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 10701321 Кирилюк А.А.

Принял**:** ст.пр. Давыденко

Минск 2023

# **Лабораторная работа № 3**

**Цель работы**

Закрепить на практике основы работы со сценариями командной обо-

лочки 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.



Рис. 1. Скрипт 1



Рис. 2. Скрипт 2

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

файл list.txt.

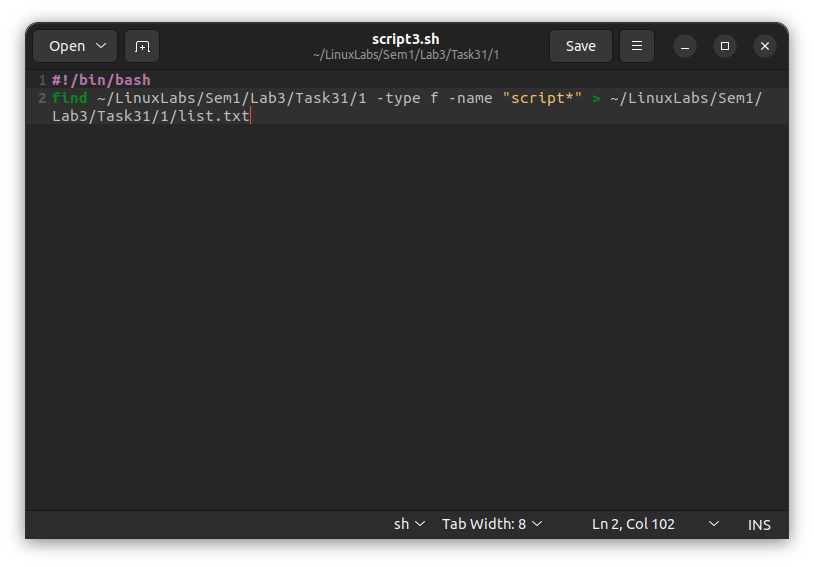


Рис. 3. Скрипт

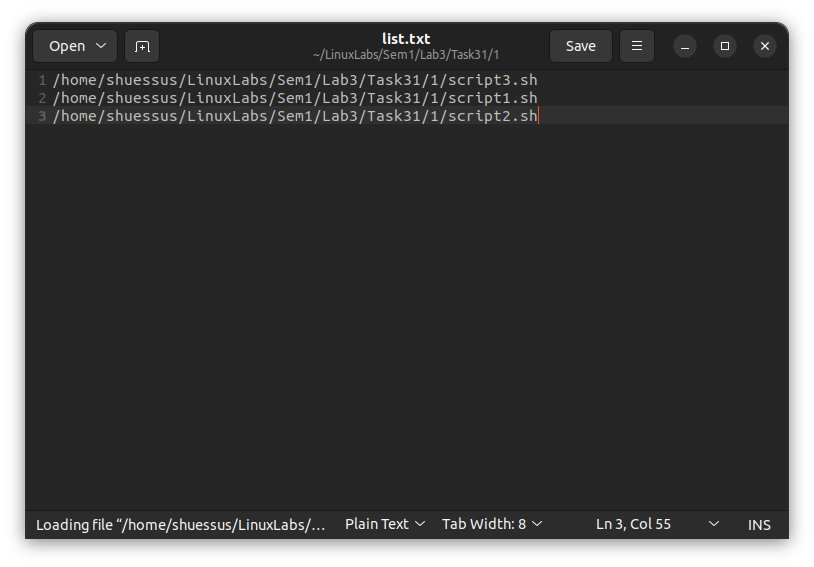


Рис. 4. Содержимое файла list.txt

3. Написать скрипт, который проверяет, содержится ли скриптовый файл в

директории, если да, то выполнить его.



Рис. 5. Скрипт

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



Рис. 6. Скрипт

5. А какие командные интерпретаторы установлены в вашей системе?

Напишите скрипт поиска местонахождения командных интерпретато-

ров вашей системы. Также определите какой интерпретатор использу-

ется сейчас в вашей системе?

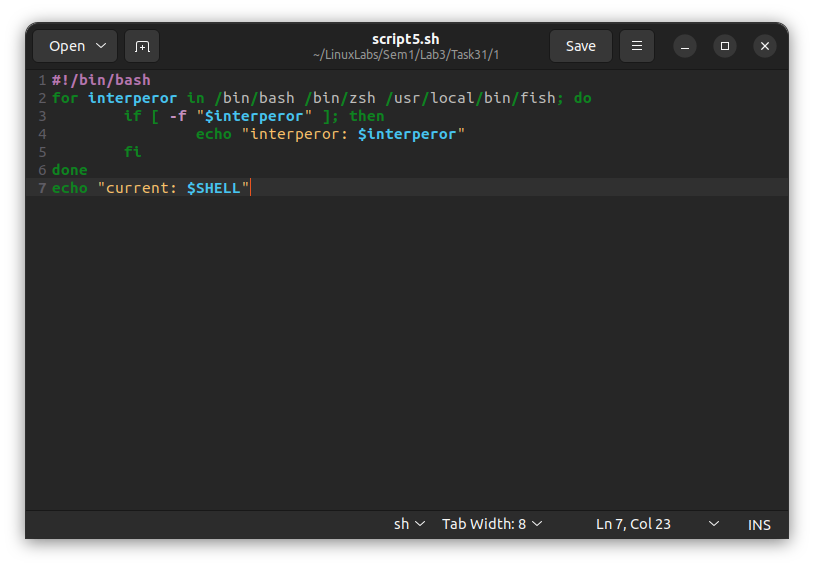


Рис. 7. Скрипт

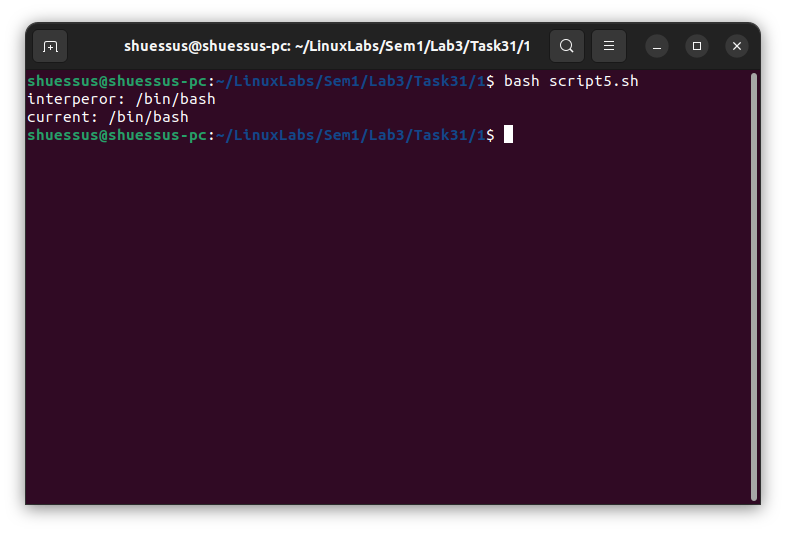


Рис. 8. Результат выполнения скрипта

**Задание 2**

6. Используйте текст первого примера, заменить некоторые фамилии на

фамилии членов вашей команды (например, Nutrina заменить на

Stankewich).

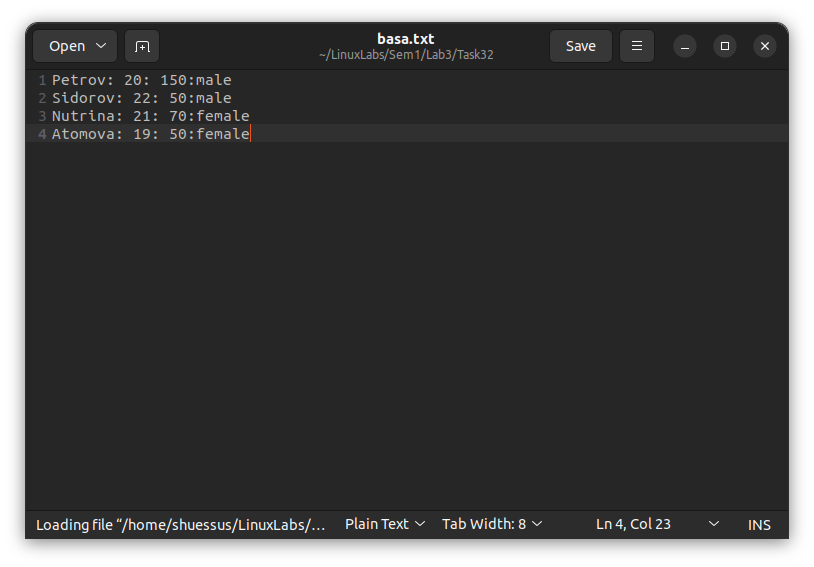


Рис. 9. Файл без замены

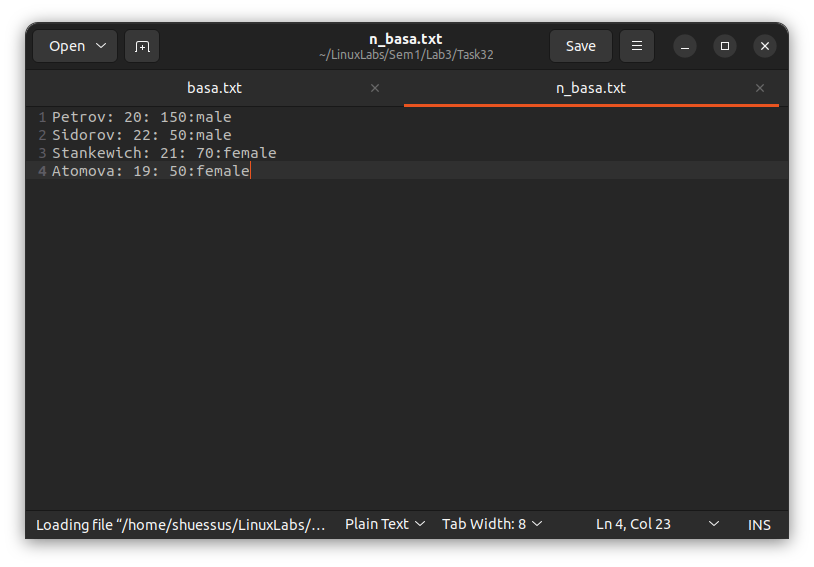


Рис. 10. Файл с заменой



Рис. 11. Скрипт

7. Отсортируйте данные файла так, чтобы он не поменял свое собствен-

ное имя. (Решение. Нужно предварительно создать промежуточный от-

сортированный файл t11.txt. Затем выполнить нужные команды).

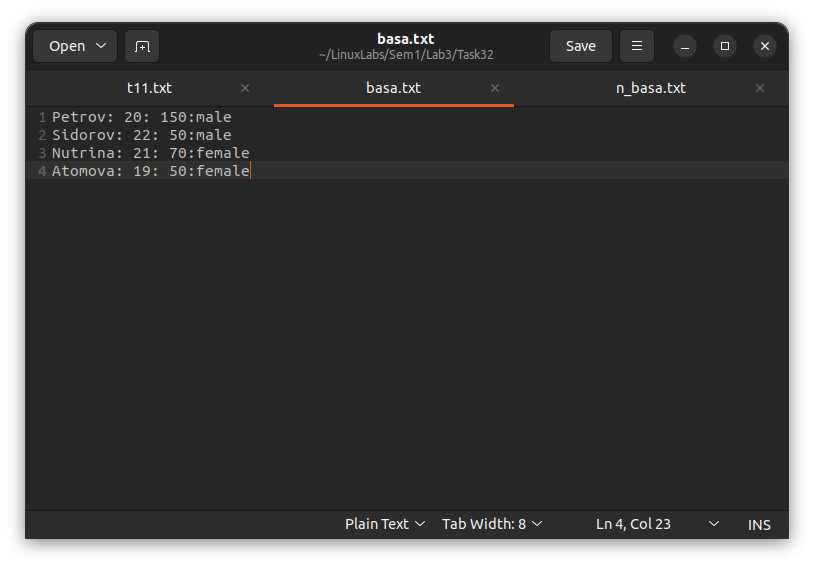


Рис. 12. Файл до сортировки

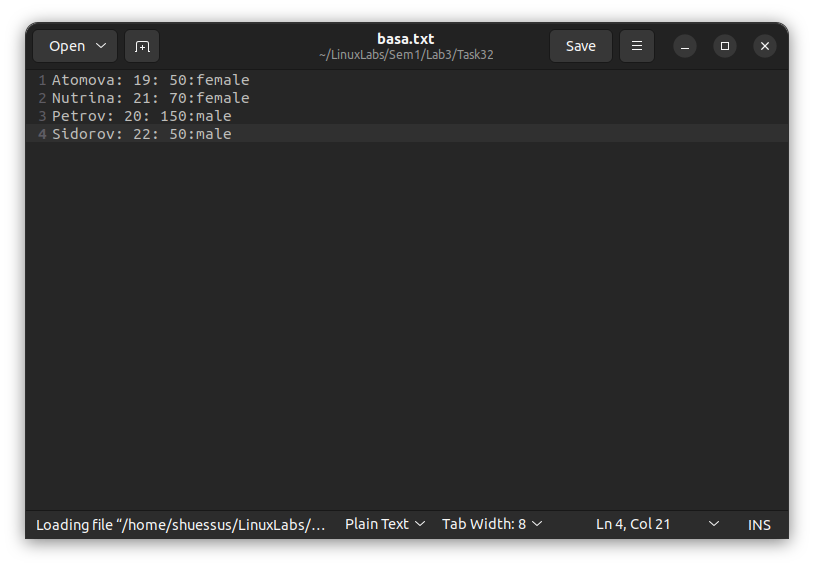


Рис. 13. Файл после сортировки



Рис. 14. Скрипт

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.

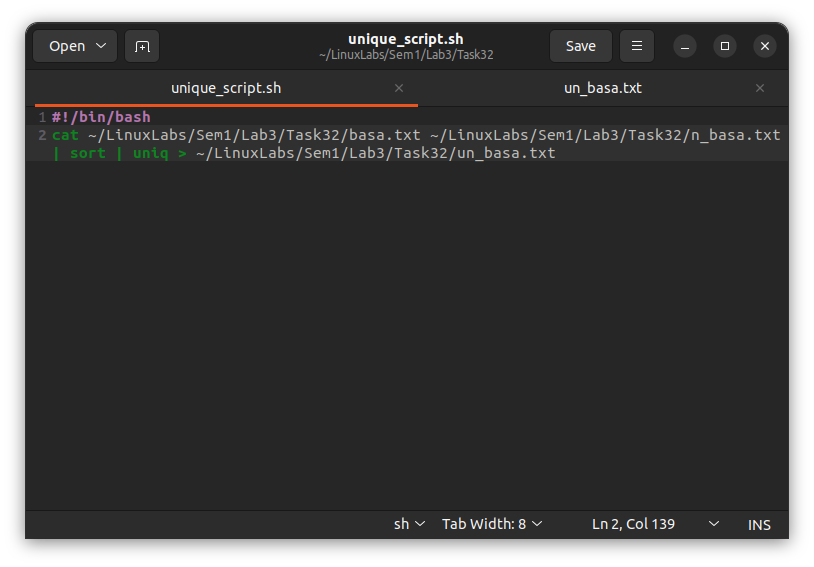


Рис. 15. Скрипт

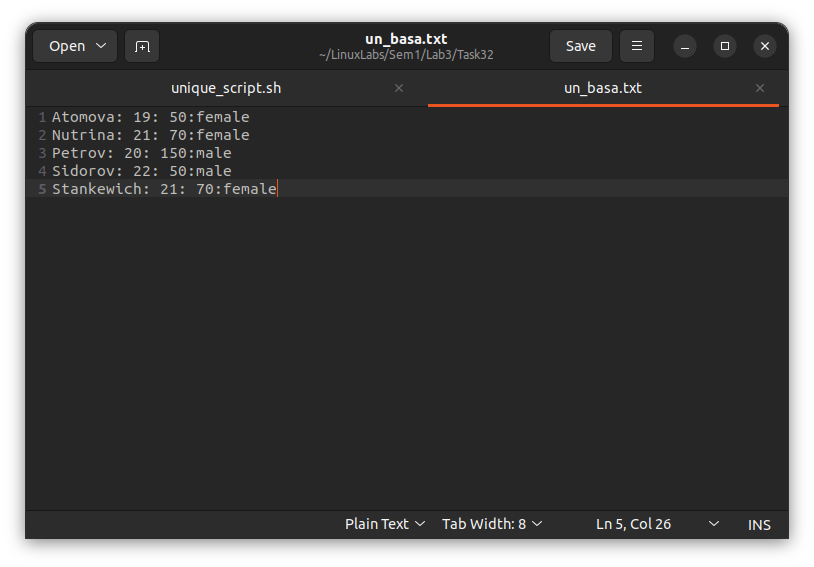


Рис. 16. Содержание файла

10. Вывести фамилию самого молодого человека (в файле basa.txt – это вто-

рой столбец).



Рис. 17. Скрипт

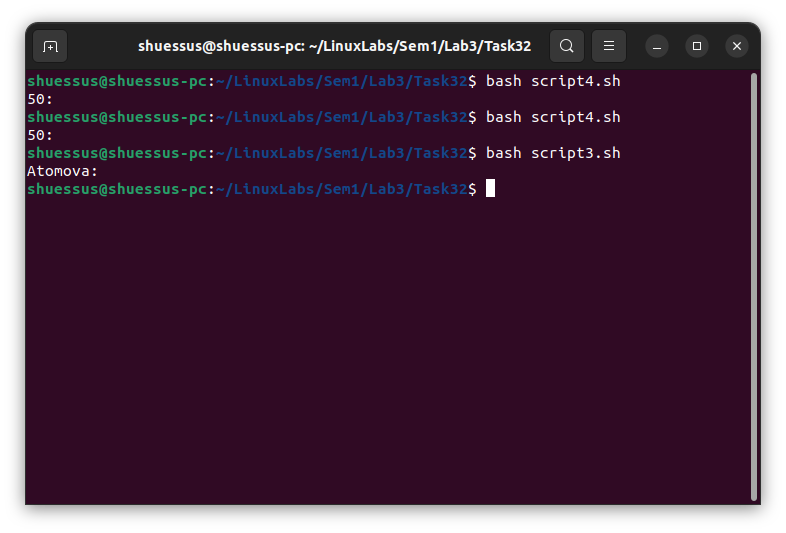


Рис. 17. Результат

11. Вывести зарплату самого молодого человека (зарплата – третий стол-

бец).

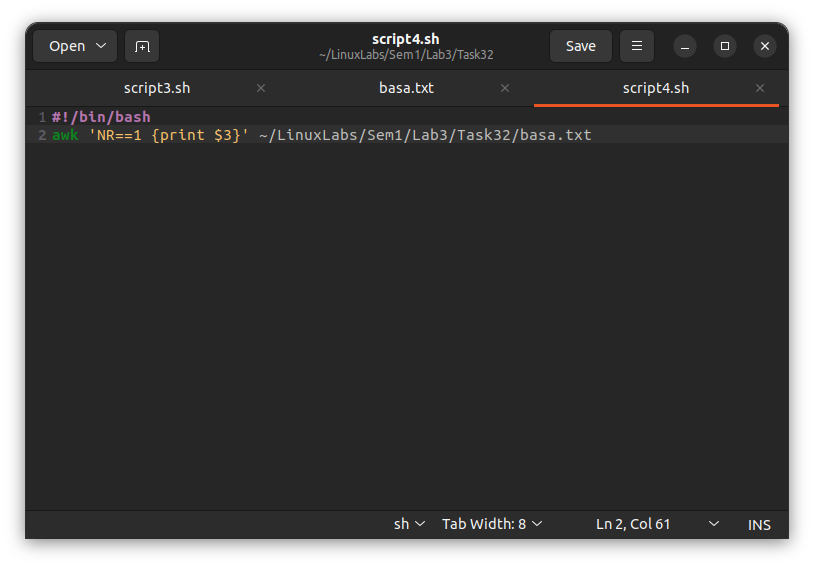


Рис. 18. Скрипт

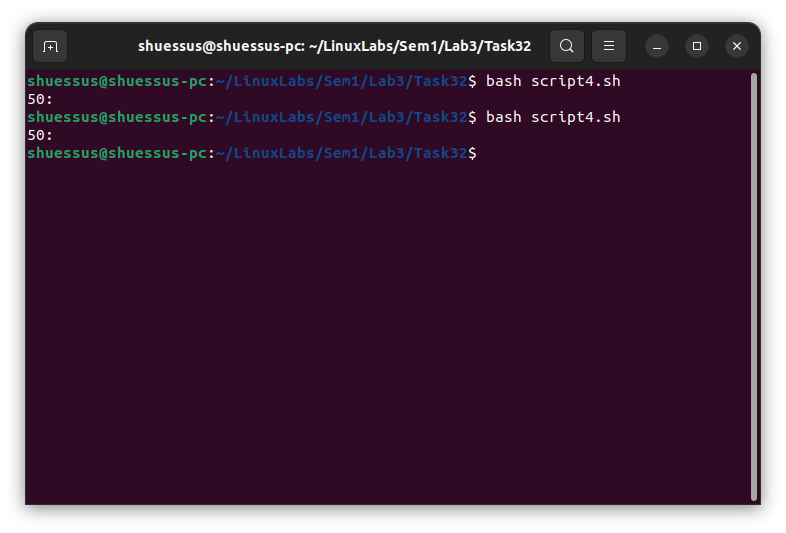


Рис. 19. Результат

12. Вывести отсортированный список имен файлов, начальная часть имени

есть script.

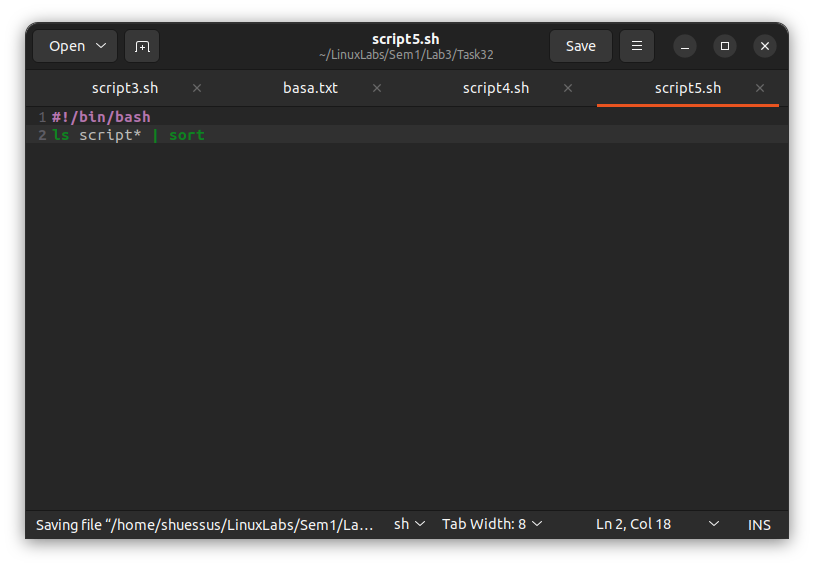


Рис. 20. Скрипт

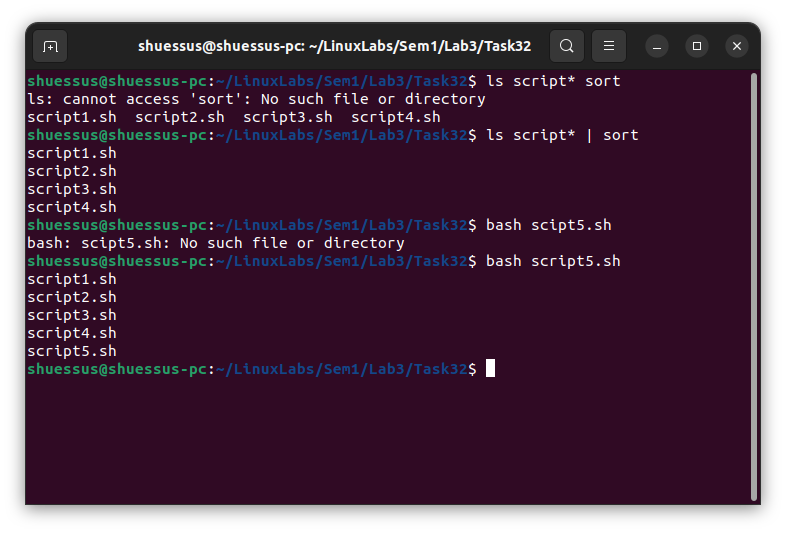


Рис. 21. Результат

13. Написать скрипт, который в каждой строке файла оставляет только

буквенные символы, а остальные символы выбрасывает.

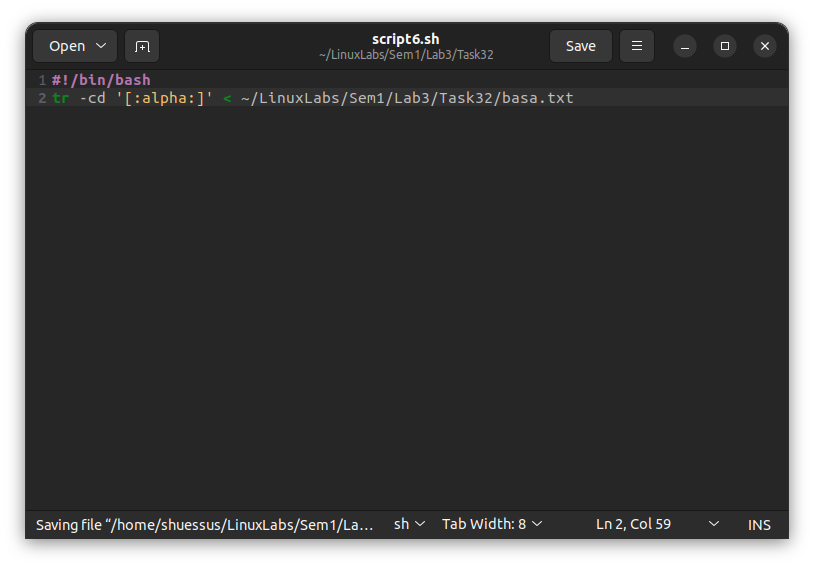


Рис. 22. Скрипт

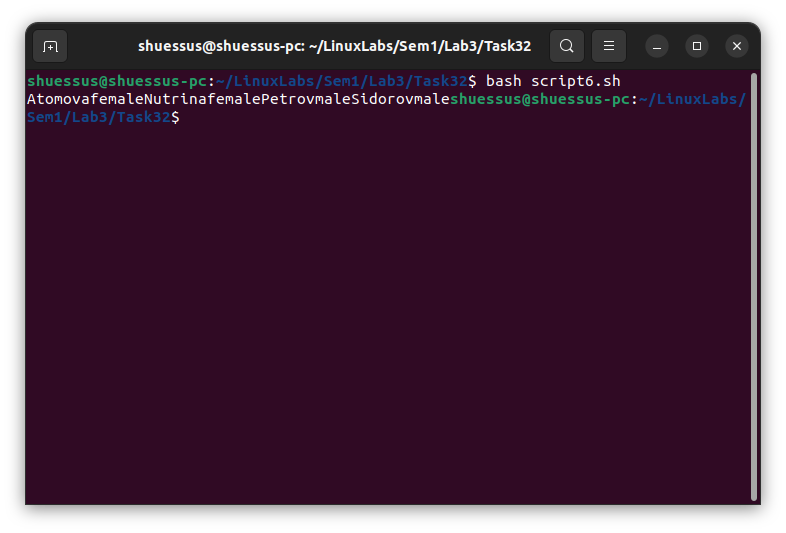


Рис.23. Результат

14. Вывести упорядоченный список имен файлов, в именах которых содер-

жится символ подчеркивания, например, script\_sort.

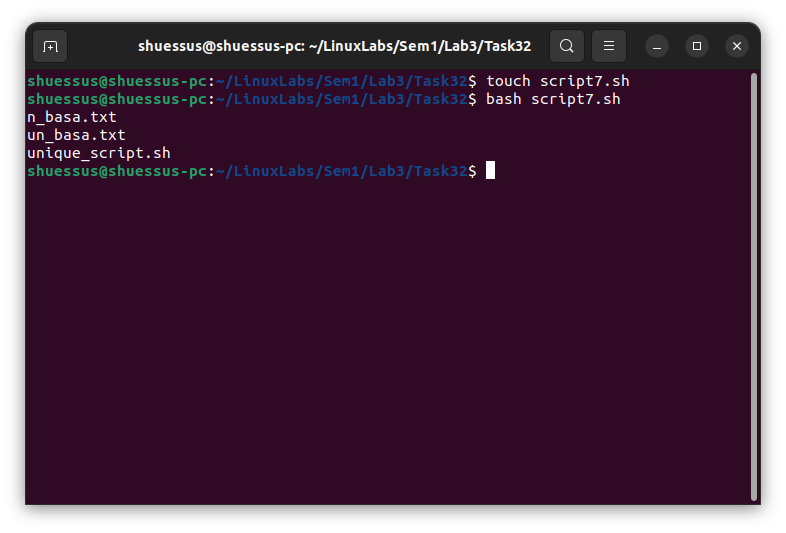


Рис. 24. Результат

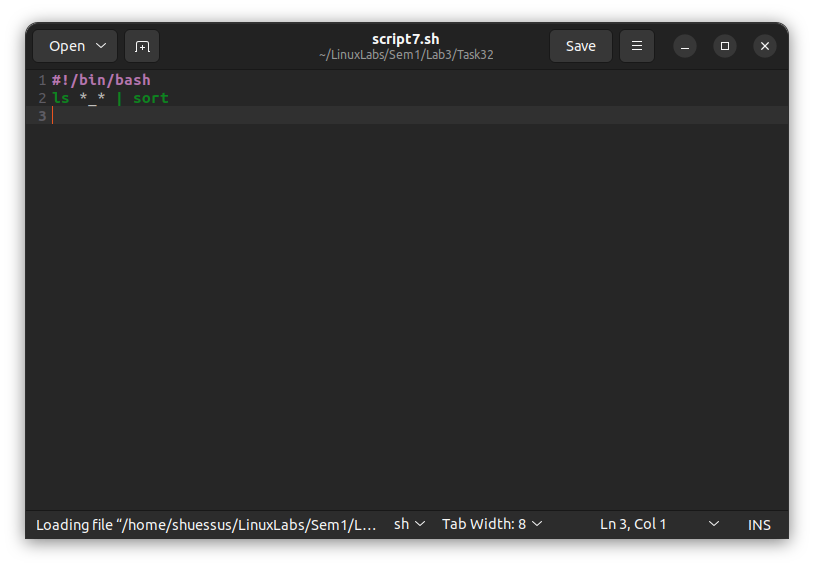


Рис. 25. Скрипт

**Задание 3**

15. Напишите скрипт запуска другого скрипта.

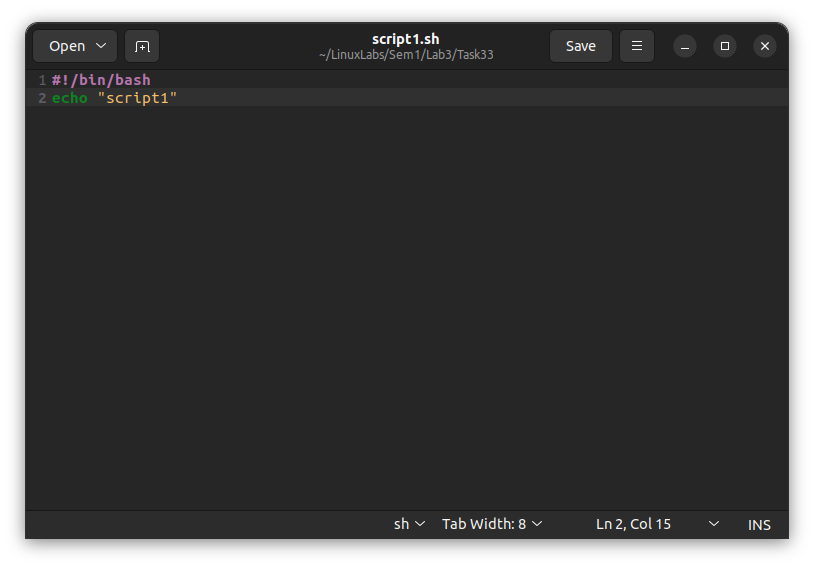


Рис. 15. Первый скрипт



Рис. 16. Второй скрипт

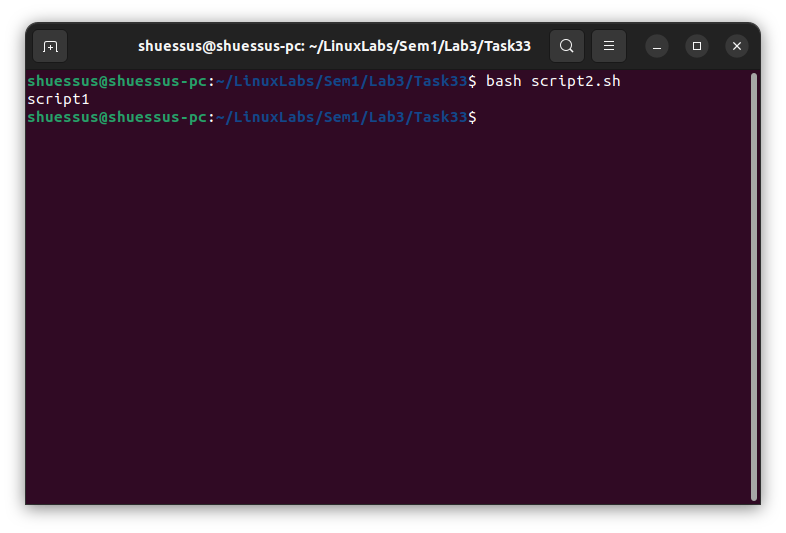


Рис. 17. Результат выполнения скрипта

16. Расположите скрипт в одной из стантадртных директорий системы, и

запустите его из домашней директории пользователя.

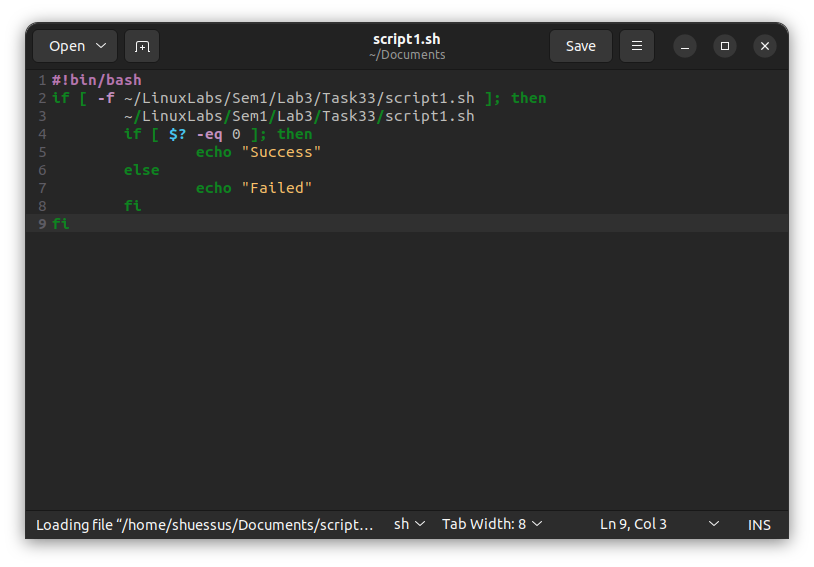


Рис. 18. Скрипт

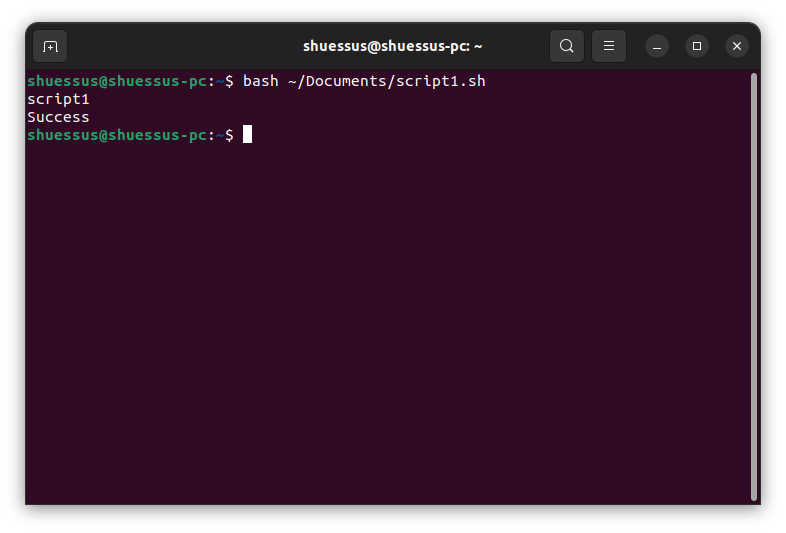


Рис. 19. Запуск скрипта из домашней директории

**Контрольные вопросы**

**Какие стандартные каталоги ситемы Linux вы знаете, и для скрипты и**

**программ каких пользователей эти каталоги предназначены?**

В Linux существует несколько стандартных каталогов системы, которые имеют определенное предназначение для скриптов и программ. Ниже приведены некоторые основные каталоги и их назначение:

1. /bin - содержит исполняемые файлы (программы), доступные всем пользователям, включая основные утилиты командной строки, такие как ls, cp, cat и т.д.

2. /sbin - подобно каталогу /bin, содержит исполняемые файлы, но предназначенные для административных задач и требуют прав суперпользователя (root) для выполнения. Например, команды ifconfig и iptables.

3. /usr/bin - содержит бинарные файлы программ, доступные всем пользователям, но не являющиеся основными утилитами операционной системы. Здесь обычно располагаются приложения, установленные с помощью процедуры установки пакетов.

4. /usr/sbin - аналогично /usr/bin, но содержит исполняемые файлы, предназначенные для административных задач и требующие прав суперпользователя.

5. /usr/local/bin - содержит исполняемые файлы программ, установленных локально пользователем или администратором системы, в отличие от файлов, установленных с помощью процедуры установки пакетов.

6. /usr/local/sbin - также аналогично /usr/local/bin, но содержит исполняемые файлы для административных задач и требует прав суперпользователя.

7. /home - каталоги, в которых находятся персональные файлы и настройки для каждого пользователя системы. Каждый пользователь имеет свой собственный каталог внутри /home.

8. /var - содержит изменяемые файлы, такие как логи системы, базы данных и временные файлы. Например, /var/log содержит системные журналы, а /var/tmp - временные файлы.

Это лишь некоторые из основных стандартных каталогов системы Linux, их назначение может отличаться в зависимости от конкретного дистрибутива и конфигурации системы.

**Что такое перенаправление ввода/вывода и потоки данных?**

В Linux перенаправление ввода/вывода - это механизм, который позволяет изменить источник или назначение данных во время выполнения команды или программы. Он позволяет управлять потоками данных между файлами, устройствами и процессами.

В Linux существуют три основных потока данных:

1. Стандартный ввод (stdin) - это поток ввода данных, который обычно связан с клавиатурой или другим устройством ввода. Обозначается символом '<'.

2. Стандартный вывод (stdout) - это поток вывода данных, который обычно связан с терминалом или другим устройством вывода. Если команда или программа не указывает иное, вывод будет направляться в стандартный вывод. Обозначается символом '>'.

3. Стандартный поток ошибок (stderr) - это поток вывода сообщений об ошибках, который обычно также связан с терминалом или другим устройством вывода. Обозначается символом '2>'.

С помощью перенаправления ввода/вывода можно изменить направление потоков данных. Например:

- Команда "command < file.txt" перенаправляет содержимое файла "file.txt" в качестве ввода для команды "command".

- Команда "command > output.txt" перенаправляет вывод команды "command" в файл "output.txt" вместо стандартного вывода.

- Команда "command 2> error.txt" перенаправляет сообщения об ошибках от команды "command" в файл "error.txt" вместо стандартного потока ошибок.

Помимо перенаправления ввода/вывода, в Linux также существуют другие методы работы с потоками данных, такие как создание каналов (pipes) для передачи данных между процессами, использование фонового режима выполнения (background) для продолжения работы в терминале после запуска команды и многое другое. Все это позволяет эффективно управлять потоками данных в Linux системах.