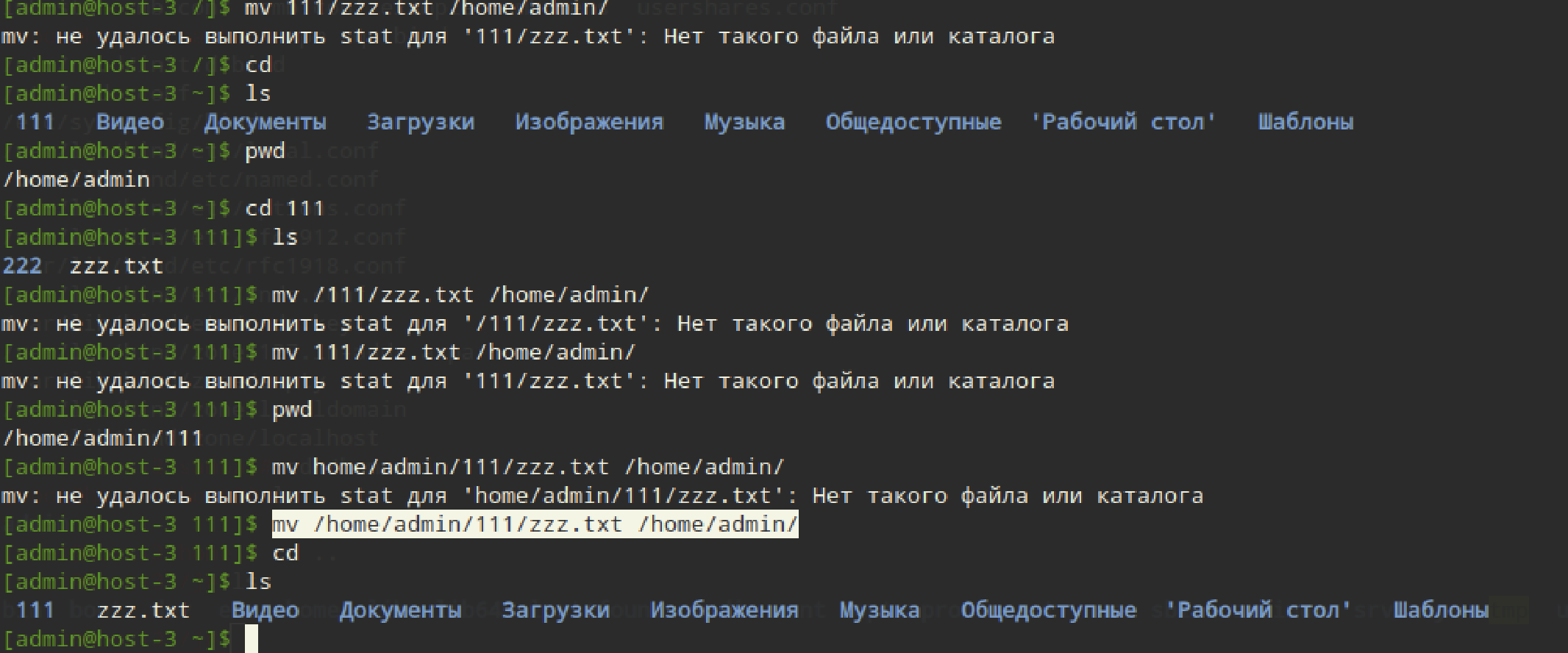
**1 – Work with files**

Task 1

**Работа в консольке**

1. Переместиться между директориями – **cd (cd ..)**
2. Вывести список файлов в директории – **ls (ls -l)**
3. Вывести список Всех файлов в директории – **ls -a**
4. Создать папку с подпапками - **mkdir -p 111/222/333**
5. Внутри папки создать файлик и записать в него что-нибудь – **echo “choto” > 111/zzz.txt**
6. Переместить файл из одной директории в другую – **mv home/admin/111/zzz.txt /home/admin/**
7. скопировать файл из одной директории в другую – **cp** **111/zzz.txt /home/admin/**
8. переименовать файл – **mv 111/zzz.txt 111/new\_zzz.txt**
9. сравнить содержимое файла – **diff 111/zzz.txt /home/user/Documents/text.txt**
10. отсортировать содержимое файла по возрастанию и убыванию – **sort 111/zzz.txt > 111/sortzzz.txt (sort -r 111/zzz.txt > 111/sortzzz.txt - поубыванию)**
11. удалить все папки и файлы. – **rm -rf 111**

Task 2

**Перенаправляем**

1. Как работают команды >,>>?

- **>** — перенаправляет вывод команды в файл, заменяя его содержимое

Пример: echo "что-то" > text.txt запишет "что-то" в файл text.txt, заменив предыдущее содержимое

- **>>** — перенаправляет вывод команды в файл, добавляя его в конец файла, не удаляя существующее содержимое

Пример: echo "Еще что-то" >> text.txt добавит "Еще что-то" в конец text.txt.

1. Что такое перенаправление ввода? stderr,stdout;

- **stdout** (стандартный вывод) — поток, куда программа отправляет результаты своей работы (по умолчанию выводится на экран)

- **stderr** (стандартный ввод ошибок) — поток, куда программа отправляет сообщения об ошибках (также выводится на экран)

1. Вывести содержание файла не используя текстовые редакторы;

- **cat file.txt**

1. Создать файл с содержимым не используя текстовые редактор;

- **echo "что-то" > file.txt**

1. перенаправить stdout в stderr и обратно на примере команды kinit, ping, tracert

**kinit user 2> errors.txt # Перенаправляем stderr в файл**

**ping example.com > output.txt 2>&1 # Перенаправляем stdout и stderr в один файл (output.txt)**

1. чем отличаются stdout и stderr

- **stdout** используется для нормального вывода программы (результаты выполнения)

- **stderr** используется для вывода сообщений об ошибках и предупреждениях. Позволяет разделять нормальный вывод и сообщения об ошибках

1. что такое stdin?

**stdin** (стандартный ввод) — это поток ввода, с которого программа принимает данные

1. как отправить весь вывод команды в пустоту?

- **команда > /dev/null**

1. можно ли отправить одновременно stdin и stdout в пустоту?

Не-a, **stdin** (ввод) и **stdout** (вывод) нельзя отправить одновременно в пустоту, поскольку **stdin** используется для получения данных, а **stdout** — для вывода. Можно игнорировать **stdout** с помощью **> /dev/null**, а **stdin** останется активным для других команд или данных ввода.