МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

## Факультет информационных технологий и робототехники

Кафедра программного обеспечения информационных систем

и технологий

**Отчет**

**по лабораторной работе № 3**

по дисциплине: ”Системное программирование”

на тему: ***”*****Командный язык и скрипты Shell*”***

Выполнил**:** студент группы *10702121* Мялик А. С.

Принял**:** пр. Давыденко Н. В.

Минск 2023

# Лабораторная работа №3.

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

## Задание 1

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

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

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

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

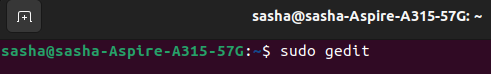
5. А какие командные интерпретаторы установлены в вашей системе?

Напишите скрипт поиска местонахождения командных интерпретаторов вашей системы. Также определите какой интерпретатор используется сейчас в вашей системе?

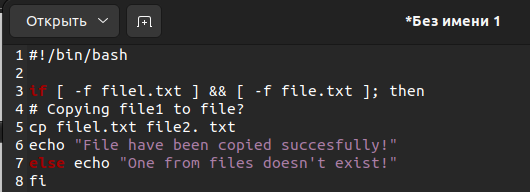
Решение

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

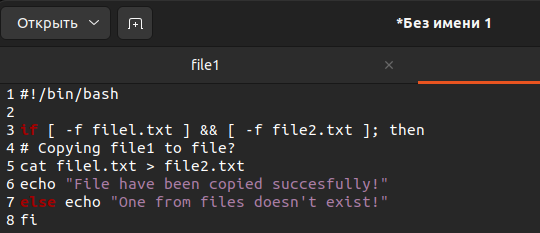
1. Вызываем текстовый редактор



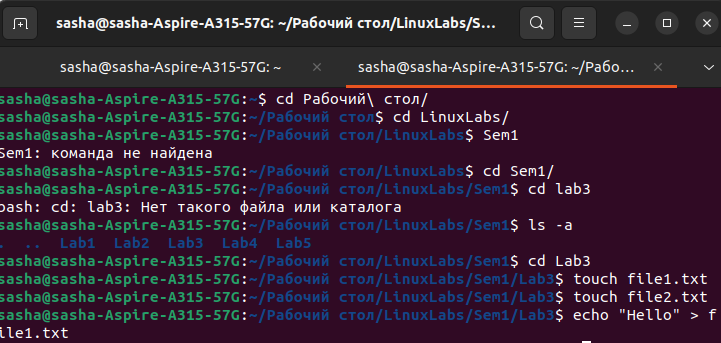
1. Пишем код сценария и сохраняем файл

 1) Используя команду `cp` для копирования файла:

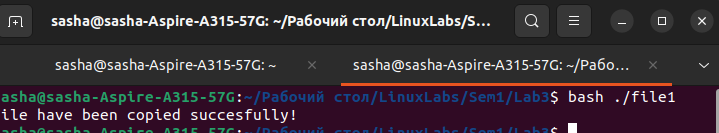
2) Используя оператор `cat` для чтения и записи файлов:



3. Создаем два текстовых файла и заносим текст в один из них.



4. Запускаем скрипт



5. Проверяем наш второй файл:



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

1. Выполняем команду find

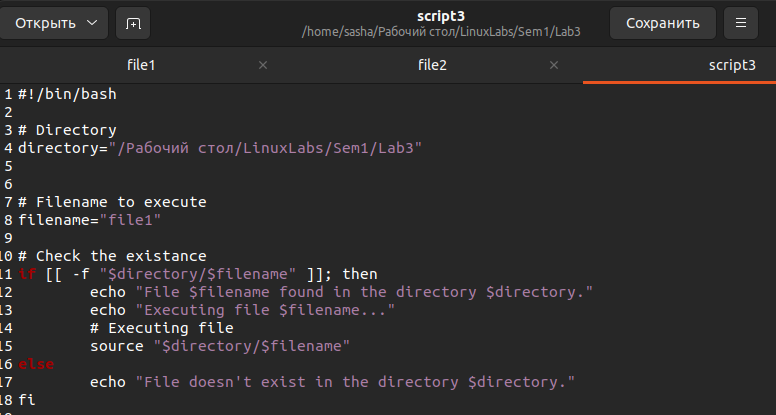


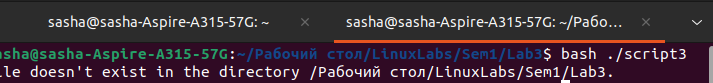
2. Проверяем результат



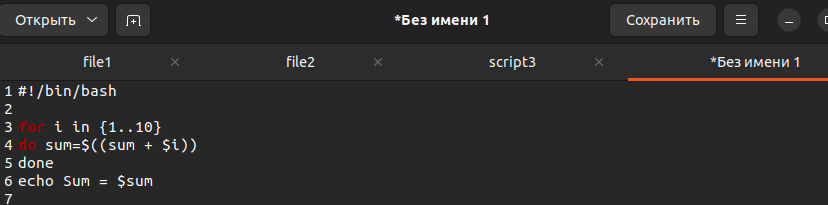
**3.** Написать скрипт, который проверяет, содержится ли скриптовый файл в директории, если да, то выполнить его.

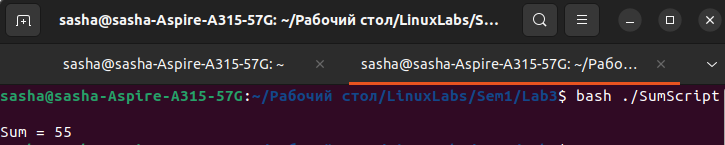
1. Пишем скрипт.

 2. Запускаем и проверяем работу



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

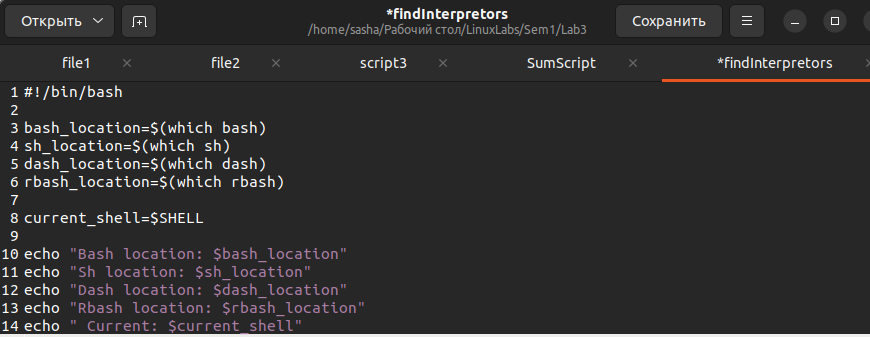


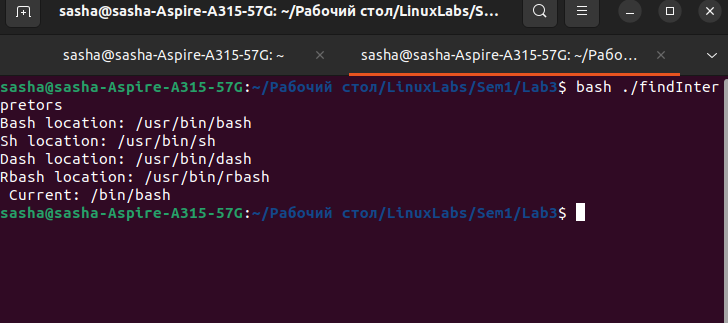


**5.** А какие командные интерпретаторы установлены в вашей системе?

sh, bash, rbash, dash

Напишите скрипт поиска местонахождения командных интерпретаторов вашей системы. Также определите какой интерпретатор используется сейчас в вашей системе?





**Задание 2.**

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

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

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

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

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

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

12.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.

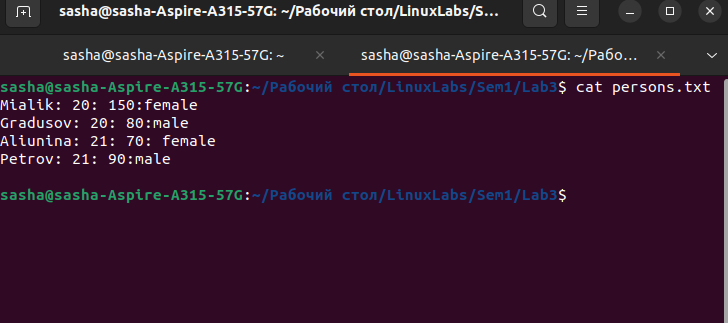
Должны получить такой файл:

Privet year

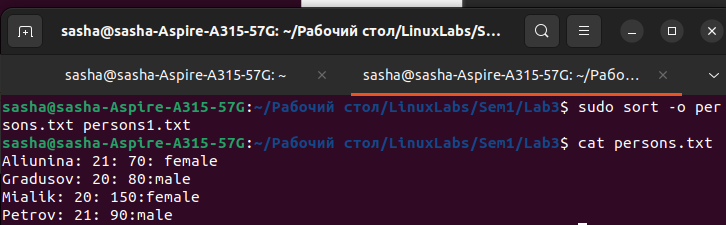
Good bye year

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

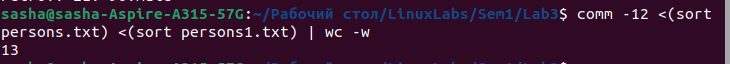
**6.** Используйте текст первого примера, заменить некоторые фамилии на фамилии членов вашей команды (например, Nutrina заменить на Stankewich).



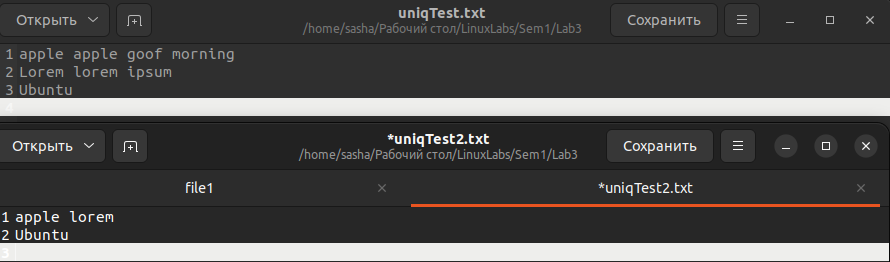
**7.** Отсортируйте данные файла так, чтобы он не поменял свое собственное имя. (Решение. Нужно предварительно создать промежуточный отсортированный файл t11.txt. Затем выполнить нужные команды).



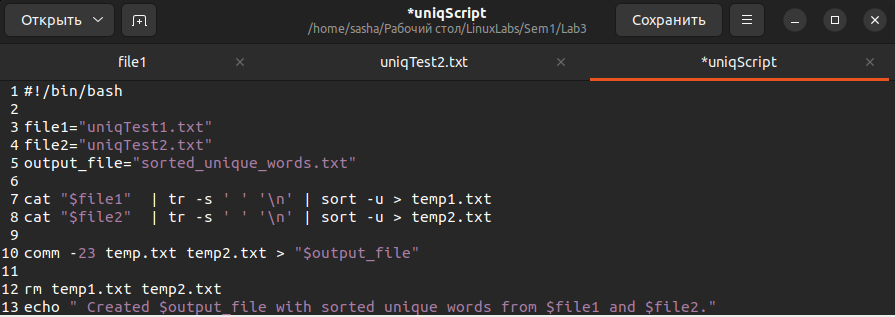
**8.** Подсчитайте число одинаковых слов в обоих файлах. Для подсчета числа слов в файле используйте команду: wc -w file1.

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

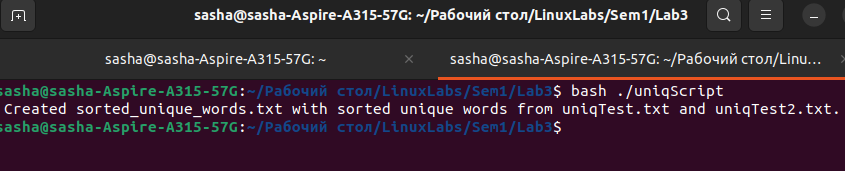
1. Создаем два файла и заполняем их.

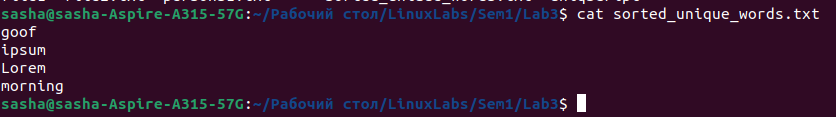


2. Пишем скрипт.

Здесь мы при помощи tr -s разделяем заменяем пробелы, на символ '\n', то есть переводим слова из одной строки на новые. Затем при помощи sort -u сортируем файл и удаляем дубликаты. И записываем результаты в два временных файла. Потом при помощи comm -23 сравниваем содержимое временных файлов и выводим только строки, которые есть только в первом файле (**-2**) и только во втором файле (**-3**).

3. Проверяем результат.

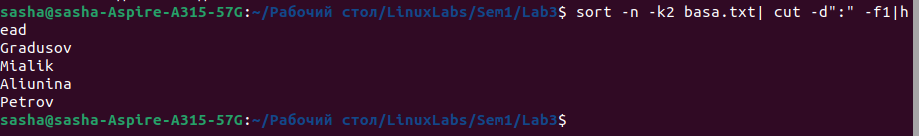




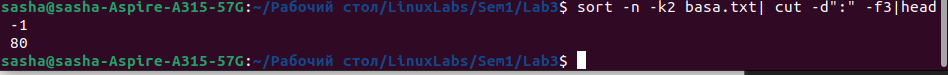
**10.** Вывести фамилию самого молодого человека (в файле basa.txt – это второй столбец).

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание



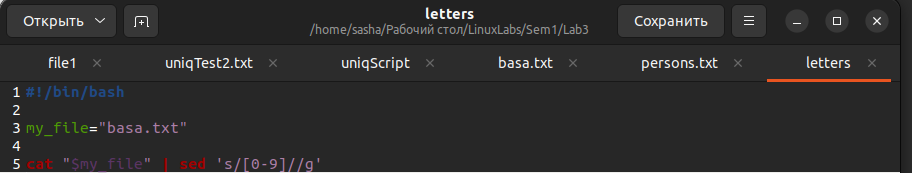
**11.** Вывести зарплату самого молодого человека (зарплата – третий столбец).

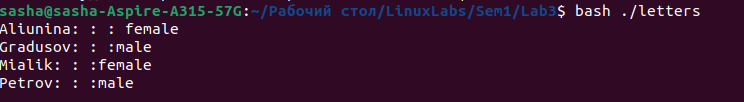


**12.** Вывести отсортированный список имен файлов, начальная часть имени есть script.



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





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

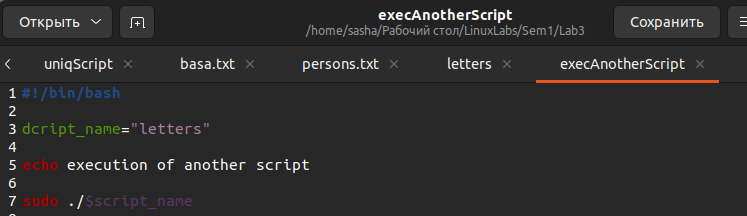


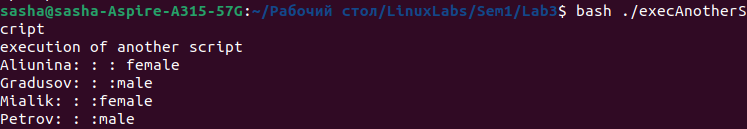
**Задание 3.**

15.Напишите скрипт запуска другого скрипта.

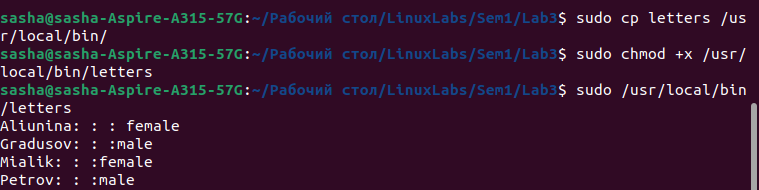
16.Расположите скрипт в одной из стантадртных директорий системы, и запустите его из домашней директории пользователя.

**15.** Напишите скрипт запуска другого скрипта.





**16.** Расположите скрипт в одной из стантартных директорий системы, и запустите его из домашней директории пользователя.



Контрольные вопросы:

10) Для чего используются команды фильтры? Назовите некоторые из них.

Команды-фильтры (или просто фильтры) в Linux используются для обработки и манипуляции текстовыми данными, которые потоково передаются через конвейер (pipe) или из файлов. Они позволяют фильтровать, преобразовывать и анализировать текстовую информацию. Вот несколько распространенных команд-фильтров в Linux:

grep: Используется для поиска текста в файлах или выводе команды. Он фильтрует строки, соответствующие заданному регулярному выражению.

sed: Позволяет выполнять текстовую замену и преобразования в потоке данных. Это мощный текстовый редактор на основе скриптов.

awk: Программированный язык для обработки и анализа текстовых данных. Он может выполнять различные операции, такие как извлечение и манипуляция данными в столбцах.

cut: Используется для вырезания (удаления) определенных столбцов или символов из каждой строки текста.

sort: Сортирует строки текста по заданным критериям, например, в алфавитном порядке.

uniq: Удаляет дубликаты последовательных строк в текстовых данных. Обычно используется совместно с sort.

tr: Преобразует (транслирует) одни символы в другие. Он может использоваться, например, для изменения регистра букв или замены символов.

wc: Считает количество строк, слов и символов в текстовом вводе или файле.

head и tail: Они используются для вывода начальных или конечных строк из файла или потока.

tee: Перенаправляет поток вывода в файл и, при этом, также отправляет его на стандартный вывод.

Эти команды-фильтры позволяют легко выполнять различные манипуляции с текстовыми данными в командной строке Linux, что делает их мощными инструментами для обработки и анализа информации.