быть смонтирована по сети только для чтения и быть общей для нескольких машин

Обратиться к файлу, расположенному в каком-то каталоге, можно указав путь к нему. Существует несколько видов путей к файлу:

- полный или абсолютный путь — начинается от корня (/), образуется перечислением всех каталогов, разделённых прямым слешем (/), и завершается именем файла (например, 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.

## 2.4. Базовые команды bash

В операционной системе GNU Linux взаимодействие пользователя с системой обычно осуществляется с помощью командной строки посредством построчного ввода команд. Общий формат команд можно представить следующим образом:

<имя\_команды> <разделитель> <аргументы>

Первые задачи, которые приходится решать в любой системе это — работа с данными (обычно хранящимися в файлах) и управление работающими в системе программами (процессами). Для получения достаточно подробной информации по каждой из команд используйте команду man, например: user@dk4n31:~\$ man ls.

В таблице 1.2 приведены основные команды взаимодействия пользователя с файловой системой в GNU Linux посредством командной строки.

Таблица 1.2 - Основные команды взаимодействия пользователя с файловой системой

Команда	Расшифровка	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

## 2.5. Полезные комбинации клавиш

Для удобства и экономии времени при работе в терминале существует большое количество сокращённых клавиатурных команд.

Клавиши  $\uparrow$  и  $\downarrow$  позволяют увидеть историю предыдущих команд в bash. Количество хранимых строк определено в переменной окружения HISTSIZE.

Клавиши  $\leftarrow$  и  $\rightarrow$  перемещают курсор влево и вправо в текущей строке, позволяя редактировать команды.

Сочетания клавиш **Ctrl + A** и **Ctrl + E** перемещают курсор в начало и в конец текущей строки. Клавиши **Ctrl + K** удаляет всё от текущей позиции курсора до конца строки, а **Ctrl + W** или **Alt + Backspace** удаляют слово перед курсором.

Сочетание клавиш **Ctrl + D** в пустой строке служит для завершения текущего сеанса. Для завершения выполняющейся в данный момент команды можно использовать **Ctrl + C**. Также данное сочетание отменит редактирование командной строки и вернёт приглашение командной строки. **Ctrl + I** очищает экран.

## 3. Порядок выполнения работы

### 3.1. Техническое обеспечение

Лабораторная работа подразумевает работу с операционной системой ОС Linux на уровне командной строки. Выполнение работы возможно как в дисплейном классе факультета физико-математических и естественных наук РУДН, так и дома. Описание выполнения работы приведено для дисплейного класса со следующими



характеристиками техники:

- Intel Core i3-550 3.2 GHz, 4 GB оперативной памяти, 8 GB свободного места на жёстком диске;
- ОС Linux Gentoo (<http://www.gentoo.ru/>).

### 3.2. Перемещение по файловой системе

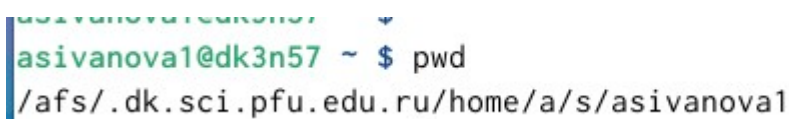
Откроем терминал. По умолчанию терминал открывается в домашнем каталоге пользователя, который обозначается символом ~ (Рисунок 2.2).

Убедимся, что Мы находимся в домашнем каталоге. Если это не так, то это можно сделать с помощью команды `cd` без аргументов.

```
user@dk4n31:/tmp$ cd .....
```

```
user@dk4n31:~$
```

С помощью команды `pwd` узнаем полный путь к Нашему домашнему каталогу (рисунок 3.2):

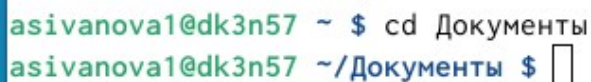


```
asivanova1@dk3n57 ~ $ pwd
/afs/.dk.sci.pfu.edu.ru/home/a/s/asivanova1
```

Рисунок 3.2 - Полный путь к домашнему каталогу

Команда `cd` позволяет сменить текущий каталог на другой, указав путь к нему в качестве параметра. Формат команды: `cd [путь_к_каталогу]`

Перейдем в подкаталог Документы Нашего домашнего каталога, указав относительный путь (рисунок 3.2):



```
asivanova1@dk3n57 ~ $ cd Документы
asivanova1@dk3n57 ~/Документы $
```

Рисунок 3.2 - Переход к папке Документы

Перейдем в каталог `local` – подкаталог `usr` корневого каталога, указав абсолютный путь к нему - `/usr/local` (рисунок 3.3):

```
asivanova1@dk3n57 ~ $ cd /usr/local
asivanova1@dk3n57 /usr/local $
```

Рисунок 3.3 – Переход в папку local по абсолютному пути

Обратите внимание абсолютный путь всегда начинается от корневого каталога (т.е. с символа /).

Можно использовать комбинацию 'cd -' для возвращения в последний посещённый пользователем каталог. А 'cd ..' используется для перехода на один каталог выше по иерархии.

Команда ls выдаёт список файлов указанного каталога и имеет следующий синтаксис: ls [опции] [каталог] [каталог...]

Для просмотра списка файлов текущего каталога может быть использована команда ls без аргументов.

Перейдя в домашний каталог, выведем список файлов Нашего домашнего каталога через ls (рисунок 3.5):

```
asivanova1@dk3n57 ~ $ ls
public      Видео      Загрузки   Музыка     'Рабочий стол'
public_html Документы  Изображения  Общедоступные  Шаблоны
```

Рисунок 3.5 – Вывод файлов домашнего каталога

Откроем домашний каталог с помощью файлового менеджера графического окружения (рисунок 3.6):

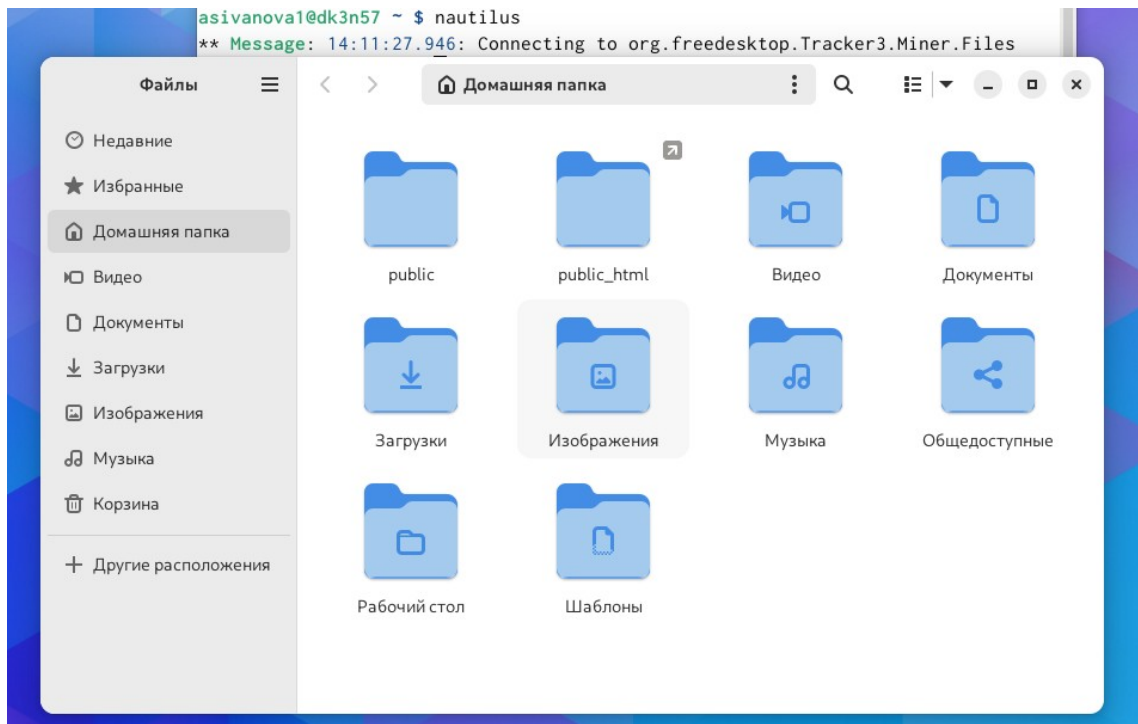


Рисунок 3.6 – Домашний каталог

Также как и команда `cd`, команда `ls` работает как с абсолютными, так и с относительными путями.

Выведем список файлов подкаталога `Документы` Нашего домашнего каталога указав относительный путь (рисунок 3.7):

```
asivanova1@dk3n57 ~ $ ls Документы
Документы
```

Рисунок 3.7 – Файлы каталога `Документы`

Выведем список файлов каталога `/usr/local`, указав абсолютный путь к нему (рисунок 3.8):

```
asivanova1@dk3n57 ~ $ ls /usr/local
bin  games  info  lib  lib32  lib64  man  sbin  share  src  texlive
```

Рисунок 3.8 – Файлы каталога `local` через абсолютный путь

Для данной команды существует довольно много опций (ключей), ниже дано

описание некоторых из них.

Таблица 3.1 - Опции команды ls

Ключ	Описание
-a	вывод списка всех файлов, включая скрытые файлы (в Linux названия скрытых файлов начинаются с точки)
-R	рекурсивный вывод списка файлов и подкаталогов
-h	вывод для каждого файла его размера
-l	вывод дополнительной информации о файлах (права доступа, владельцы и группы, размеры файлов и время последнего доступа)
Ключ	Описание
-i	вывод уникального номера файла (inode) в файловой системе перед каждым файлом
-d	обработка каталогов, указанных в командной строке, так, как если бы они были обычными файлами, вместо вывода списка их файлов

Примеры:

- команда `ls -R` рекурсивно выводит список содержимого текущего каталога;
- команда `ls -is images/ ..` выводит список файлов каталога `images` и родительского по отношению к текущему каталога, при этом для каждого файла указан номер `inode` и его размер в килобайтах;
- команда `ls -l images/*.png` выводит список всех файлов в каталоге `images`, чьи имена заканчиваются на `.png`, включая скрытый файл `.png`, если таковой существует.

Используем команду `ls` с ключем `-R` (рисунок 3.9):

```

asivanova1@dk3n57 ~ $ ls -R /usr/local
/usr/local:
bin  games  info  lib  lib32  lib64  man  sbin  share  src  texlive

/usr/local/bin:
cpntools  gpss  sali_prepare

/usr/local/games:

/usr/local/info:

/usr/local/lib:

/usr/local/lib64:

/usr/local/sbin:

/usr/local/share:
ca-certificates  config  doc  man  portage

/usr/local/share/ca-certificates:
russian_trusted_root_ca.pem.crt  russian_trusted_sub_ca.pem.crt
ls: невозможно открыть каталог '/usr/local/share/config': Отказано в доступе

```

Рисунок 3.9 – Использование команды ls с ключом -R

### 3.3 Создание пустых каталогов и файлов

Для создания каталогов используется команда `mkdir`. Её синтаксис имеет вид:  
`mkdir [опции] <каталог> [каталог...]`

Создадим в домашнем каталоге подкаталог с именем `parentdir` и с помощью команды `ls` проверим, что каталог создан (рисунок 3.10):

```

asivanova1@dk3n57 ~ $ cd
asivanova1@dk3n57 ~ $ mkdir parentdir
asivanova1@dk3n57 ~ $ ls
parentdir  public_html  Документы  Изображения  Общедоступные  Шаблоны
public     Видео        Загрузки   Музыка        'Рабочий стол'

```

Рисунок 3.10 – Создание каталога `parentdir`

Создадим подкаталог в существующем каталоге `parentdir/dirls` (рисунок 3.11):

```
asivanova1@dk3n57 ~ $ mkdir parentdir/dir
```

Рисунок 3.11 – Создание подкаталога через каталог

Зададим нескольких аргументов и создадим несколько каталогов (рисунок 3.12):

```
asivanova1@dk3n57 ~ $ cd parentdir
asivanova1@dk3n57 ~/parentdir $ mkdir dir1 dir2 dir3
```

Рисунок 3.12 – Создание нескольких каталогов

Создадим подкаталог в каталоге, отличном от текущего и проверим это с помощью команды `ls ~` (рисунок 3.13):

```
asivanova1@dk3n57 ~/parentdir $ mkdir ~/newdir
asivanova1@dk3n57 ~/parentdir $ ls ~
newdir      public_html  Загрузки    Общедоступные
parentdir   Видео       Изображения 'Рабочий стол'
public      Документы   Музыка      Шаблоны
```

Рисунок 3.13 – Создание каталога в домашнем каталоге

Опция – `parents` (краткая форма `-p`) позволяет создавать иерархическую цепочку подкаталогов, создавая все промежуточные каталоги. Создадим следующую последовательность вложенных каталогов `newdir/dir1/dir2` в домашнем каталоге (рисунок 3.14):

```
asivanova1@dk3n57 ~ $ mkdir -p ~/newdir/dir1/dir2
```

Рисунок 3.14 – Создание цепочки каталогов

Для создания файлов может быть использована команда `touch`, которая имеет следующий синтаксис: `touch [опции] файл [файл...]`

Создадим файл `test.txt` в каталоге `~/newdir/dir1/dir2` и проверим наличие файла

с помощью команды `ls ~/newdir/dir1/dir2` (рисунок 3.15):

```
asivanova1@dk3n57 ~ $ touch ~/newdir/dir1/dir2/test.txt
asivanova1@dk3n57 ~ $ ls ~/newdir/dir1/dir2
test.txt
```

Рисунок 3.15 – Создание файла через цепочку каталогов

### 3.4 Перемещение и удаление файлов или каталогов

Команда `rm` удаляет файлы и (или) каталоги и имеет следующий синтаксис: `rm [опции] <файл|каталог> [файл|каталог...]`

Опции команды `rm`:

- `r` или `-R`: рекурсивное удаление (это обязательная опция для удаления любого каталога, пустого или содержащего файлы и (или) подкаталоги);
- `i`: запрос подтверждения перед удалением;
- `v`: вывод подробной информации при выполнении команды;
- `f`: принудительное удаление файлов или каталогов.

Для удаления пустых каталогов можно воспользоваться командой `rmdir`.

Запросив подтверждение на удаление каждого файла в текущем каталоге, удалим в подкаталоге `/newdir/dir1/dir2/` все файлы с именами, заканчивающимися на `.txt` (рисунок 3.16):

```
asivanova1@dk3n57 ~ $ rm -i ~/newdir/dir1/dir2/*.txt
rm: удалить пустой обычный файл '/afs/.dk.sci.pfu.edu.ru/home/a/s/asivanova1/new
dir/dir1/dir2/test.txt'? yes
asivanova1@dk3n57 ~ $
```

Рисунок 3.16 – Удаление всех файлов с расширением `.txt`

Рекурсивно удалим из текущего каталога без запроса подтверждения на удаление каталог `newdir`, а также файлы, чьи имена начинаются с `dir` в каталоге `parentdir` (рисунок 3.17):

```
asivanova1@dk3n57 ~ $ rm -R ~/newdir ~/parentdir/dir*
```

Рисунок 3.17 – Удаление пустых каталогов

Команда `rm` удаляет файлы безвозвратно, и не существует способа для их восстановления. Команда `mv` служит для перемещения файлов и каталогов и имеет следующий синтаксис: `mv [опции] <файл|каталог> [файл|каталог...] <назначение>`

Некоторые опции:

- `f`: принудительное выполнение операции (предупреждение не будет выводиться даже при перезаписи существующего файла);
- `i`: запрашивается подтверждение перед перезаписью существующего файла;
- `v`: подробный режим, который сообщает обо всех изменениях и действиях при выполнении команды.

Команда `cp` копирует файлы и каталоги и имеет следующий синтаксис: `cp`

```
asivanova1@dk3n57 ~ $ cp parentdir3/test2.txt parentdir3/subtest2.txt
asivanova1@dk3n57 ~ $ mv -i parentdir3/test1.txt parentdir3/newtest.txt
asivanova1@dk3n57 ~ $ ls parentdir3
```

`[опции] <файл|каталог> [файл|каталог ...] <назначение>`

Некоторые опции команды `cp`:

- `R`: рекурсивное копирование; является обязательной опцией для копирования каталогов;
- `i`: запрос подтверждения перед перезаписью любых файлов;
- `f`: заменяет любые существующие файлы без запроса подтверждения;
- `v`: подробный режим, сообщает обо всех изменениях и действиях.

Для демонстрации работы команд `cp` и `mv` приведём следующие примеры. Создадим следующие файлы и каталоги в домашнем каталоге (рисунок 3.18):

```
asivanova1@dk3n57 ~ $ cd
asivanova1@dk3n57 ~ $ mkdir -p parentdir1/dir1 parentdir2/dir2 parentdir3
asivanova1@dk3n57 ~ $ touch parentdir1/dir1/test1.txt parentdir2/dir2/test2.txt
```

Рисунок 3.18 – Создание каталогов и файлов в этих каталогах



Используя команды `cp` и `mv`, файл `test1.txt` скопируем, а `test2.txt` переместим в каталог `parentdir3` и с помощью команды `ls` проверим корректность выполненных команд (рисунок 3.19):

```
asivanova1@dk3n57 ~ $ mv parentdir1/dir1/test1.txt parentdir3
asivanova1@dk3n57 ~ $ cp parentdir2/dir2/test2.txt parentdir3
asivanova1@dk3n57 ~ $ ls parentdir3
test1.txt  test2.txt
asivanova1@dk3n57 ~ $ ls parentdir1/dir1
asivanova1@dk3n57 ~ $ ls parentdir2/dir2
test2.txt
```

Рисунок 3.19 – Перемещение и копирование файлов

Также команда `mv` может быть использована для переименования файлов и каталогов, а команда `cp` позволяет сделать копию файла с новым именем. Переименуем файл `test1.txt` из каталога `parentdir3` в `newtest.txt` (рисунок 3.20):

Рисунок 3.20 – Переименование файла

Переименуем каталог `dir1` в каталоге `parentdir1` в `newdir` (рисунок 3.21):

```
asivanova1@dk3n57 ~ $ cd parentdir1
asivanova1@dk3n57 ~/parentdir1 $ ls
dir1
asivanova1@dk3n57 ~/parentdir1 $ mv dir1 newdir
asivanova1@dk3n57 ~/parentdir1 $ ls
newdir
```

Рисунок 3.21 – Переименование каталога

### 3.5 Команда `cat`: вывод содержимого файлов

Команда `cat` объединяет файлы и выводит их на стандартный вывод - обычно это экран (рисунок 3.22):

```

asivanova1@dk3n57 ~ $ cat /etc/hosts
# /etc/hosts: Local Host Database
#
# This file describes a number of aliases-to-address mappings for the for
# local hosts that share this file.
#
# In the presence of the domain name service or NIS, this file may not be
# consulted at all; see /etc/host.conf for the resolution order.
#
# IPv4 and IPv6 localhost aliases
127.0.0.1      dk3n21 localhost.localdomain localhost
::1           localhost
#
# Imaginary network.
#10.0.0.2      myname
#10.0.0.3      myfriend
#
# According to RFC 1918, you can use the following IP networks for private
# nets which will never be connected to the Internet:
#
#      10.0.0.0      -   10.255.255.255
#      172.16.0.0    -   172.31.255.255
#      192.168.0.0   -   192.168.255.255
#
# In case you want to be able to connect directly to the Internet (i.e. not
# behind a NAT, ADSL router, etc...), you need real official assigned
# numbers. Do not try to invent your own network numbers but instead get one
# from your network provider (if any) or from your regional registry (ARIN,
# APNIC, LACNIC, RIPE NCC, or AfriNIC.)

```

Рисунок 3.22 – Выполнение команды cat

#### 4. Задание для самостоятельной работы

1) Воспользовавшись командой `pwd`, узнаем полный путь к своей домашней директории.

```

asivanova1@dk3n57 ~ $ pwd
/afs/.dk.sci.pfu.edu.ru/home/a/s/asivanova1

```

Рисунок 4.1 – Полный путь к домашнему каталогу

2) Введем следующую последовательность команд (рисунок 4.2):

3) 1) `cd`

- 2) mkdir tmp
- 3) cd tmp
- 4) pwd
- 5) cd /tmp
- 6) pwd

```

asivanova1@dk3n57 ~ $ cd
asivanova1@dk3n57 ~ $ mkdir tmp
asivanova1@dk3n57 ~ $ cd tmp
asivanova1@dk3n57 ~/tmp $ pwd
/afs/.dk.sci.pfu.edu.ru/home/a/s/asivanova1/tmp
asivanova1@dk3n57 ~/tmp $ cd /tmp
asivanova1@dk3n57 /tmp $ pwd
/tmp

```

Рисунок 4.2 – Результат выполнения последовательности команд

Объясним, почему такой результат: при добавлении / в команду cd при переходе на 1 каталог используется как абсолютный переход к каталогу. После выполнения команды cd / путь терминала начинается с папки назначения.

4) Пользуясь командами cd и ls, посмотрим содержимое корневого каталога, домашнего каталога, каталогов /etc и /usr/local. Для начала проверим корневой каталог, он проверяется командой ls / (рисунок 4.3):

```

asivanova1@dk3n57 ~ $ ls /
afs  boot  dev  home  lib64      media  net  proc  run  srv  tmp  var
bin  com   etc  lib   lost+found mnt    opt  root  sbin sys  usr

```

Рисунок 4.3 – Файлы корневого каталога

Затем проверим домашний каталог командой cd и ls (рисунок 4.4):

```

asivanova1@dk3n57 ~ $ cd
asivanova1@dk3n57 ~ $ ls
'~p'      parentdir2  public_html  Документы  Музыка      Шаблоны
parentdir  parentdir3  tmp          Загрузки   Общедоступные
parentdir1 public       Видео        Изображения 'Рабочий стол'

```

Рисунок 4.4 – Файлы домашнего каталога

Затем проверим введенные в задании каталоги (рисунки 4.5-4.6):

```
asivanova1@dk3n57 ~ $ cd /etc
asivanova1@dk3n57 /etc $ ls
a2ps                hotplug             pkcs11
acpi                hotplug.d           pki
adjtime             hsqlldb             pmount.allow
afs.keytab          i3blocks.conf       pmount.conf
alsa               i3status.conf       polkit-1
apparmor.d         idmapd.conf         portage
apt               idn2.conf           postgresql-10
ati              idn2.conf.sample   postgresql-11
audit            idnalias.conf       postgresql-12
autofs          idnalias.conf.sample postgresql-13
avahi           ImageMagick-7       postgresql-17
```

Рисунок 4.5 – Файлы каталога etc

```
asivanova1@dk3n57 ~ $ cd /usr/local
asivanova1@dk3n57 /usr/local $ ls
bin  games  info  lib  lib32  lib64  man  sbin  share  src  texlive
```

Рисунок 4.6 – Файлы каталога usr/local

5) Пользуясь изученными консольными командами, в своём домашнем каталоге создадим каталог temp и каталог labs с подкаталогами lab1, lab2 и lab3 одной командой. В каталоге temp создадим файлы text1.txt, text2.txt, text3.txt. Пользуясь командой ls, убедимся, что все действия выполнены успешно (каталоги и файлы созданы). Создадим цепочку каталогов командой mkdir с параметром -p (рисунки 4.7):

```
asivanova1@dk3n57 ~ $ mkdir temp labs labs/lab1 labs/lab2 labs/lab3
asivanova1@dk3n57 ~ $ touch ~/temp/text1.txt
asivanova1@dk3n57 ~ $ touch ~/temp/text2.txt
asivanova1@dk3n57 ~ $ touch ~/temp/text3.txt
asivanova1@dk3n57 ~ $ cd labs
asivanova1@dk3n57 ~/labs $ ls
lab1  lab2  lab3
```

Рисунок 4.7 – Создание каталогов и файлов

С помощью текстового редактора (например, редактора mcedit) запишем в файл text1.txt свое имя, в файл text2.txt фамилию, в файл text3.txt учебную группу и выведем на экран содержимое файлов, используя команду cat (рисунок 4.8):

```
asivanova1@dk3n57 ~ $ cd temp
asivanova1@dk3n57 ~/temp $ ls
text1.txt  text2.txt  text3.txt

asivanova1@dk3n57 ~ $ cd temp
asivanova1@dk3n57 ~/temp $ mcedit text1.txt

asivanova1@dk3n57 ~/temp $ mcedit text2.txt

asivanova1@dk3n57 ~/temp $ mcedit text3.txt

asivanova1@dk2n23 ~ $ cat temp/text1.txt
Анастасия
asivanova1@dk2n23 ~ $ cat temp/text2.txt
Иванова
asivanova1@dk2n23 ~ $ cat temp/text3.txt
НКАбд-07-25
```

Рисунок 4.8 – Вывод содержимого текстовых файлов

6) Скопируем все файлы, чьи имена заканчиваются на .txt, из каталога ~/temp в каталог labs. После этого переименуем файлы каталога labs и переместим их: text1.txt переименуем в firstname.txt и переместим в подкаталог lab1, text2.txt в lastname.txt в подкаталог lab2, text3.txt в id-group.txt в подкаталог lab3. Пользуясь командами ls и cat, убедимся, что все действия выполнены верно (рисунок 4.9):

```
asivanova1@dk2n23 ~ $ cp temp/text1.txt temp/text2.txt temp/text3.txt labs
asivanova1@dk2n23 ~ $ cd labs
asivanova1@dk2n23 ~/labs $ ls
lab1  lab2  lab3  text1.txt  text2.txt  text3.txt
asivanova1@dk2n23 ~ $ mv -i labs/text1.txt labs/firstname.txt
asivanova1@dk2n23 ~ $ mv labs/firstname.txt labs/lab1
asivanova1@dk2n23 ~ $ mv -i labs/text2.txt labs/lastname.txt
asivanova1@dk2n23 ~ $ mv labs/lastname.txt labs/lab2
asivanova1@dk2n23 ~ $ mv -i labs/text3.txt labs/id-group.txt
asivanova1@dk2n23 ~ $ mv labs/id-group.txt labs/lab3
```

```

asivanova1@dk2n23 ~ $ cat labs/lab1/firstname.txt
Анастасия
asivanova1@dk2n23 ~ $ cat labs/lab2/lastname.txt
И
asivanova1@dk2n23 ~ $ cd labs
asivanova1@dk2n23 ~/labs $ ls
lab1  lab2  lab3
asivanova1@dk2n23 ~/labs $ cd lab1
asivanova1@dk2n23 ~/labs/lab1 $ ls
firstname.txt
asivanova1@dk2n23 ~/labs/lab1 $ cd
asivanova1@dk2n23 ~ $ cd labs
asivanova1@dk2n23 ~/labs $ cd lab2
asivanova1@dk2n23 ~/labs/lab2 $ ls
lastname.txt
asivanova1@dk2n23 ~/labs/lab2 $ cd
asivanova1@dk2n23 ~ $ cd labs
asivanova1@dk2n23 ~/labs $ cd lab3
asivanova1@dk2n23 ~/labs/lab3 $ ls
id-group.txt

```

Рисунок 4.9 – Переименование и перемещение текстовых файлов

7) Удалим все созданные в ходе выполнения лабораторной работы файлы и каталоги. В ходе работы все папки создавались в домашнем каталоге, проверим все папки командой `ls`, и созданные удалим с помощью `rm -R` (рисунок 4.10):

```

asivanova1@dk2n23 ~ $ rm -R parentdir parentdir1 parentdir2 parentdir3 labs temp tmp
asivanova1@dk2n23 ~ $ ls
public      Видео      Загрузки   Музыка      'Рабочий стол'
public_html Документы  Изображения Общедоступные Шаблоны

```

Рисунок 4.10 – Удаление всех созданных ранее каталогов

## 5. Заключение

Мы приобрели практические навыки работы с операционной системой на уровне командной строки (организация файловой системы, навигация по файловой

системе, создание и удаление файлов и директорий).