## Факультет информационных технологий и робототехники

Кафедра программного обеспечения информационных систем и технологий

**Отчет по лабораторной работе № 2**

по дисциплине: ”Системное программирование”

на тему: *”Основы работы с терминалом и командной строкй UNIX”*

Вариант -

Выполнили:студенты группы 10701222 Медведский Е.В Зухта К.М.

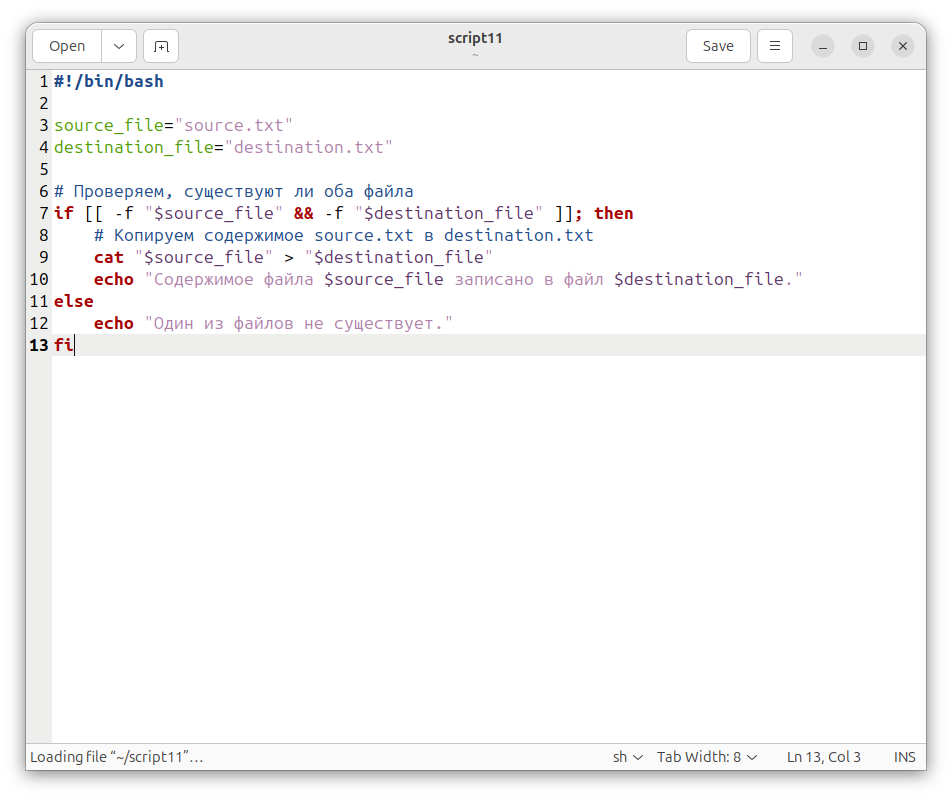
Принял**:** пр. Давыденко Н.В.

Минск 2024

**Цель работы:**Закрепить на практике основы работы со сценариями командной оболочки и перенаправление ввода/вывода.

**Задание1**

1. Написать скрипт, который записывает содержимое одного файла в другой и при этом оба файла существуют до операции. Дать два разных варианта выполнения.



Код программы

#!/bin/bash

source\_file="source.txt"

destination\_file="destination.txt"

# Проверяем, существуют ли оба файла

if [[ -f "$source\_file" && -f "$destination\_file" ]]; then

# Копируем содержимое source.txt в destination.txt

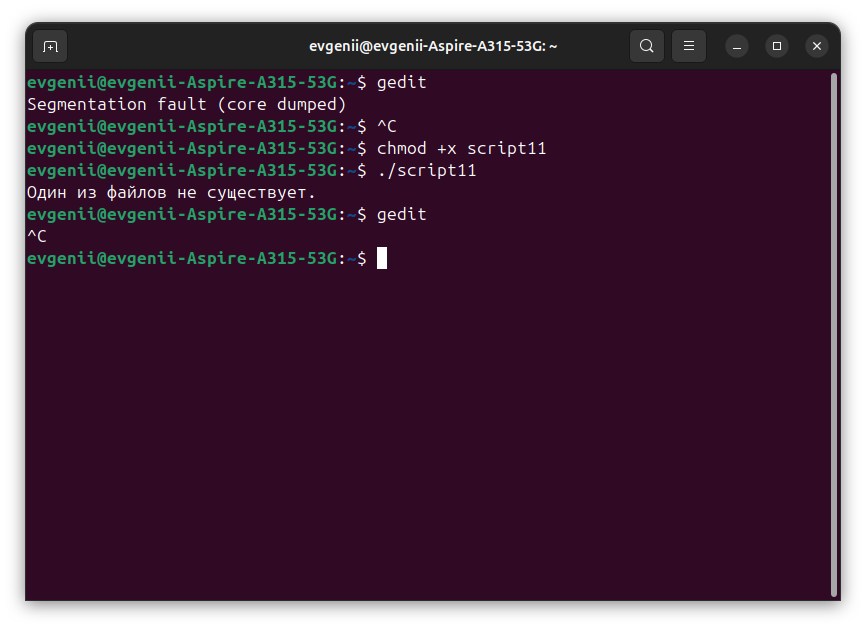
cat "$source\_file" > "$destination\_file"

echo "Содержимое файла $source\_file записано в файл $destination\_file."

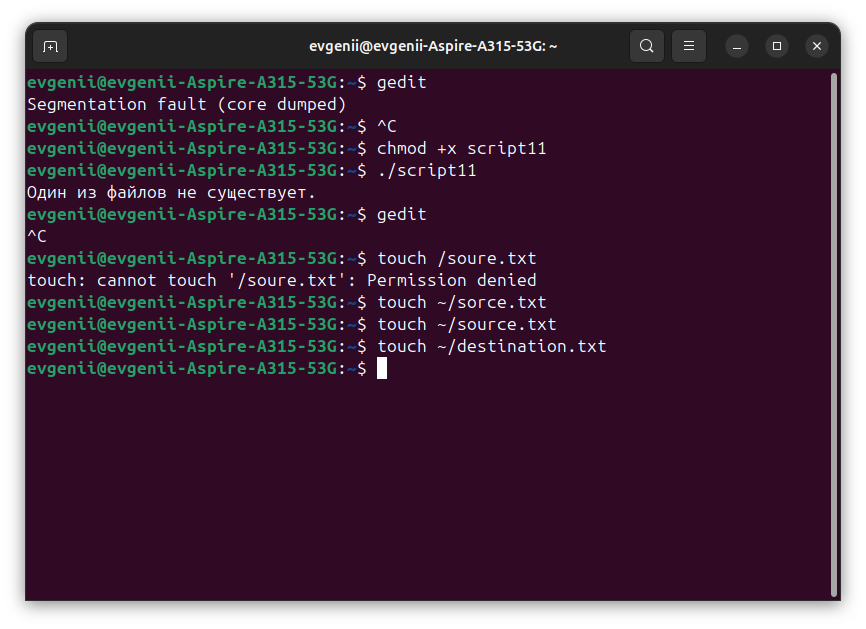
else

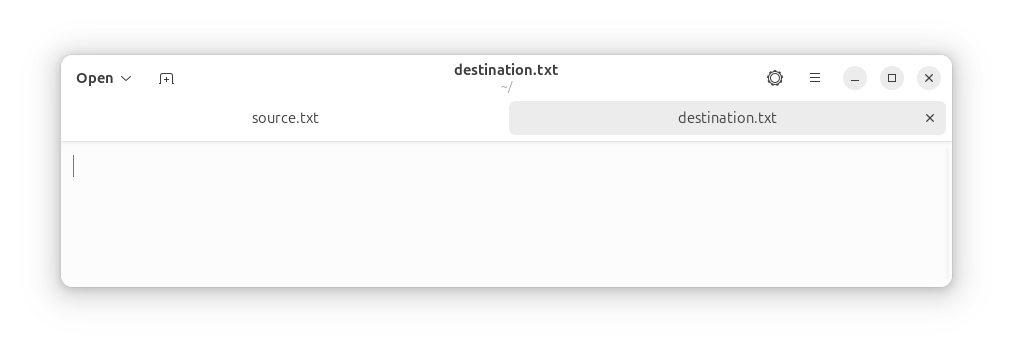
echo "Один из файлов не существует."

fi

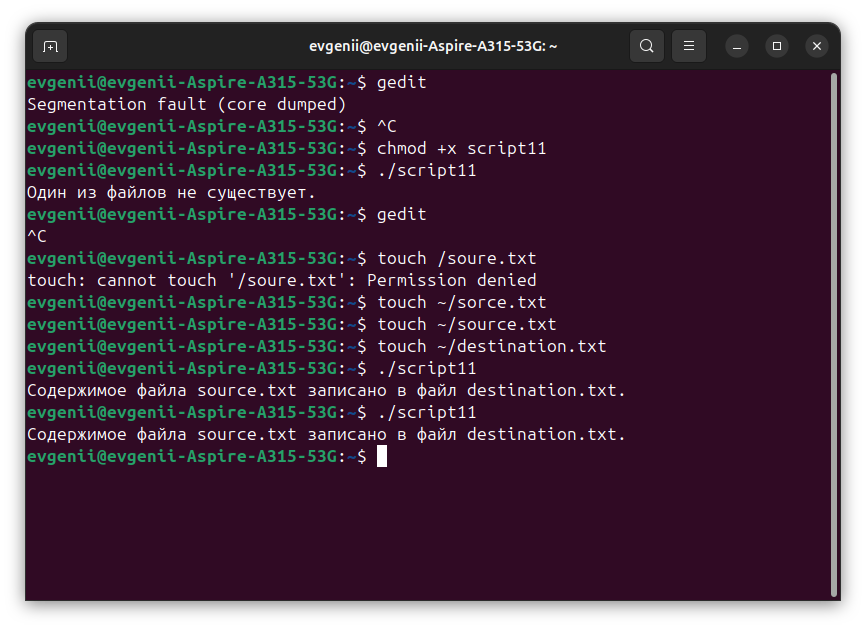


Как видим, т.к. таких файлов не существует - выводит соответствующее сообщение. Создадим 2 файла source.txt и destination.txt

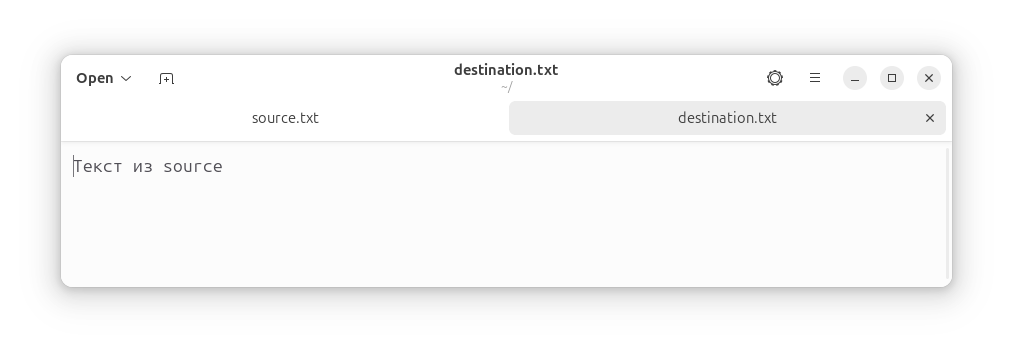




Содержимое файлов source.txt и estination.txtперед исполнением команды

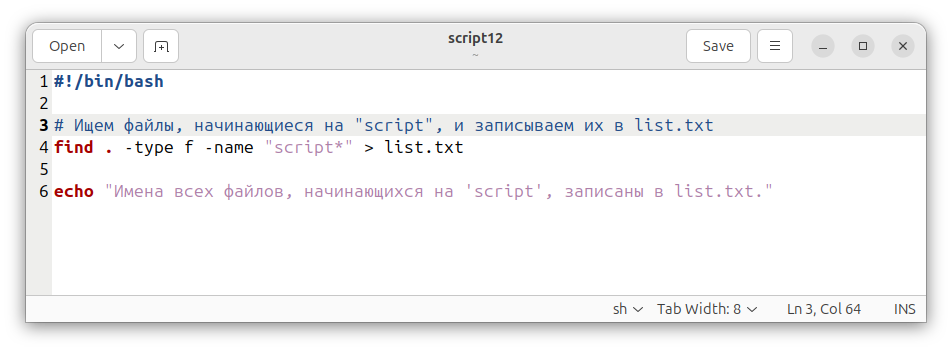


Выполнение скрипта



Видим результат

1. Найти все файлы, начинающиеся на слово script и записать их имена в файл list.txt.



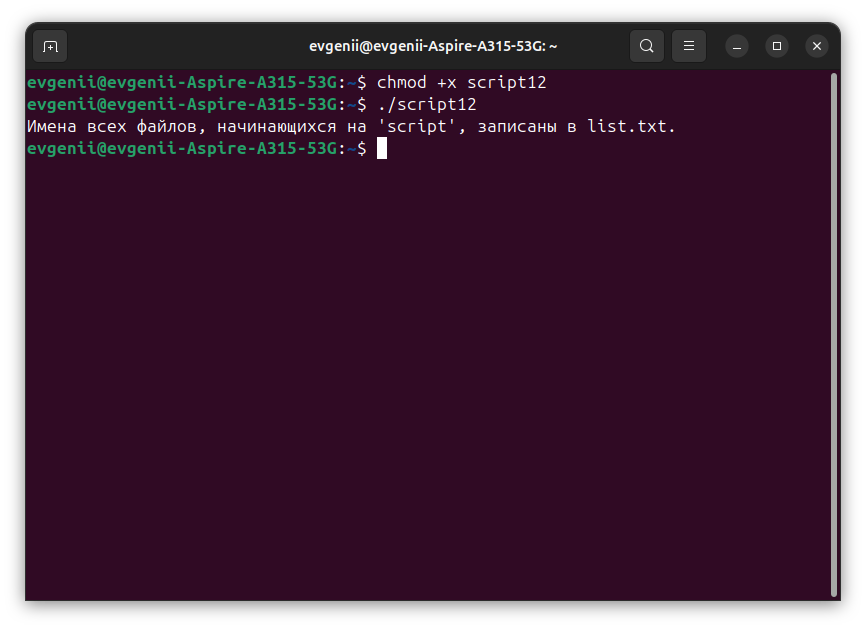
Код программы

#!/bin/bash

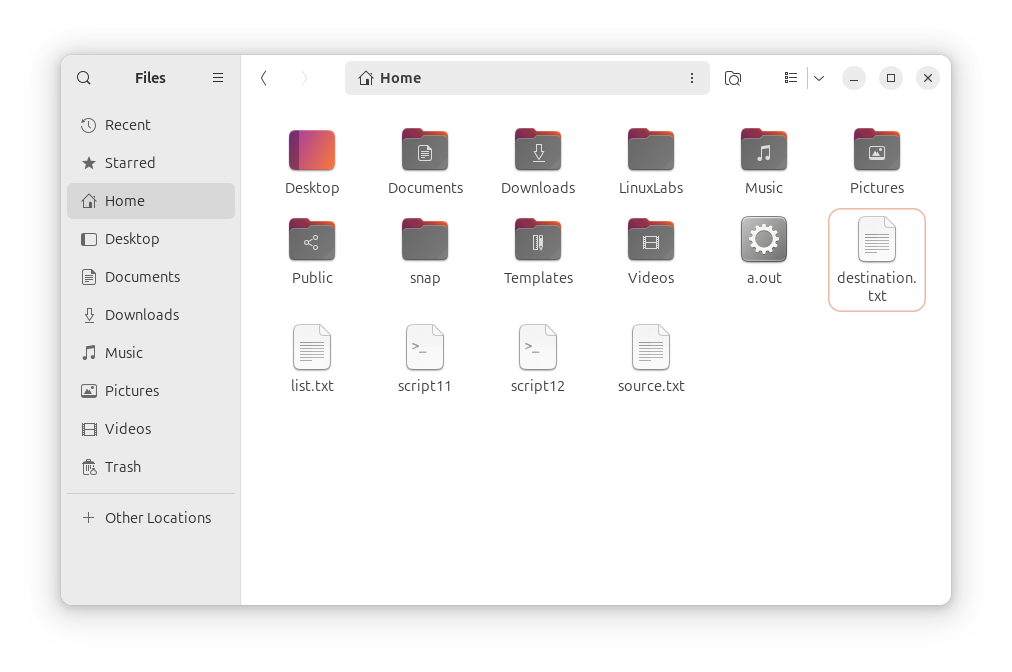
# Ищем файлы, начинающиеся на "script", и записываем их в list.txt

find . -type f -name "script\*" > list.txt

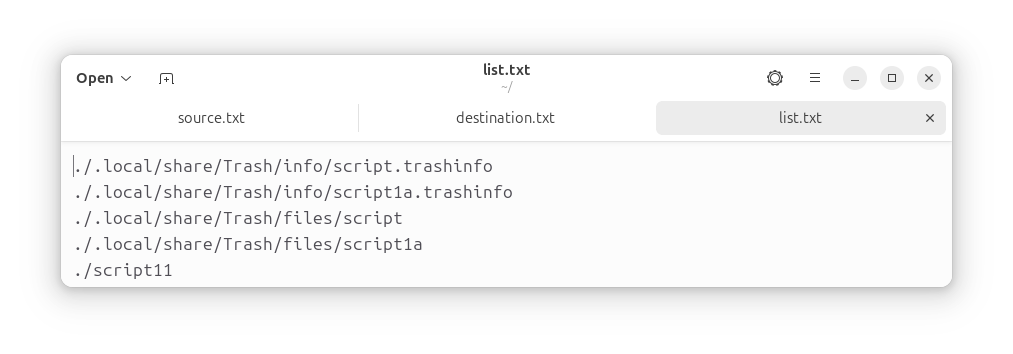
echo "Имена всех файлов, начинающихся на 'script', записаны в list.txt."

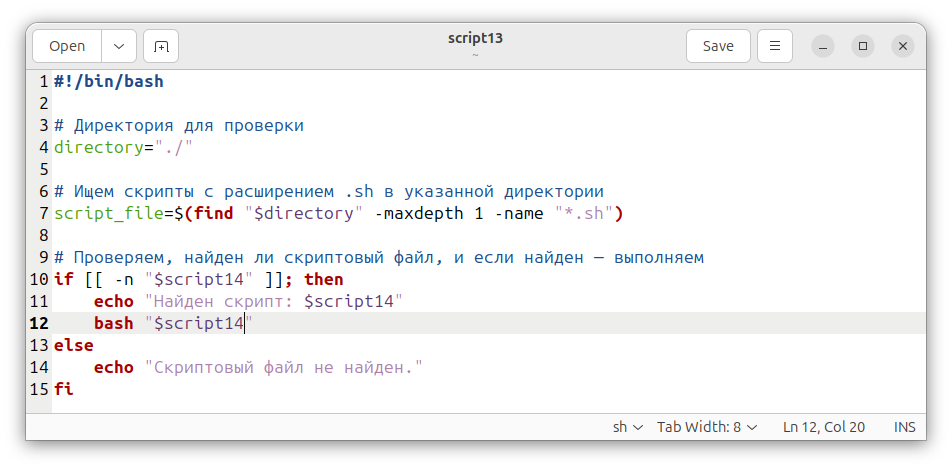


Делаем файл script12 исполняемым и запускаем его



Видим, что в проводнике появился файл list.txt. Откроем его.

3. Написать скрипт, который проверяет, содержится ли скриптовый файл в директории, если да, то выполнить его.



Код программы

#!/bin/bash

# Директория для проверки

directory="./"

# Ищем скрипты с расширением .sh в указанной директории

script\_file=$(find "$directory" -maxdepth 1 -name "\*.sh")

# Проверяем, найден ли скриптовый файл, и если найден — выполняем

if [[ -n "$script14" ]]; then

echo "Найден скрипт: $script14"

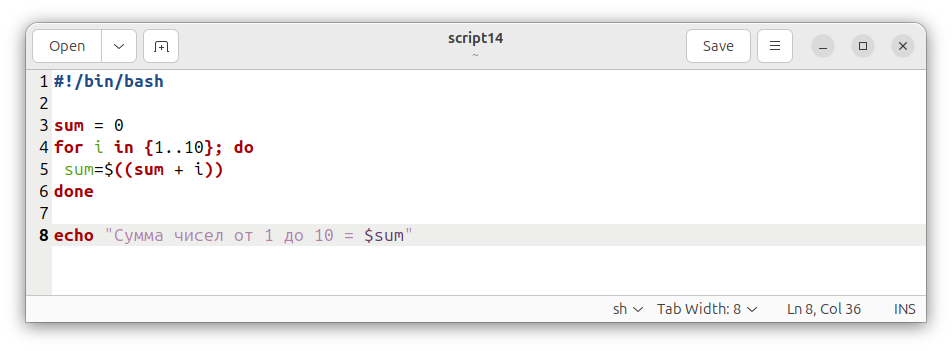
bash "$script14"

else

echo "Скриптовый файл не найден."

fi

4. Написать скрипт, подсчитывающий сумму от 1 до 10.



Код программы

#!/bin/bash

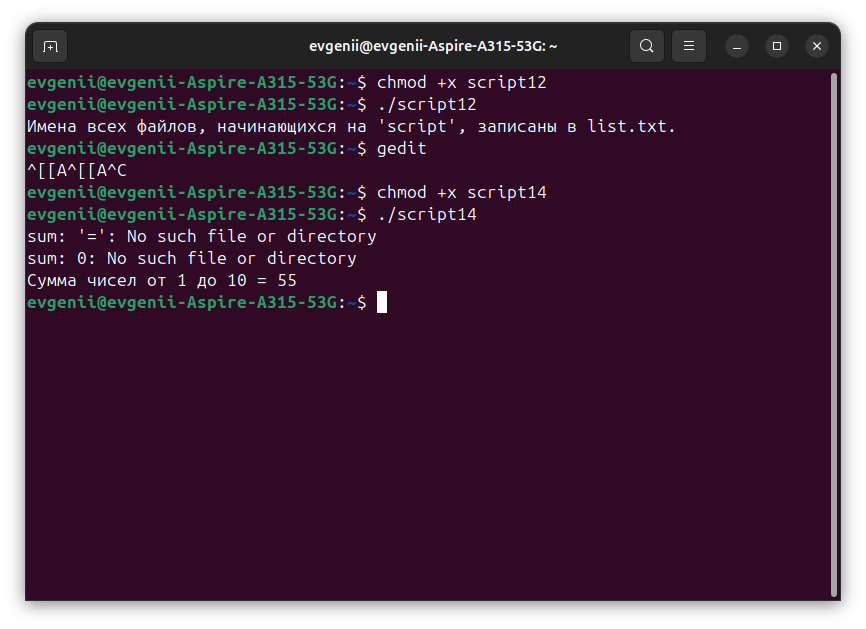
sum = 0

for i in {1..10}; do

sum=$((sum + i))

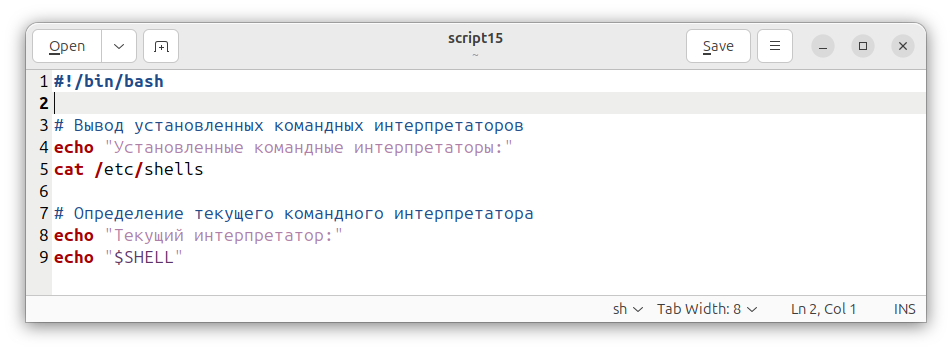
done

echo "Сумма чисел от 1 до 10 = $sum"



Получаем результат

1. А какие командные интерпретаторы установлены в вашей системе? Напишите скрипт поиска местонахождения командных интерпретаторов вашей системы. Также определите какой интерпретатор используется сейчас в вашей системе?



#!/bin/bash

# Вывод установленных командных интерпретаторов

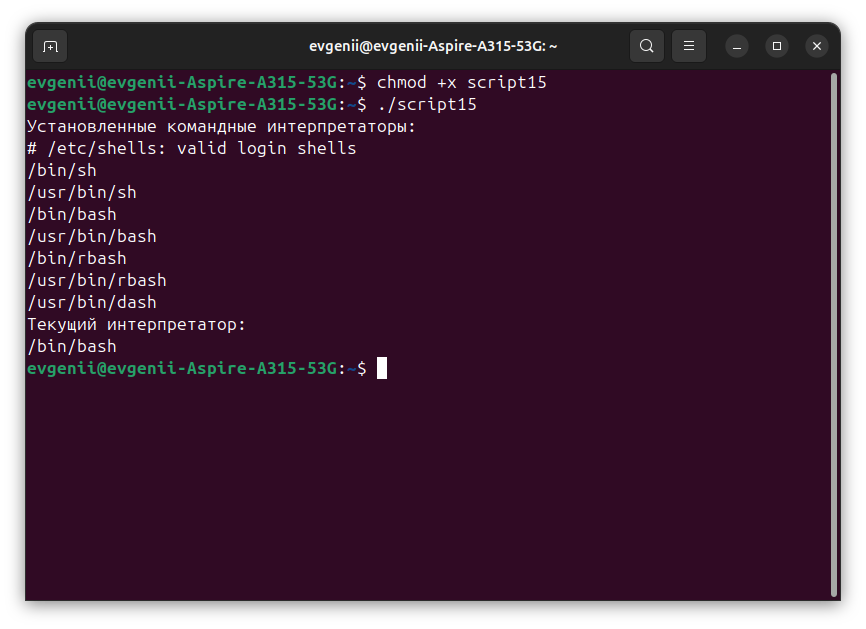
echo "Установленные командные интерпретаторы:"

cat /etc/shells

# Определение текущего командного интерпретатора

echo "Текущий интерпретатор:"

echo "$SHELL"



**Задание2**

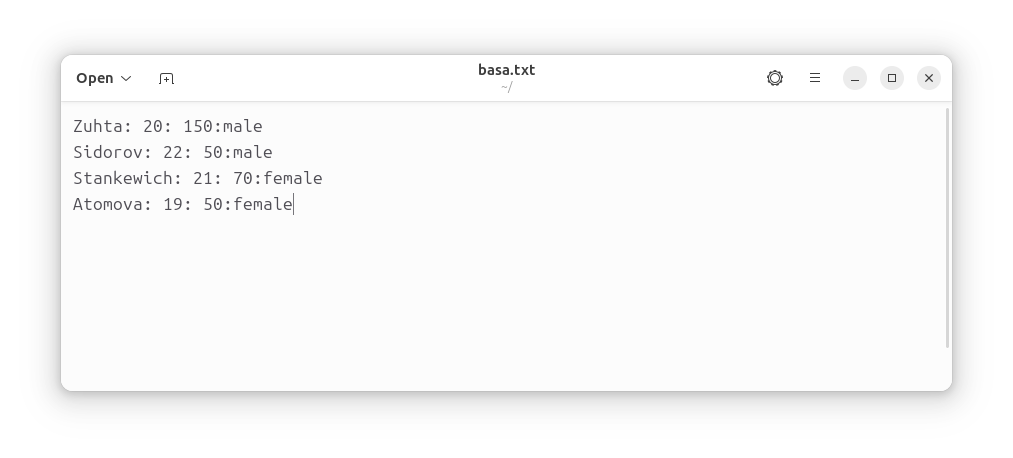
1. Используйте текст первого примера, заменить некоторые фамилии на фамилии членов вашей команды (например, Nutrina заменить на Stankewich).

Zuhta: 20: 150:male

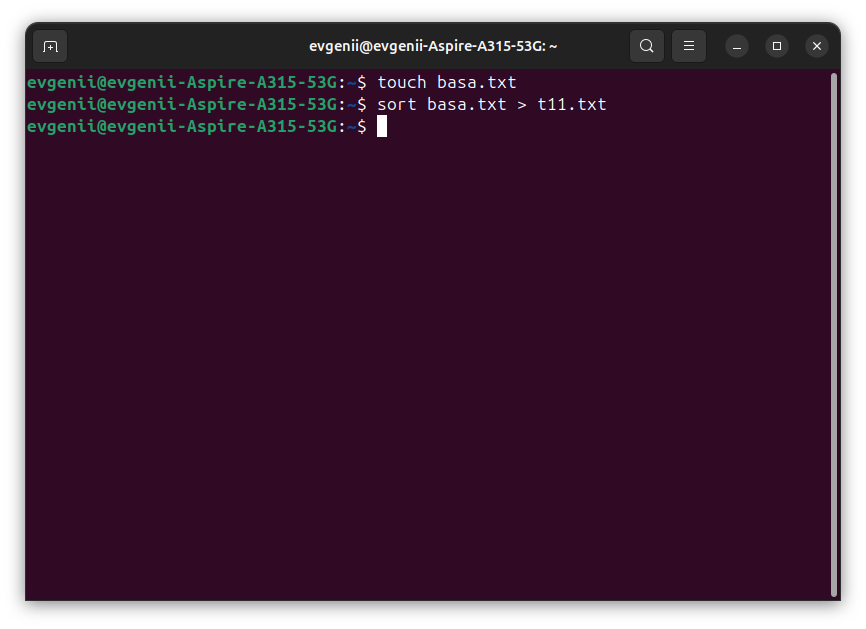
Sidorov: 22: 50:male

Stankewich: 21: 70:female

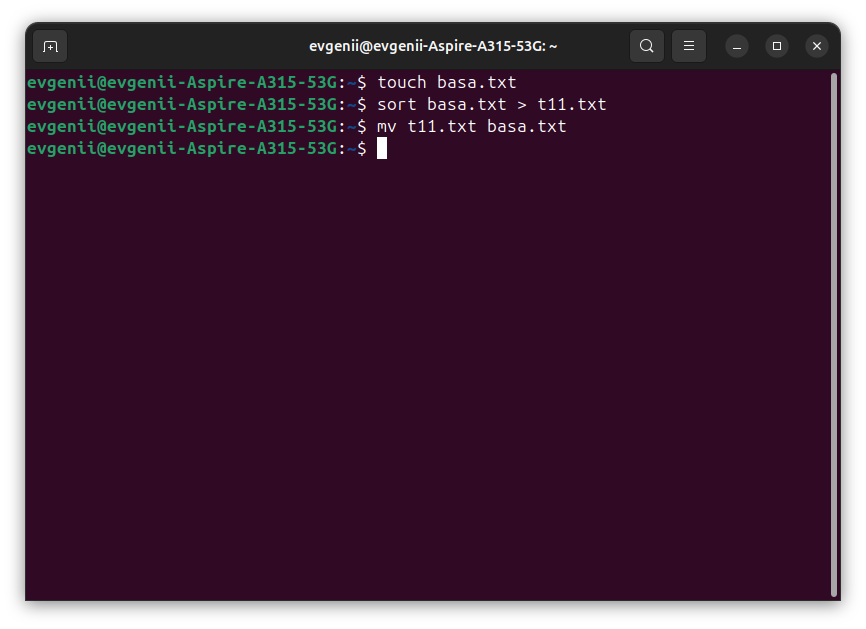
Atomova: 19: 50:female



1. Отсортируйте данные файла так, чтобы он не поменял свое собственное имя. (Решение. Нужно предварительно создать промежуточный отортированный файл t11.txt. Затем выполнить нужные команды).



Сортируем файл basa.txt посредством записи его в другой файл t11.txt



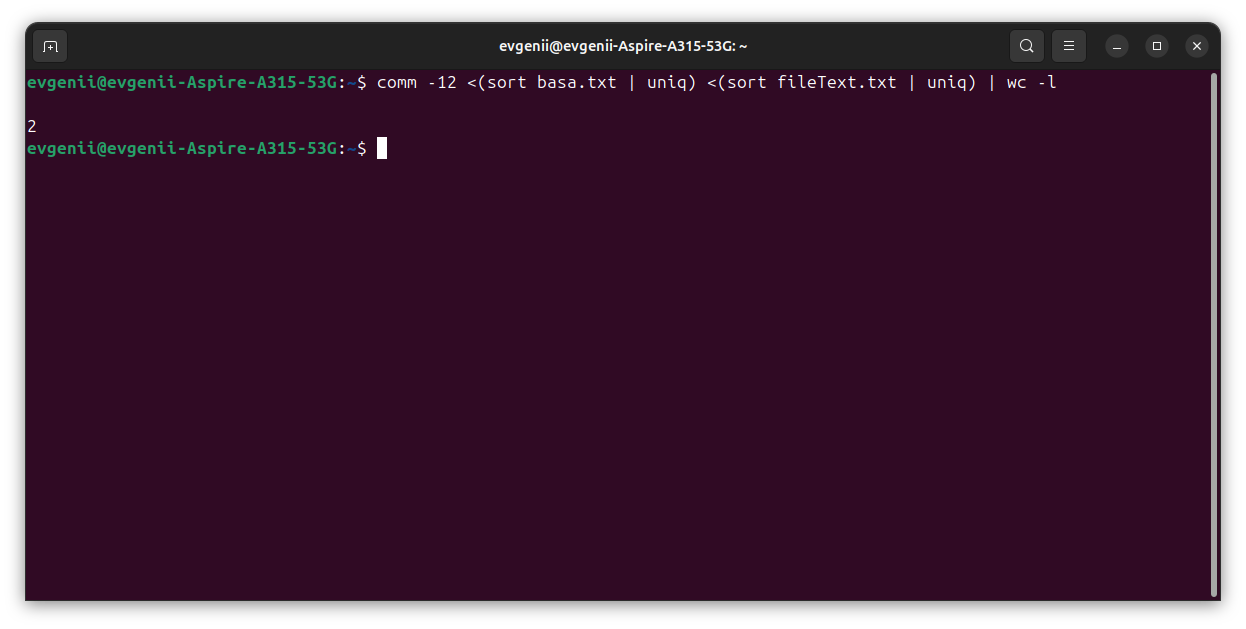
Вытащили данные в файл t11.txt и перезаписали файл с отсортированными данными в base.txt



1. Подсчитайте число одинаковых слов в обоих файлах. Для подсчета числа слов в файле используйте команду: wc -w file1.

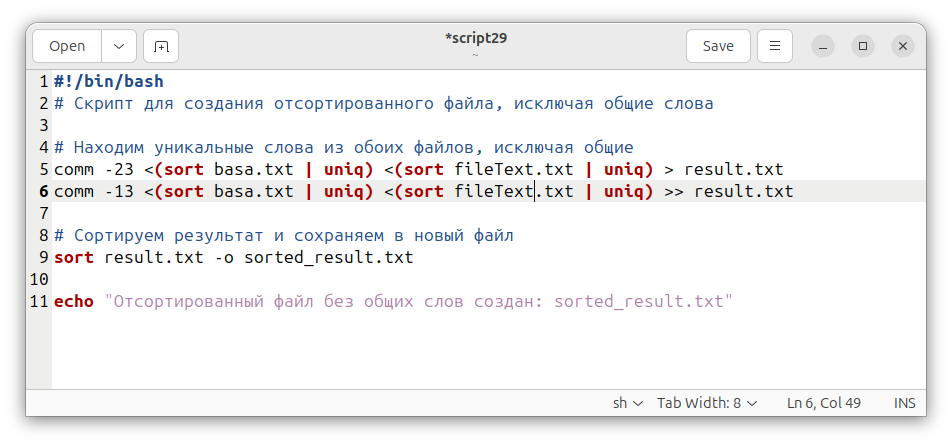


Создал файл с неизмененным текстом из первого примера



Находим число одинаковых слов в двух файлах

1. Написать скрипт, который создает отсортированный файл, содержащий слова из двух файлов, исключая их общую часть одинаковых слов.



Код программы

#!/bin/bash

# Скрипт для создания отсортированного файла, исключая общие слова

# Находим уникальные слова из обоих файлов, исключая общие

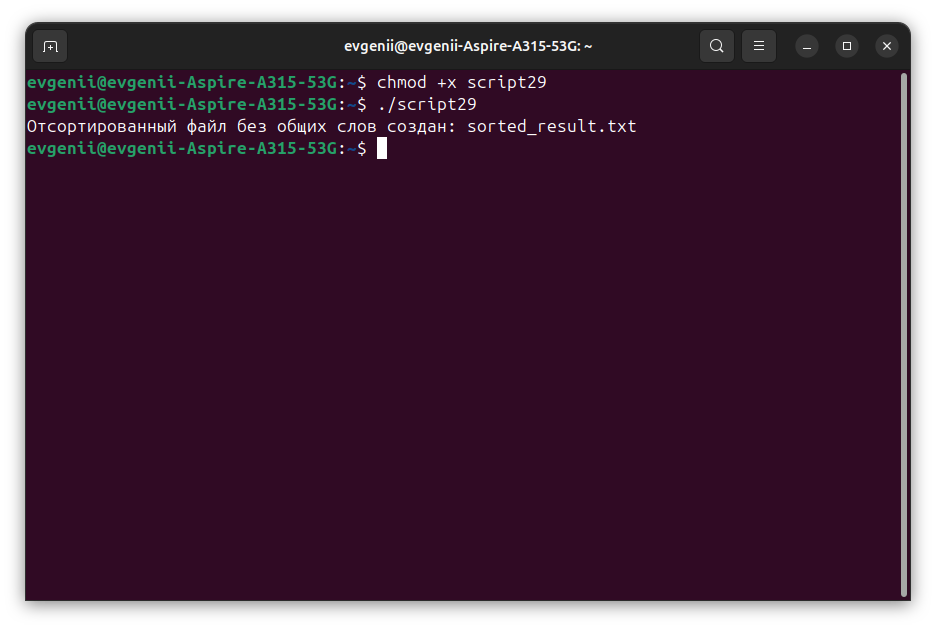
comm -23 <(sort basa.txt | uniq) <(sort fileText.txt | uniq) > result.txt

comm -13 <(sort basa.txt | uniq) <(sort fileText.txt | uniq) >> result.txt

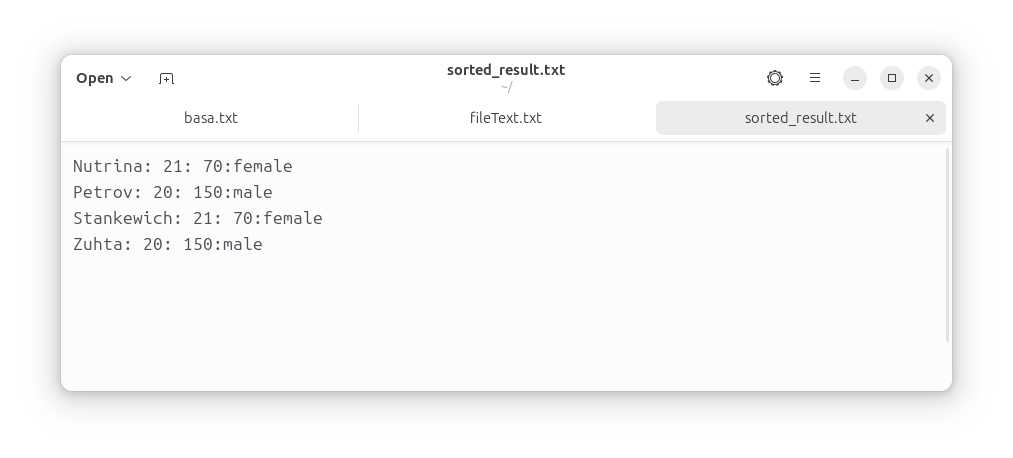
# Сортируем результат и сохраняем в новый файл

sort result.txt -o sorted\_result.txt

echo "Отсортированный файл без общих слов создан: sorted\_result.txt"

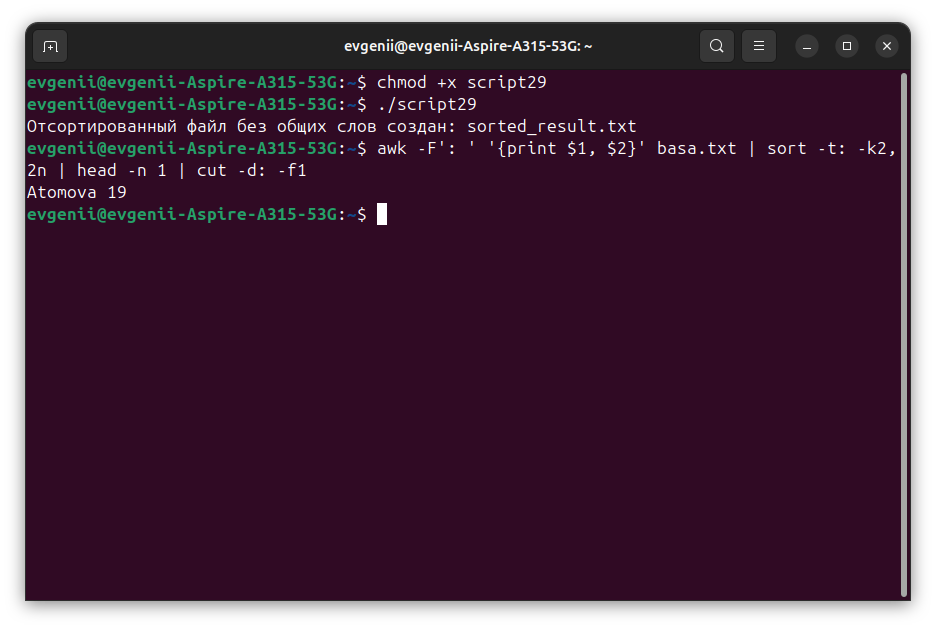


Исполняем скрипт



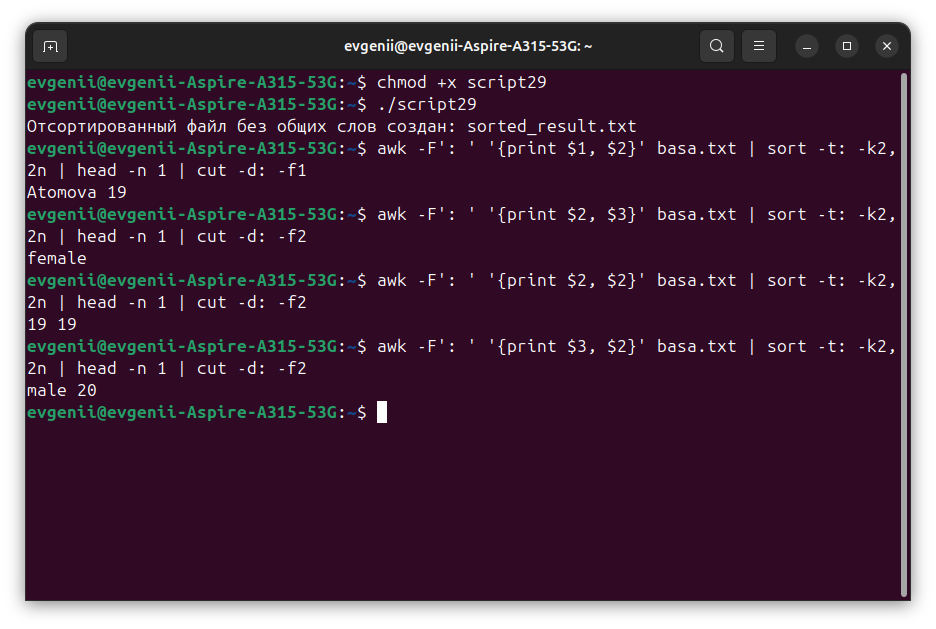
Результат видим в соответсвующем файле

1. Вывести фамилию самого молодого человека (в файле basa.txt – это второй столбец).

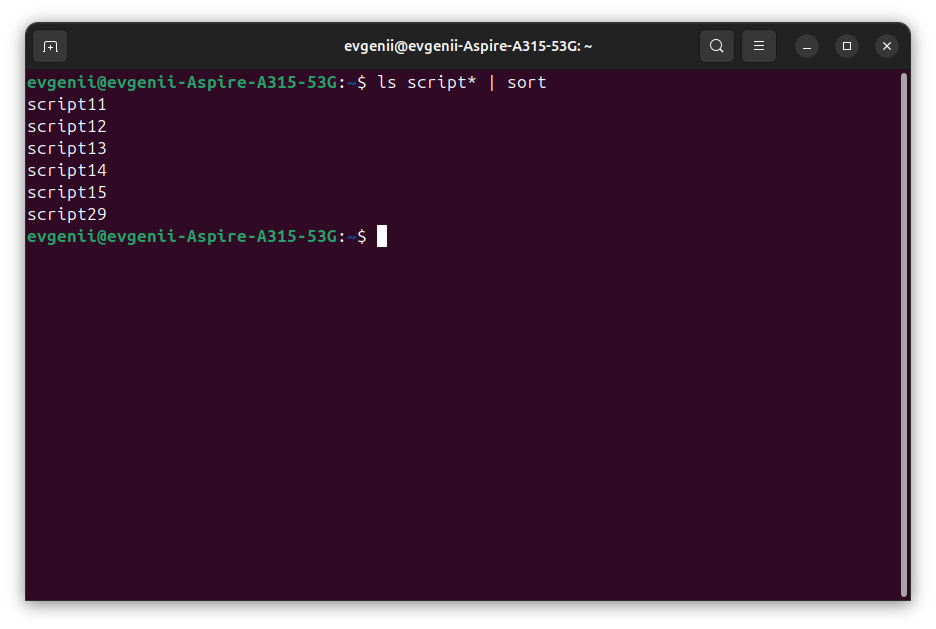


awk -F': ' '{print $1, $2}' basa.txt | sort -t: -k2,2n | head -n 1 | cut -d: -f1

1. Вывести зарплату самого молодого человека (зарплата – третий столбец).



1. Вывести отсортированный список имен файлов, начальная часть имени есть script.



13.Написать скрипт, который в каждой строке файла оставляет только буквенные символы, а остальные символы выбрасывает.

Пример.

Пусть содержимое файла есть:

Privet year 2022

Good bye year 2021.

Должны получить такой файл:

Privet year

Good bye year



Код программы

#!/bin/bash

# Скрипт для удаления всех символов, кроме буквенных

input\_file="year.txt"

output\_file="output.txt"

# Проходим по каждой строке файла

while IFS= read -r line; do

# Убираем все символы, кроме букв (A-Z, a-z, и пробелов)

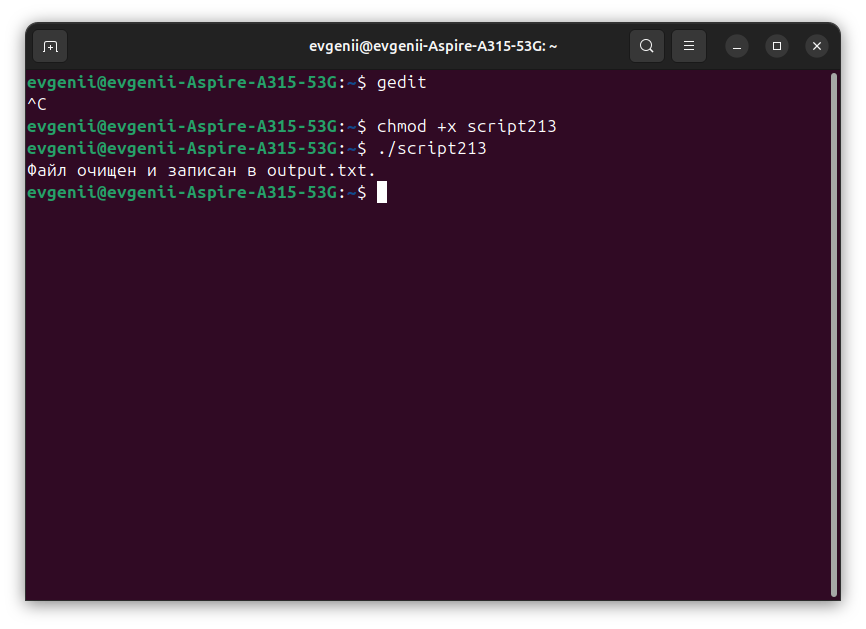
cleaned\_line=$(echo "$line" | sed 's/[^a-zA-Zа-яА-Я ]//g')

# Записываем результат в новый файл

echo "$cleaned\_line" >> "$output\_file"

done < "$input\_file"

echo "Файл очищен и записан в $output\_file."

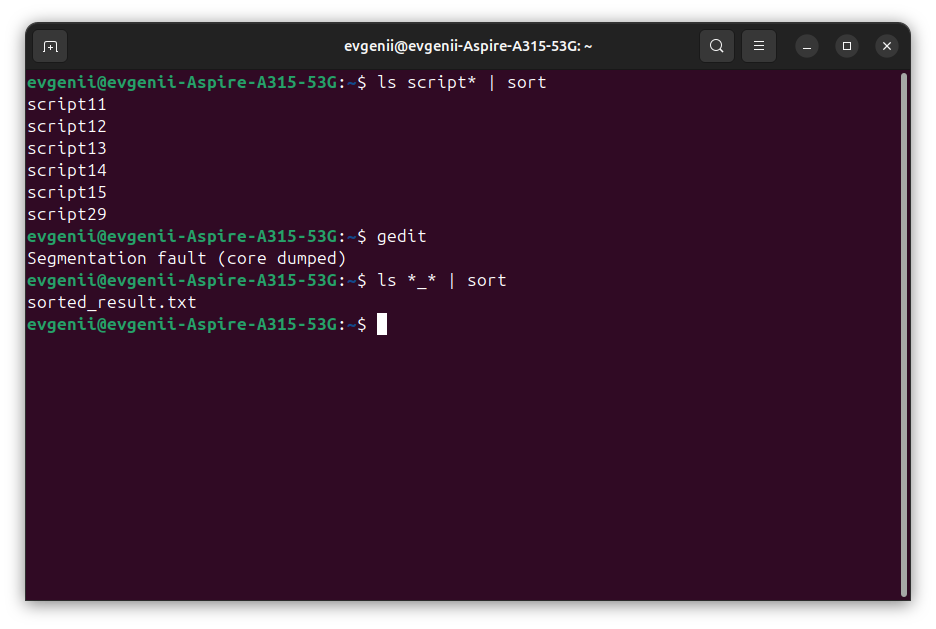


Результат работы скрипта в терминале



файл output.txt

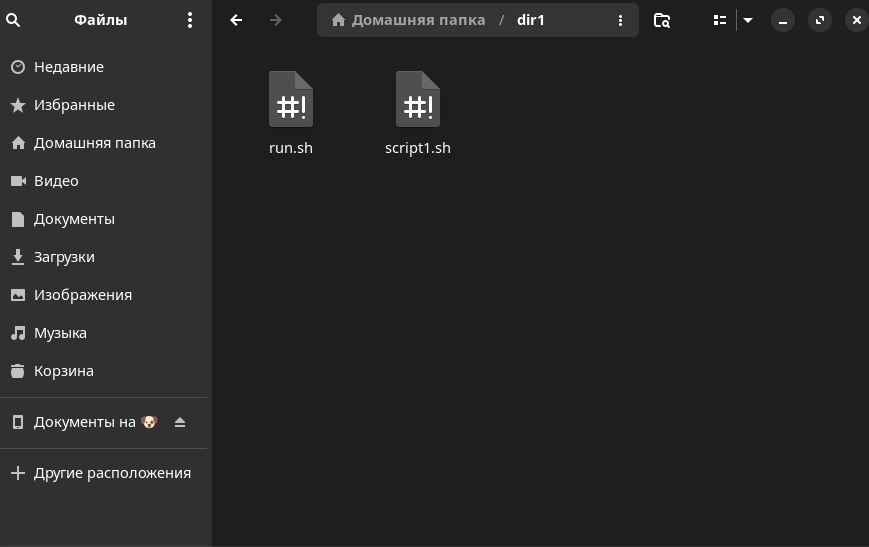
1. Вывести упорядоченный список имен файлов, в именах которых содержится символ подчеркивания, например, script\_sort.

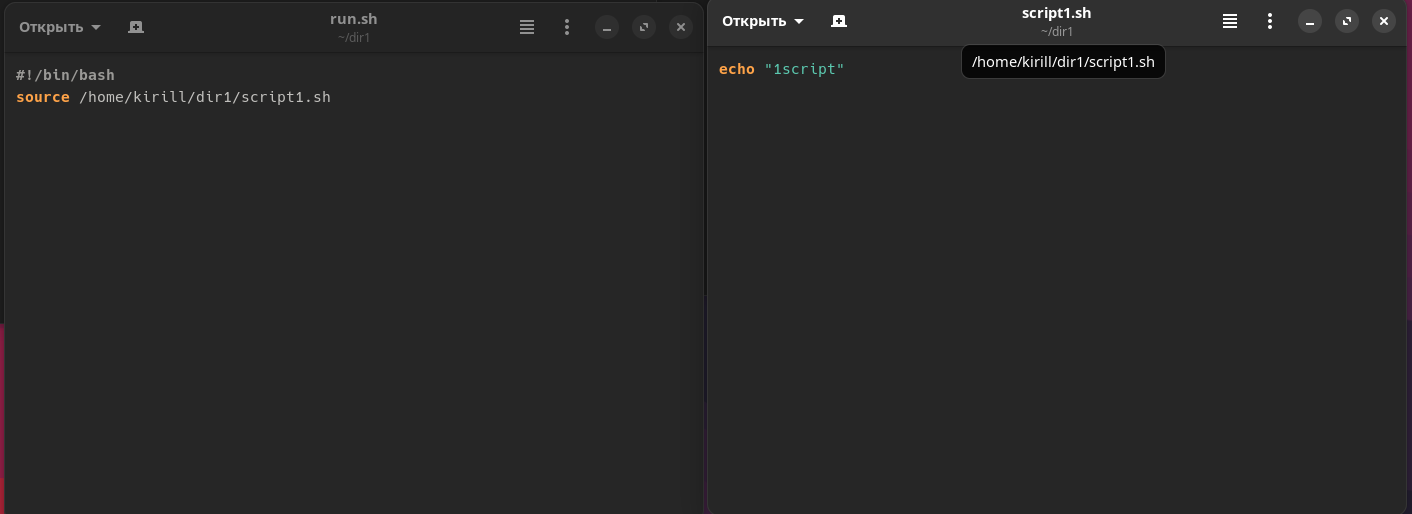


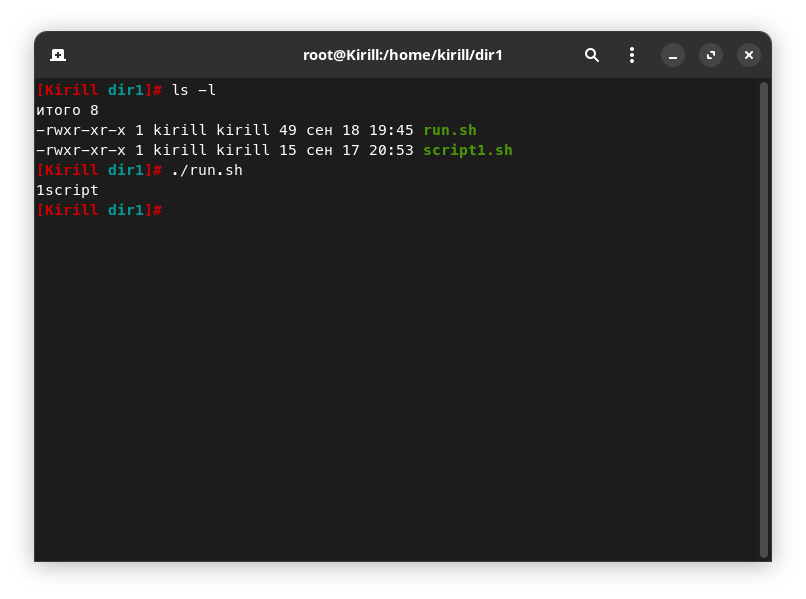
Находим файл с нижним подчеркиванием

**Задание3**

15.Напишите скрипт запуска другого скрипта из текущей директории.

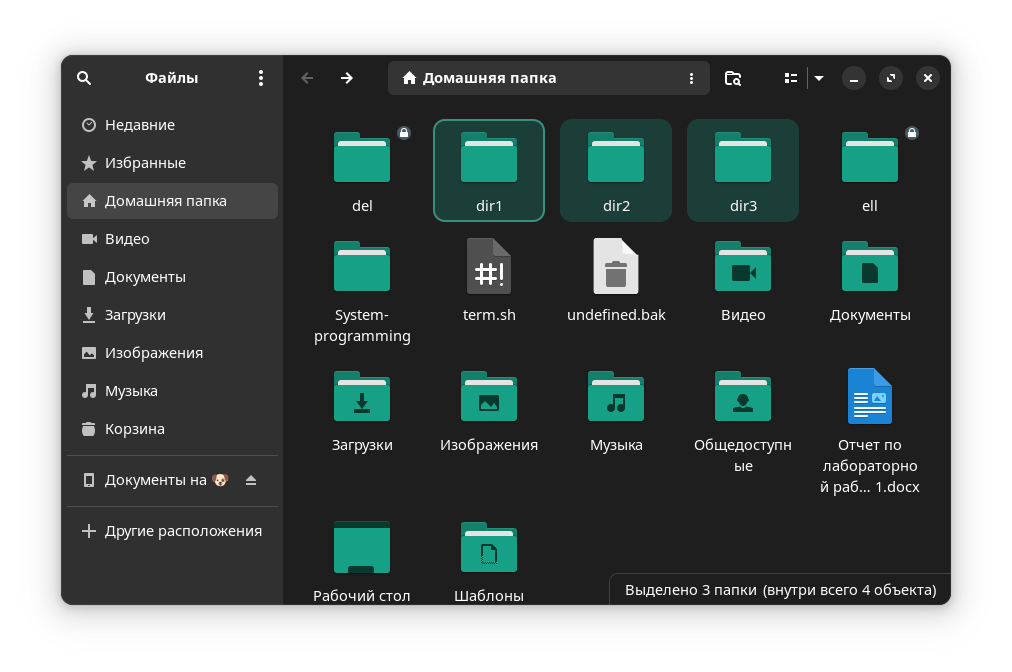
Директория dir1, создали два скрипта script1.sh (скрипт выводит текст) , run.sh ( скрипт для запуска 1 скрипта в одной директории)

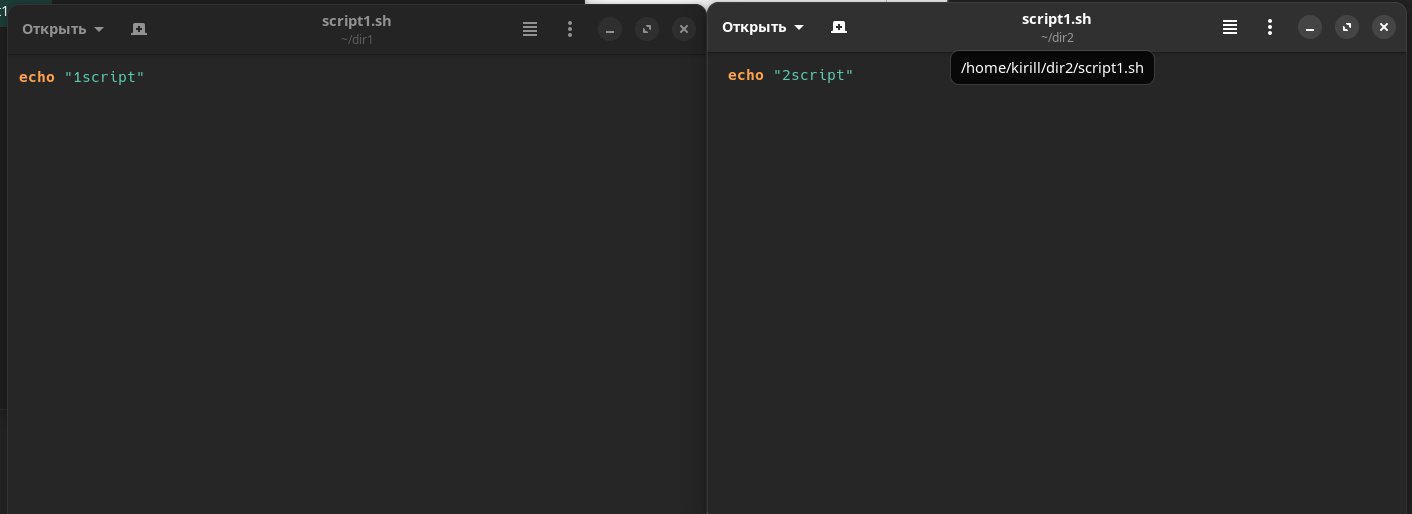
Содержимое скриптов

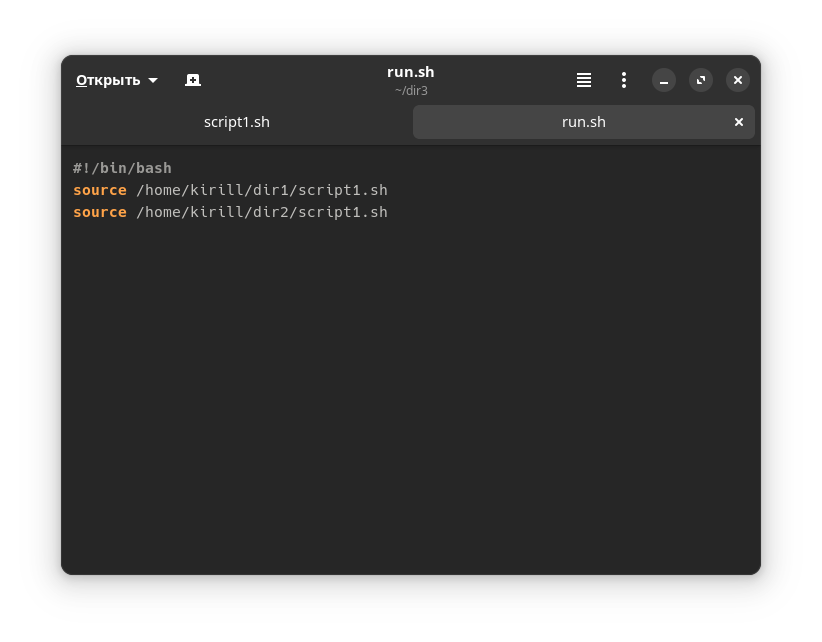
Запуск скрипта для запуска второго скрипта в этой директории

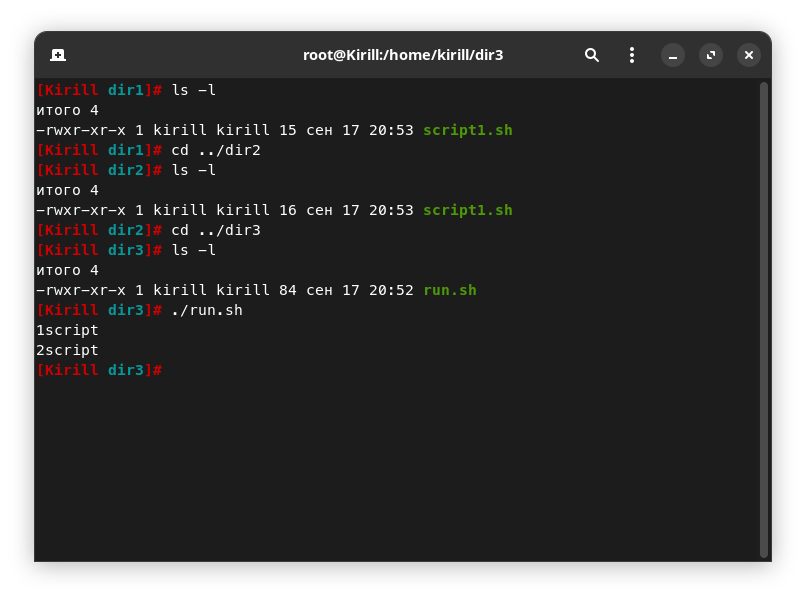
16.Создайте два скрипта с одинаковым именем, но разным текстом.

Эти скрипты должны быть в разных директориях. В третьей директории создайте скрипт, запускающий два первых скрипта. При этом используйте команду **sours**. Объясните поведение программы и расскажите с какими проблемами вы столкнулись.

Создали 3 директории с разными скриптами

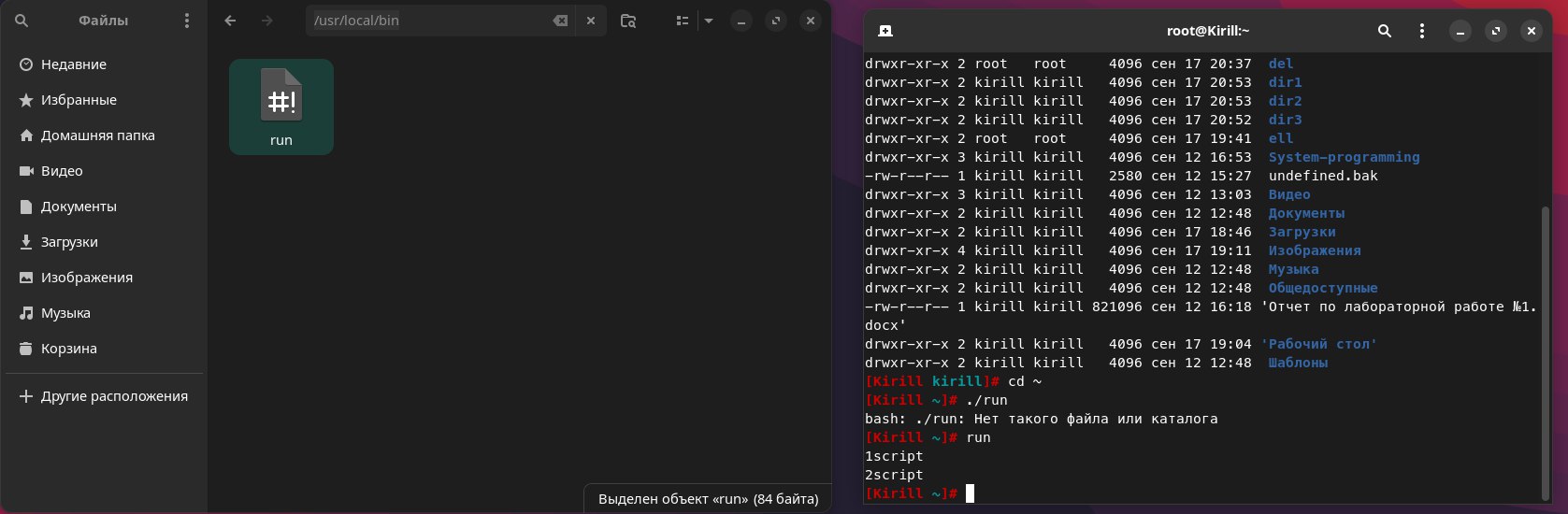
2 одинаково названных скрипта находятся в разных директориях

Скрипт запуска двух одинаково названных скрипта

Работа скрипта запуска

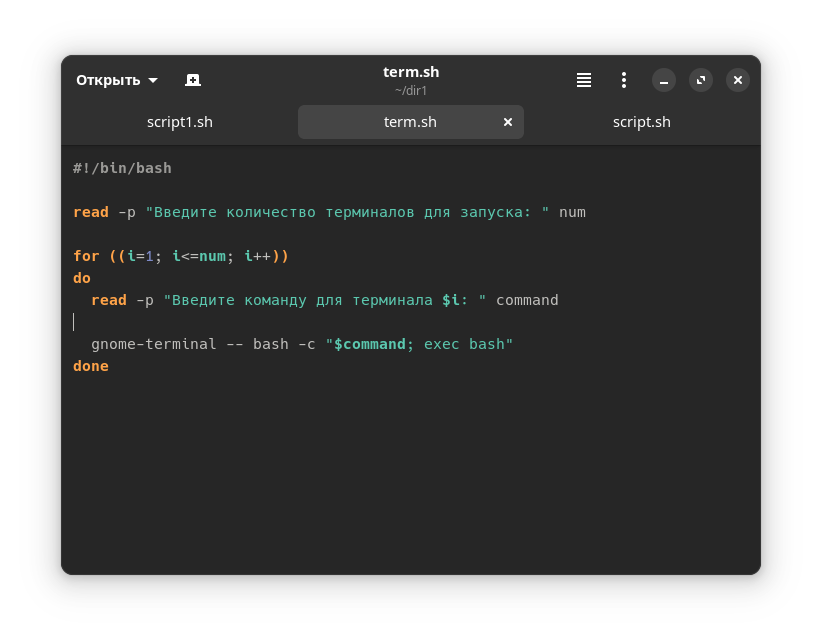
17.Расположите скрипт в одной из стантадртных директорий

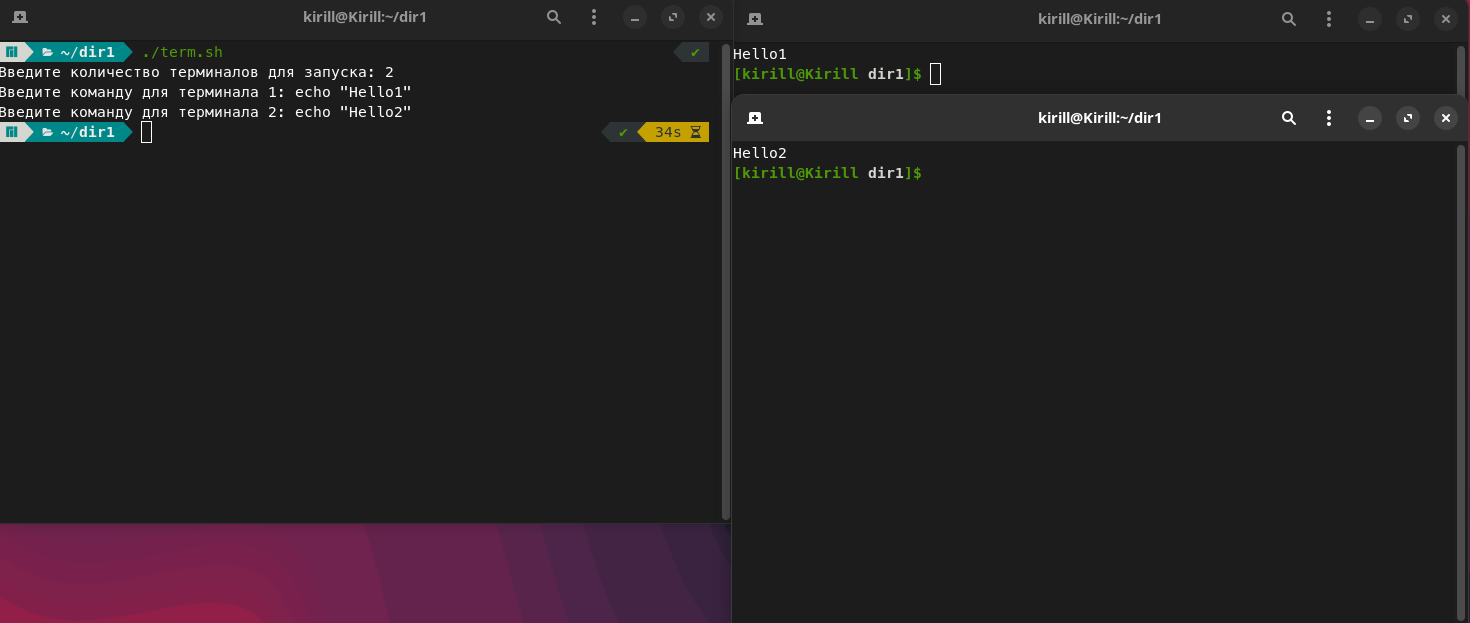
системы, и запустите его из домашней директории пользователя.

Скрипт run перенесен в директорию /bin

Комманда sudo cd (адрес файла) (адрес переноса)

1. Напишите скрипт запускающий два и более терминалов отдельно(количество терминалов зависит от количества голов в команде). Количество терминалов должне вводиться интерактивно. Желательно чтобы в этих терминалах запускалисть какие нибудь программки, можно запустить браузер.

Скрипт запускающий два и более терминалов отдельно.



**Ответы на вопросы**

Медведский Е.В.

1. Какое значение имеет месторасположение скриптов в файловой системе для их запуска?

Для запуска скрипта желательно иметь скрипт в той дериктории, где происходит работа, т.к. он может влиять на файлы в важных для системы дерикториях. А так его местоположение важно если там используются относительные пути, если мы ищем файлы в настоящем катологе(в том, котором мы находимся)

10)Для чего используются команды-фильтры? Назовите некоторые

из них.

**Команды-фильтры** используются для обработки и трансформации данных, поступающих из стандартного ввода или файла, и передачи обработанных данных на стандартный вывод или в другой файл. Такие команды работают по принципу "получить данные, обработать их, передать дальше". Фильтры широко применяются в UNIX-подобных системах для обработки текстов, поиска, сортировки и выполнения других операций с потоками данных.

grep

* Используется для поиска строк, соответствующих регулярным выражениям.

grep "pattern" file.txt

sort

* Сортирует строки входных данных по алфавиту или числовому значению.

sort file.txt

uniq

* Удаляет повторяющиеся строки в отсортированном файле.

sort file.txt | uniq

awk

* Мощная утилита для обработки текстовых файлов, позволяет выбирать определённые столбцы, строки и выполнять операции над ними.

awk '{print $2}' file.txt

Выведет второй столбец из каждой строки файла file.txt.

cut

* Извлекает определённые части строк, разделённых символами.

cut -d ':' -f 1 file.txt

Извлекает первую часть строк в файле file.txt, где части разделены двоеточием

wc **(word count)**

* Подсчитывает количество строк, слов и символов в файле
* wc -w file.txt

Вопросы Зухта К.М

1. Что такое скрипт или сценарий командной оболочки, для чего они используются?

Ответ: Скрипт или сценарий командной оболочки — это текстовый файл, содержащий последовательность команд, которые могут быть выполнены в командной строке операционной системы, обычно в Unix-подобных системах (например, Linux). Скрипты написаны на языках командной оболочки, таких как Bash, Zsh и других.

Примеры использования скриптов: - **Резервное копирование данных:** Скрипты могут автоматически копировать критически важные файлы в заданные директории или на удаленные серверы. - **Управление системными процессами:** Запуск, остановка или перезапуск сервисов. - **Обработка текстовых файлов:** Поиск, замена, фильтрация и форматирование данных в текстовых файлах.

2. Назовите три основновополагающие концепции операционных систем и системного программирования?

Ответ: 1. **Управление процессами**

**2. Управление памятью**

**3. Управление файловыми системами**