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 10702221

Колмак Е.А.

Принял**:** Давыденко Н.В.

Минск, 2023

# Лабораторная работа № 3. Командный язык и скрипты Shell.

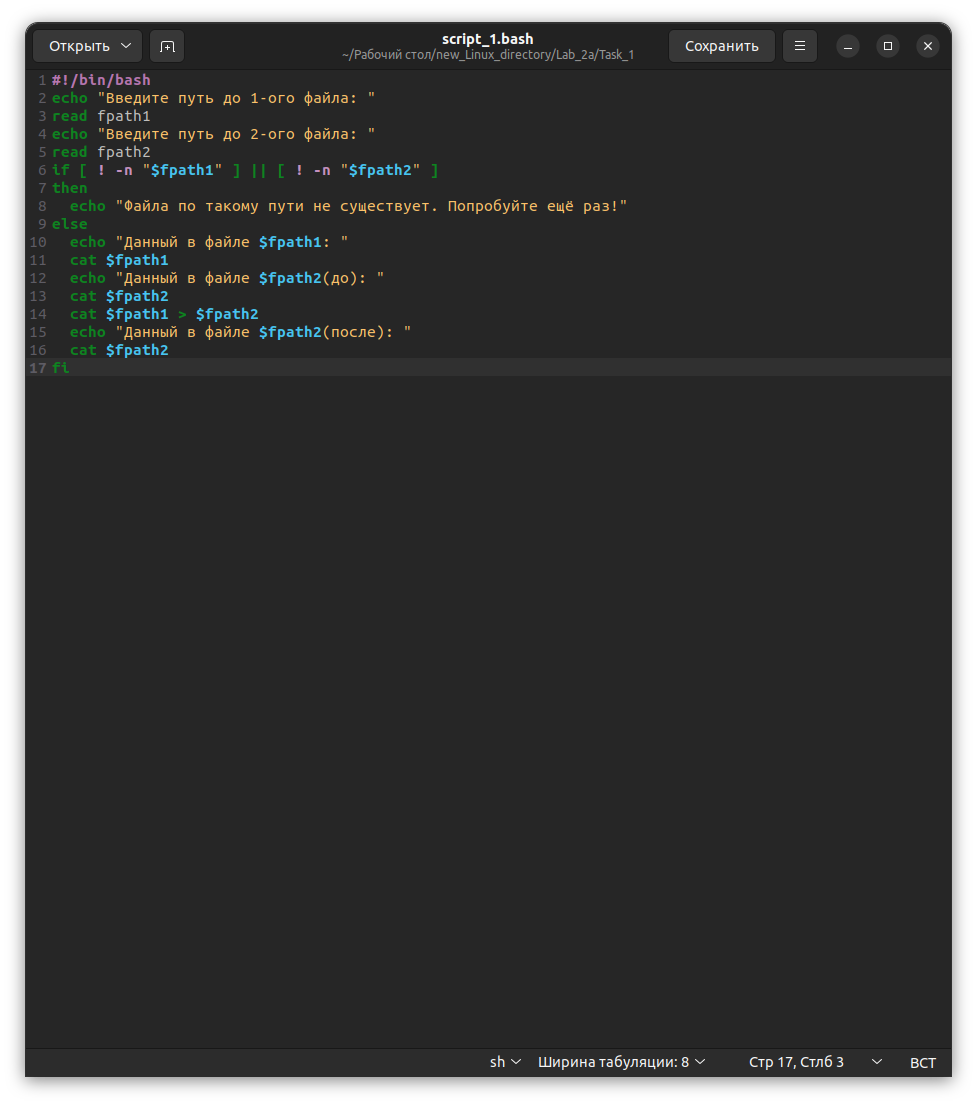
***Задание 1:***

1. Написать скрипт, который записывает содержимое одного файла в

другой и при этом оба файла существуют до операции. Дать два разных

варианта выполнения.

**Решение:**

*Рис. 1. Решение 1-го задания*

***Скрипт:***

*#!/bin/bash*

*echo "Введите путь до 1-ого файла: "*

*read fpath1*

*echo "Введите путь до 2-ого файла: "*

*read fpath2*

*if [ ! -n "$fpath1" ] || [ ! -n "$fpath2" ]*

*then*

*echo "Файла по такому пути не существует. Попробуйте ещё раз!"*

*else*

*echo "Данный в файле $fpath1: "*

*cat $fpath1*

*echo "Данный в файле $fpath2(до): "*

*cat $fpath2*

*cat $fpath1 > $fpath2*

*echo "Данный в файле $fpath2(после): "*

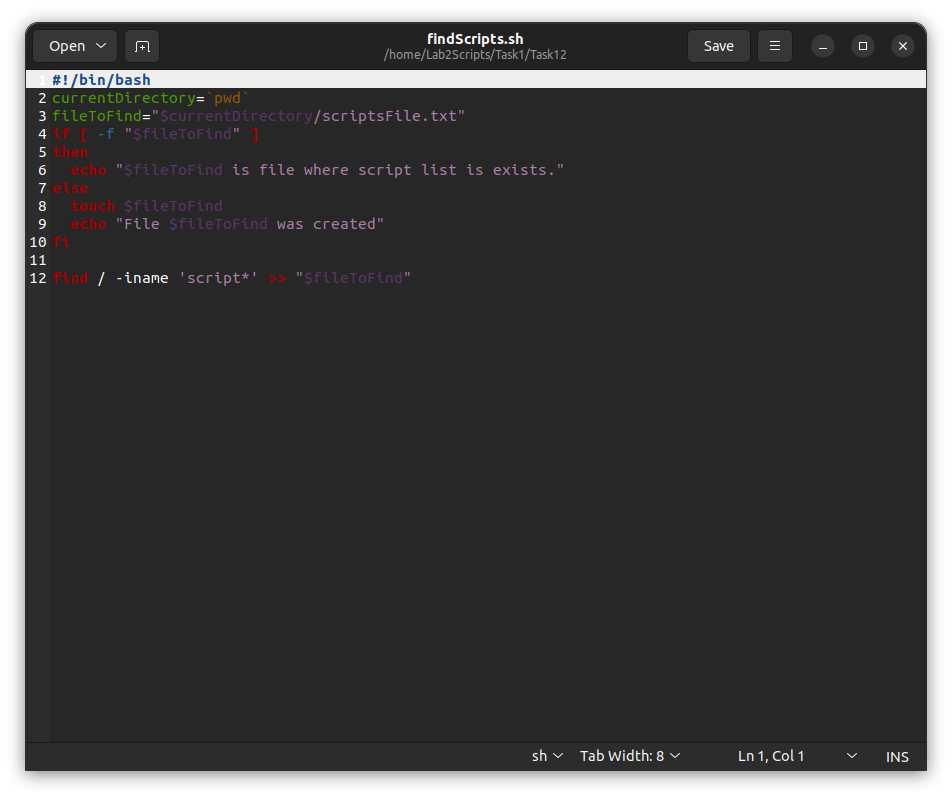
*cat $fpath2*

*fi*

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

файл list.txt.

**Решение:**

*Рис. 2. Решение 2-го подзадания 1-го задания*

***Скрипт:***

*#!/bin/bash*

*currentDirectory=`pwd`*

*fileToFind="$currentDirectory/scriptsFile.txt"*

*if [ -f "$fileToFind" ]*

*then*

*echo "$fileToFind is file where script list is exists."*

*else*

*touch $fileToFind*

*echo "File $fileToFind was created"*

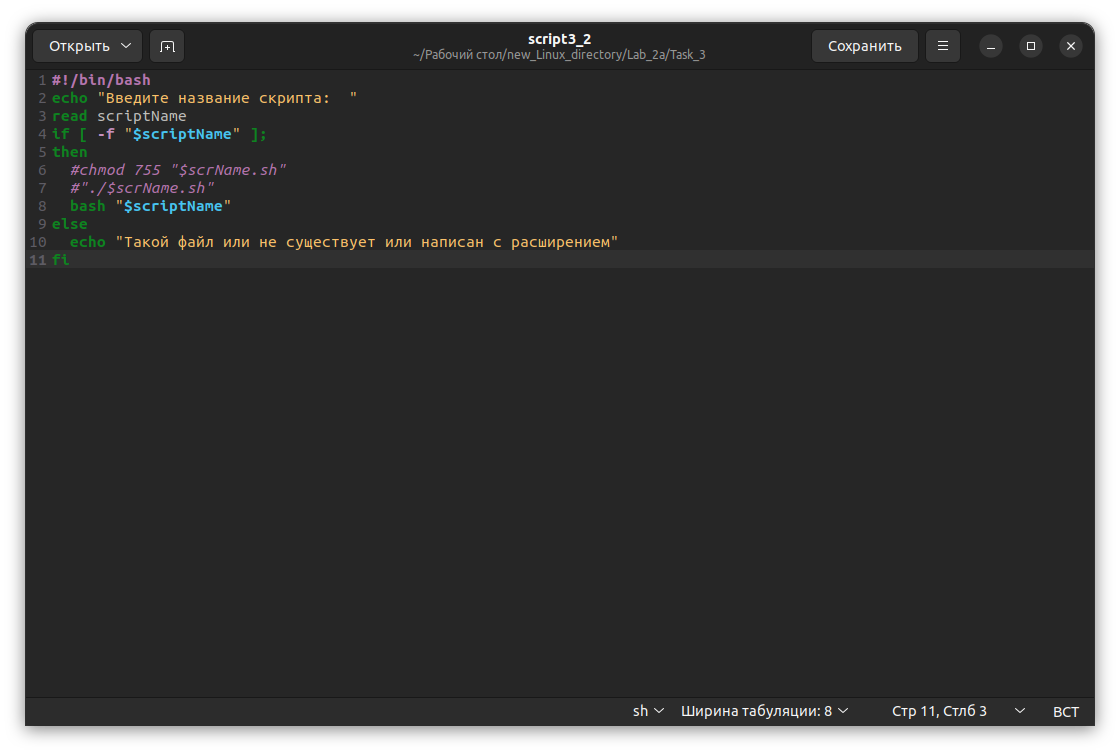
*fi*

*find / -iname 'script\*' >> "$fileToFind"*

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

директории, если да, то выполнить его.

**Решение:**

*Рис. 3. Решение 3-го подзадания 1-го задания*

***Скрипт:***

*#!/bin/bash*

*echo "Введите название скрипта: "*

*read scriptName*

*if [ -f "$scriptName" ];*

*then*

*#chmod 755 "$scrName.sh"*

*#"./$scrName.sh"*

*bash "$scriptName"*

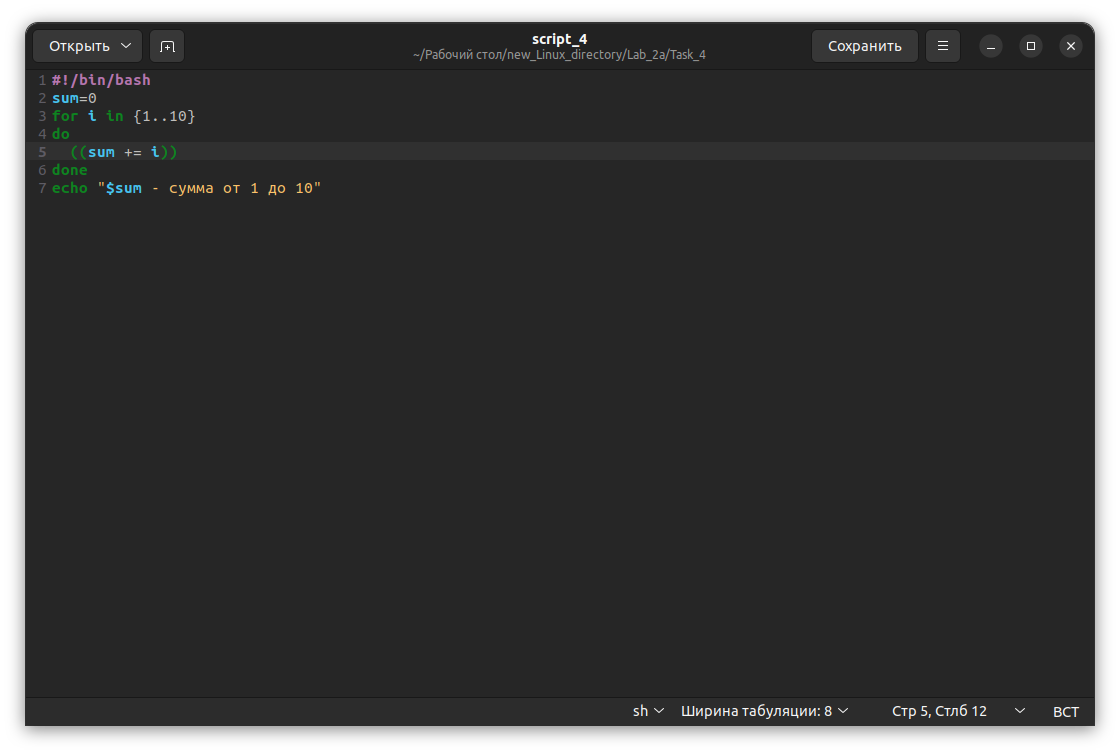
*else*

*echo "Такой файл или не существует или написан с расширением"*

*fi*

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

**Решение:**

*Рис. 4. Решение 4-го подзадания 1-го задания*

***Скрипт:***

*#!/bin/bash*

*sum=0*

*for i in {1..10}*

*do*

*((sum += i))*

*done*

*echo "$sum - сумма от 1 до 10"*

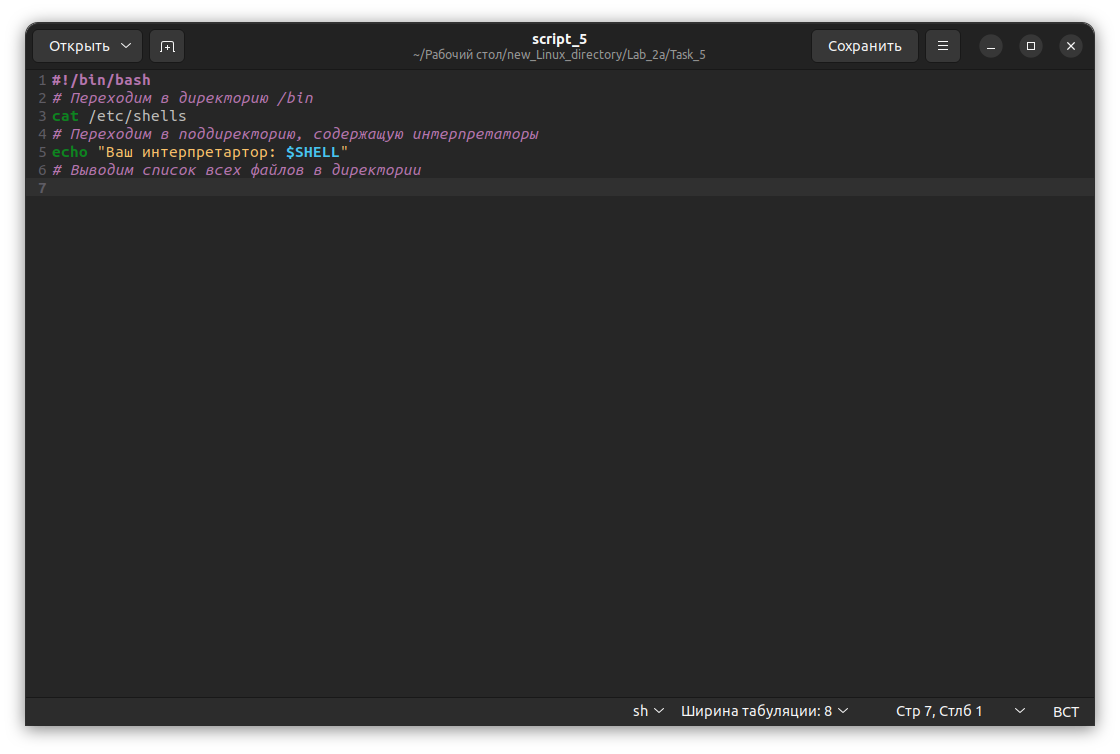
5. А какие командные интерпретаторы установлены в вашей системе?

Напишите скрипт поиска местонахождения командных интерпретато-

ров вашей системы. Также определите какой интерпретатор использу-

ется сейчас в вашей системе?

**Решение:**

*Рис. 5. Решение 5-го подзадания 1-го задания*

***Скрипт:***

*#!/bin/bash*

*# Переходим в директорию /bin*

*cat /etc/shells*

*# Переходим в поддиректорию, содержащую интерпретаторы*

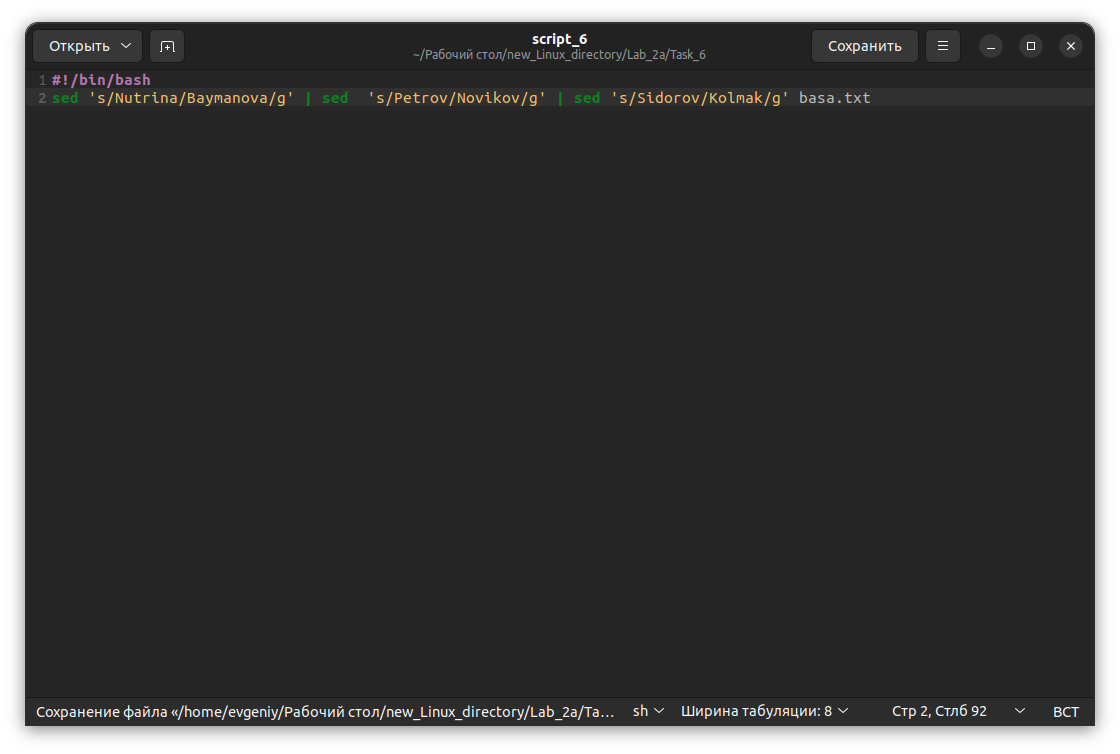
*echo "Ваш интерпретартор: $SHELL"*

*# Выводим список всех файлов в директории*

***Задание 2:***

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

**Решение:**

*Рис. 1. Решение 1-го подзадания 2-го задания*

***Скрипт:***

*#!/bin/bash*

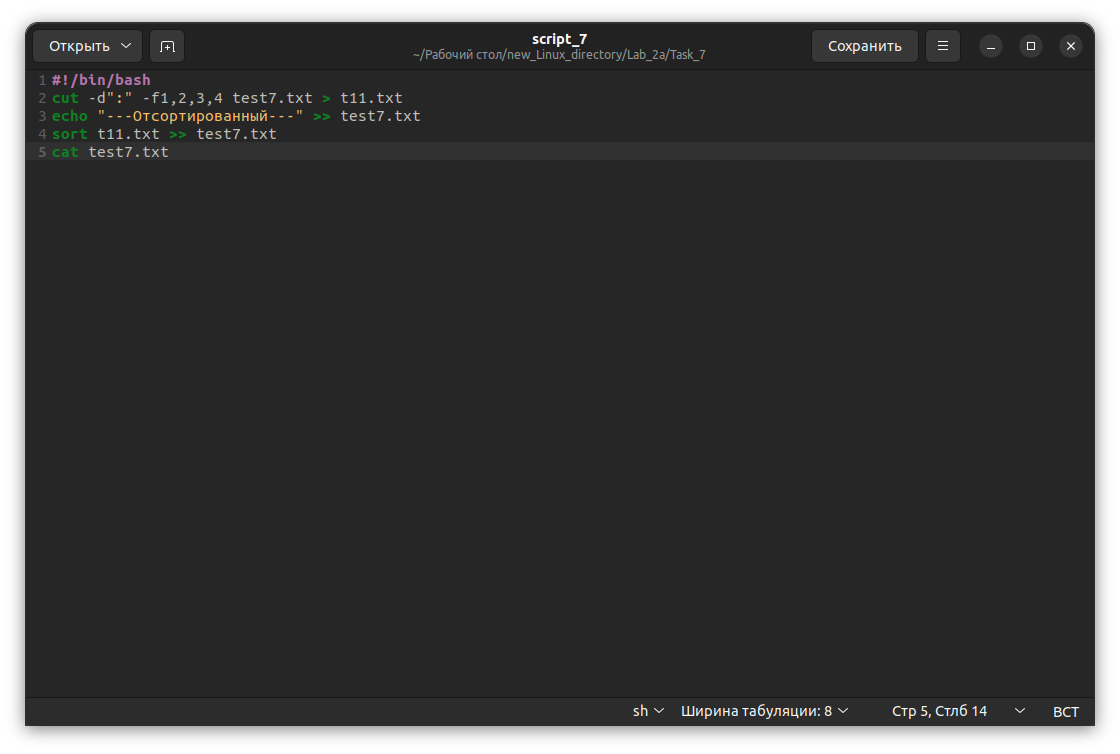
*sed 's/Nutrina/Baymanova/g' | sed 's/Petrov/Novikov/g' | sed 's/Sidorov/Kolmak/g' basa.txt*

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).

**Решение:**

*Рис. 2. Решение 2-го подзадания 2-го задания*

***Скрипт:***

*#!/bin/bash*

*cut -d":" -f1,2,3,4 test7.txt > t11.txt*

*echo "---Отсортированный---" >> test7.txt*

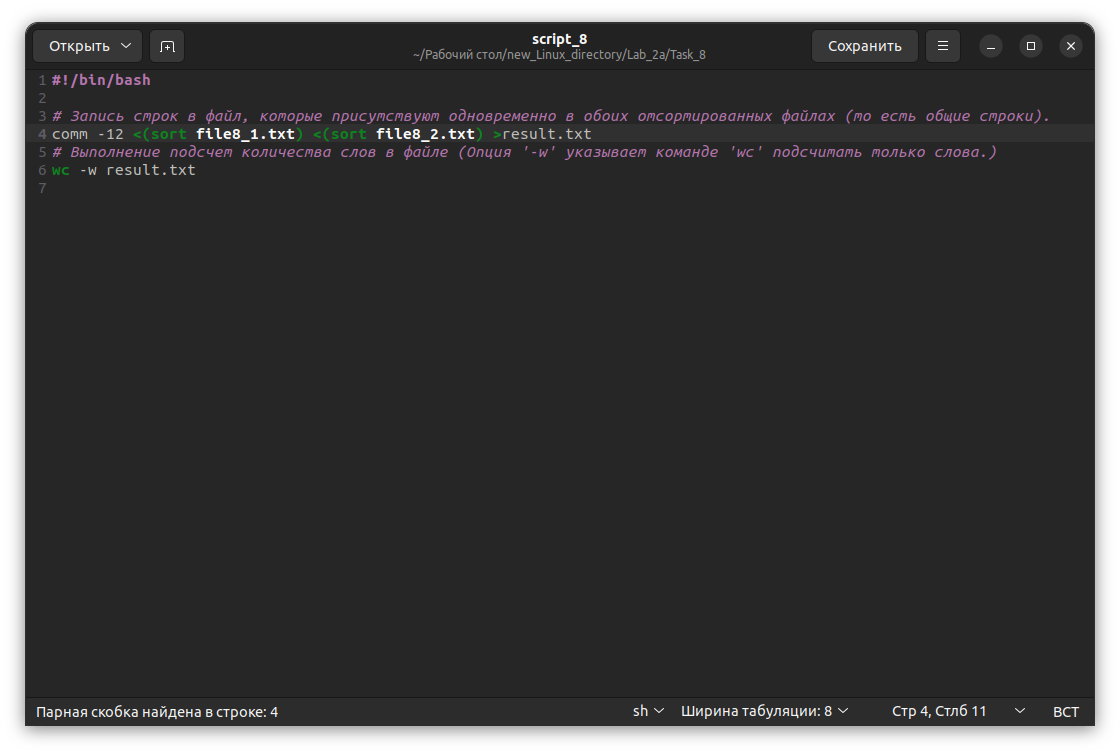
*sort t11.txt >> test7.txt*

*cat test7.txt*

8. Подсчитайте число одинаковых слов в обоих файлах. Для подсчета

числа слов в файле используйте команду: wc -w file1.

**Решение:**



*Рис. 3. Решение 3-го подзадания 2-го задания*

***Скрипт:***

*#!/bin/bash*

*# Запись строк в файл, которые присутствуют одновременно в обоих отсортированных файлах (то есть общие строки).*

*comm -12 <(sort file8\_1.txt) <(sort file8\_2.txt) >result.txt*

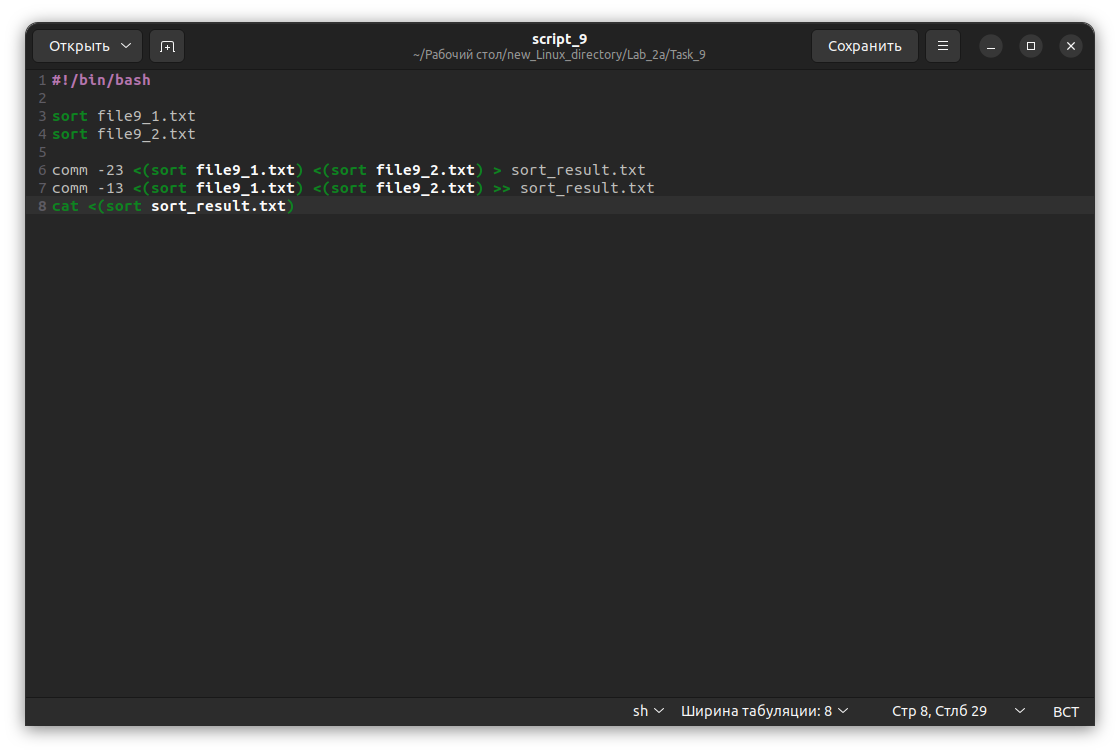
*# Выполнение подсчет количества слов в файле (Опция '-w' указывает команде 'wc' подсчитать только слова.)*

*wc -w result.txt*

9. Подсчитайте число одинаковых слов в обоих файлах. Для подсчета

числа слов в файле используйте команду: wc -w file1.

**Решение:**

****

*Рис. 4. Решение 4-го подзадания 2-го задания*

***Скрипт:***

*#!/bin/bash*

*sort file9\_1.txt*

*sort file9\_2.txt*

*comm -23 <(sort file9\_1.txt) <(sort file9\_2.txt) > sort\_result.txt*

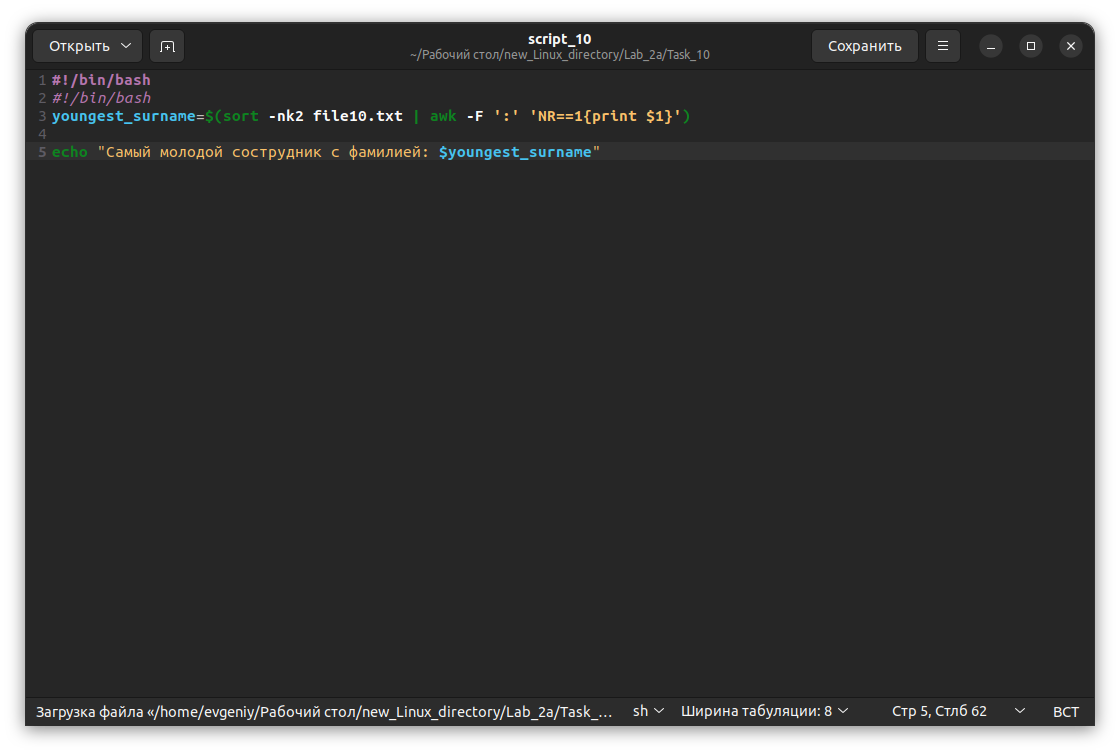
*comm -13 <(sort file9\_1.txt) <(sort file9\_2.txt) >> sort\_result.txt*

*cat <(sort sort\_result.txt)*

10. Написать скрипт, который создает отсортированный файл, содержащий

слова из двух файлов, исключая их общую часть одинаковых слов.

**Решение:**

****

*Рис. 5. Решение 5-го подзадания 2-го задания*

***Скрипт:***

*#!/bin/bash*

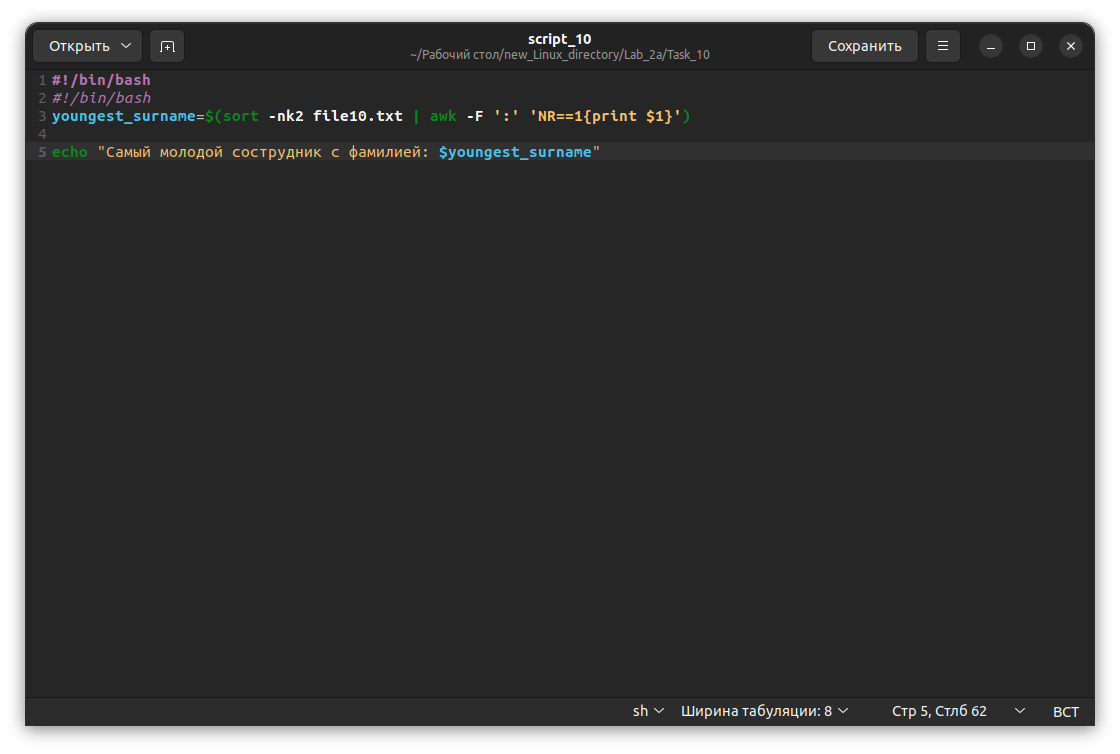
*#!/bin/bash*

*youngest\_surname=$(sort -nk2 file10.txt | awk -F ':' 'NR==1{print $1}')*

*echo "Самый молодой сострудник с фамилией: $youngest\_surname"*

11. Вывести фамилию самого молодого человека (в файле basa.txt – это вто- рой столбец).

**Решение:**

****

*Рис. 6. Решение 6-го подзадания 2-го задания*

***Скрипт:***

*#!/bin/bash*

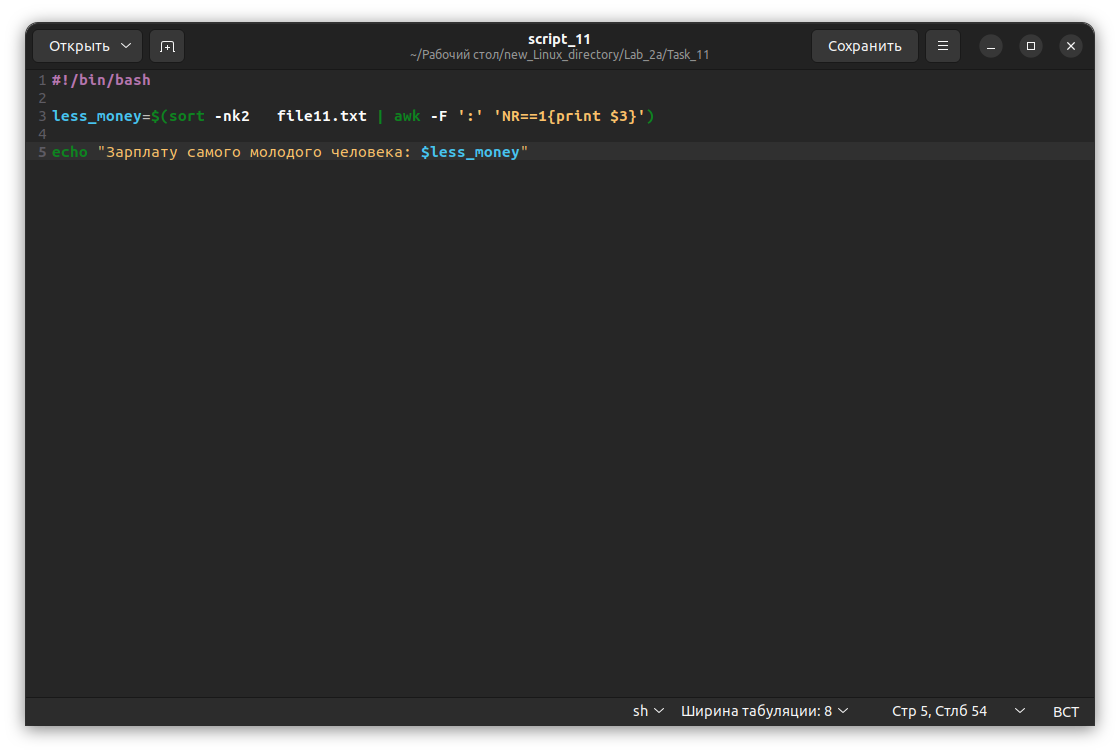
*youngest\_surname=$(sort -nk2 file10.txt | awk -F ':' 'NR==1{print $1}')*

*echo "Самый молодой сострудник с фамилией: $youngest\_surname"*

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

бец).

**Решение:**

****

*Рис. 12. Решение 7-го подзадания 2-го задания*

***Скрипт:***

*#!/bin/bash*

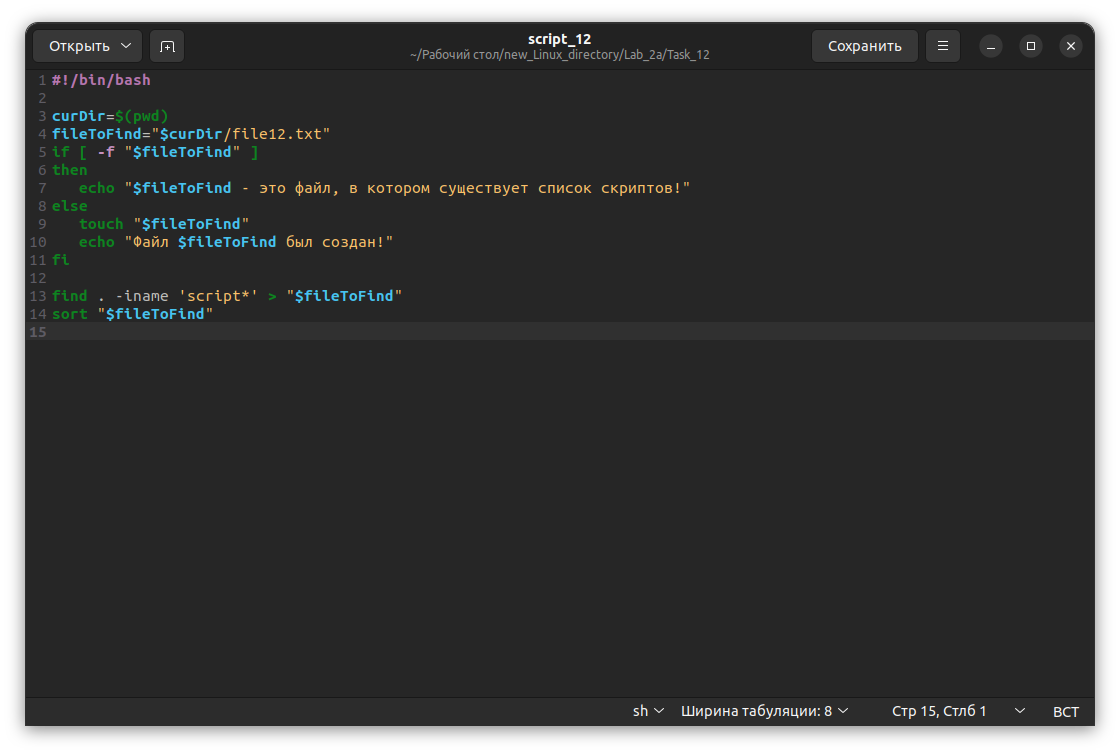
*less\_money=$(sort -nk2 file11.txt | awk -F ':' 'NR==1{print $3}')*

*echo "Зарплату самого молодого человека: $less\_money"*

13. Вывести отсортированный список имен файлов, начальная часть имени

есть script.

**Решение:**



*Рис. 8. Решение 8-го подзадания 2-го задания*

***Скрипт:***

*#!/bin/bash*

*curDir=$(pwd)*

*fileToFind="$curDir/file12.txt"*

*if [ -f "$fileToFind" ]*

*then*

*echo "$fileToFind - это файл, в котором существует список скриптов!"*

*else*

*touch "$fileToFind"*

*echo "Файл $fileToFind был создан!"*

*fi*

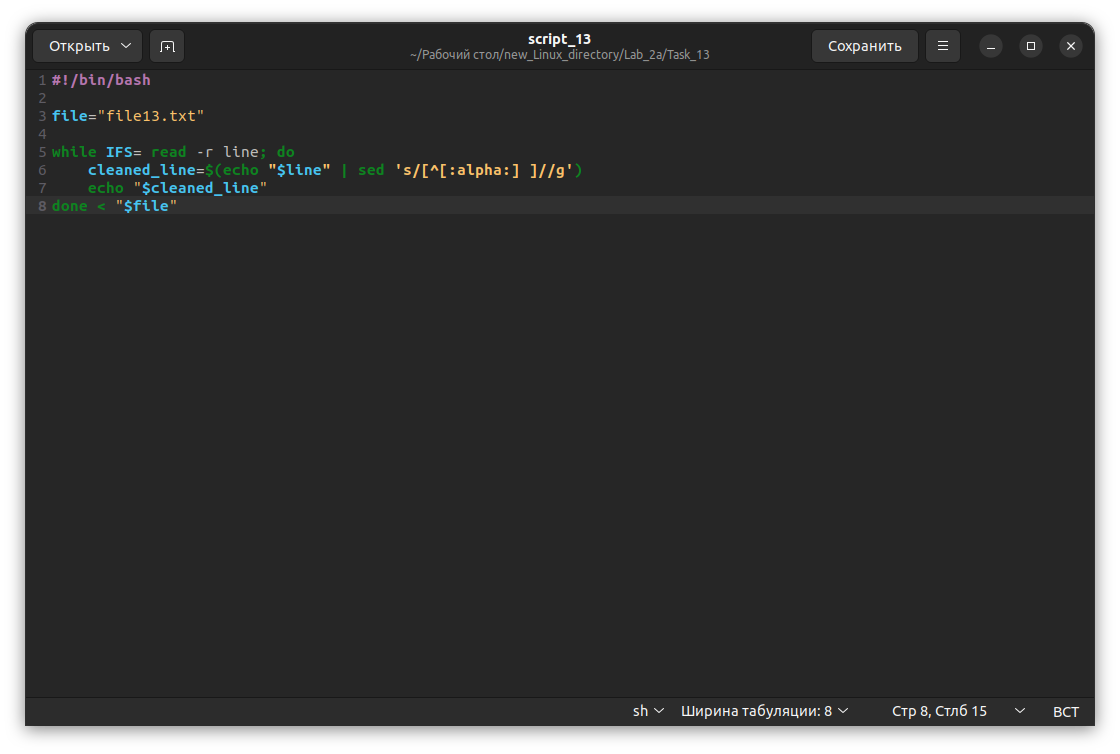
*find . -iname 'script\*' > "$fileToFind"*

*sort "$fileToFind"*

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

буквенные символы, а остальные символы выбрасывает.

**Решение:**

****

*Рис. 9. Решение 9-го подзадания 2-го задания*

***Скрипт:***

*#!/bin/bash*

*file="file13.txt"*

*while IFS= read -r line; do*

*cleaned\_line=$(echo "$line" | sed 's/[^[:alpha:] ]//g')*

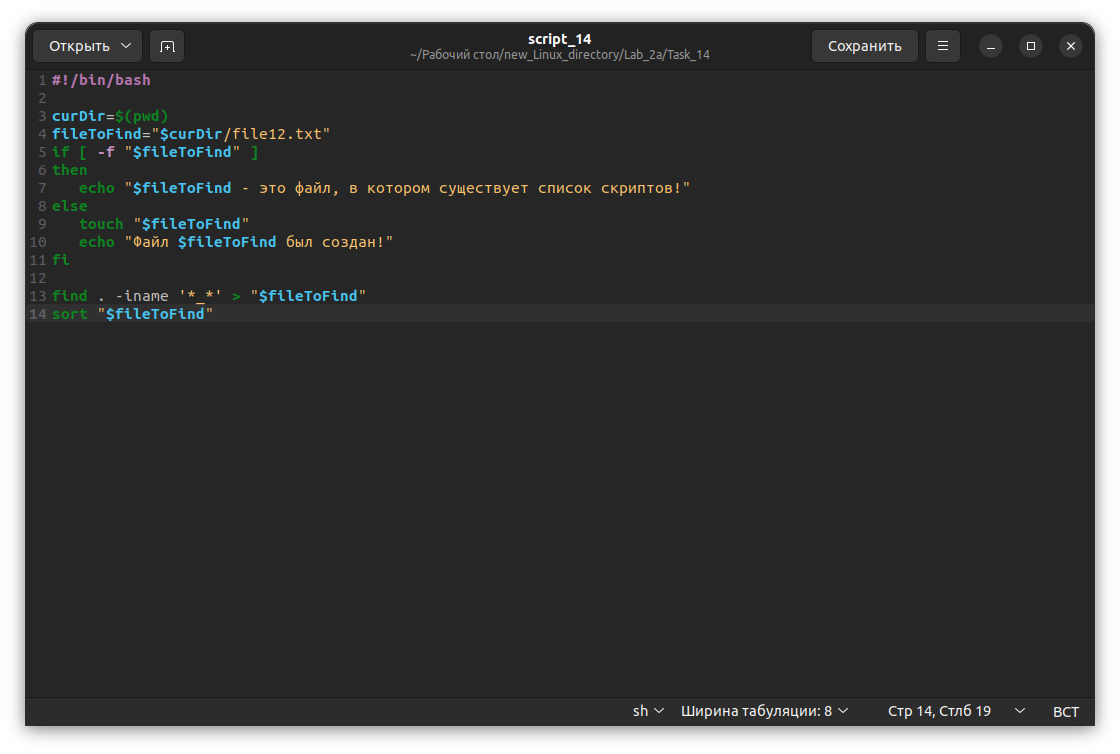
*echo "$cleaned\_line"*

*done < "$file"*

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

жится символ подчеркивания, например, script\_sort.

**Решение:**

****

*Рис. 15. Решение 15-го подзадания 1-го задания*

***Скрипт 1:***

*#!/bin/bash*

*curDir=$(pwd)*

*fileToFind="$curDir/file12.txt"*

*if [ -f "$fileToFind" ]*

*then*

*echo "$fileToFind - это файл, в котором существует список скриптов!"*

*else*

*touch "$fileToFind"*

*echo "Файл $fileToFind был создан!"*

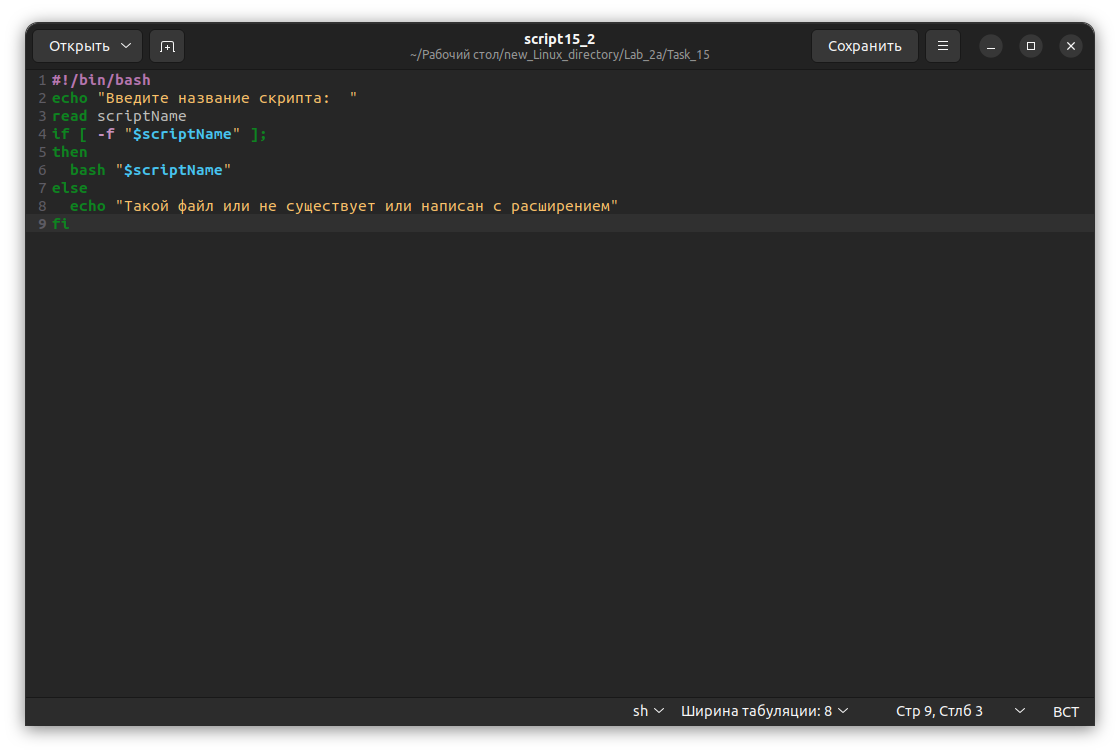
*fi*

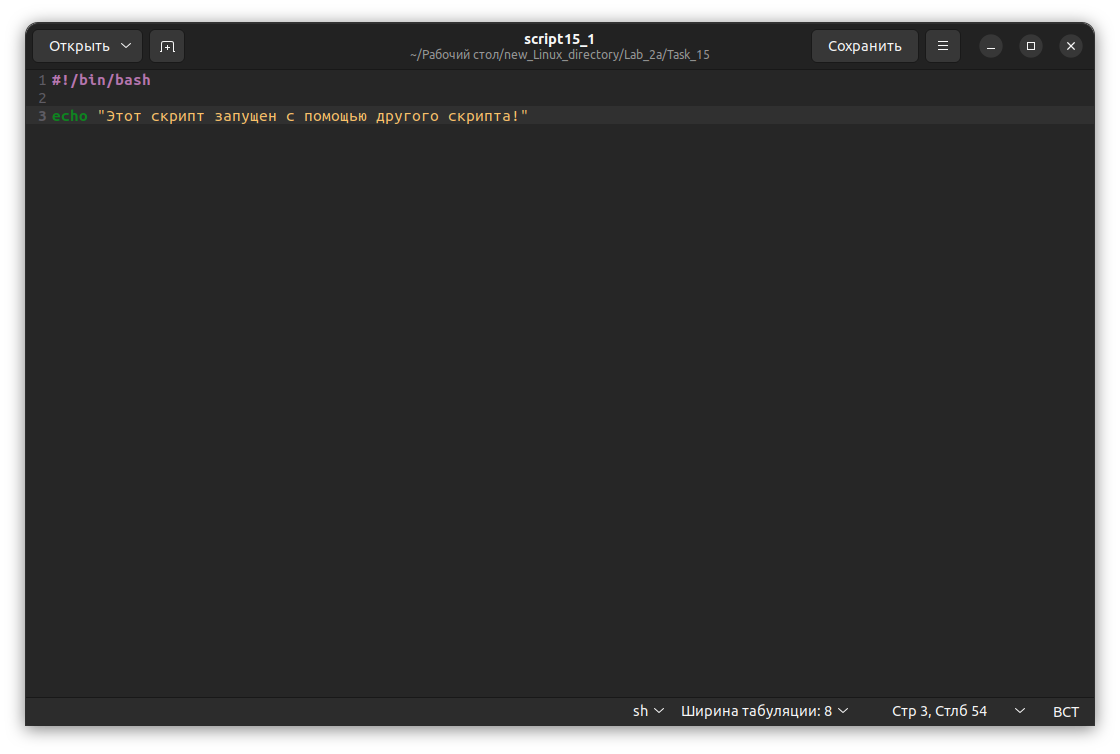
*find . -iname '\*\_\*' > "$fileToFind"*

*sort "$fileToFind"*

***Задание 3:***

16. Напишите скрипт запуска другого скрипта.

****



*Рис. 1. Решение 1-го подзадания 3-го задания*

***Скрипт 1:***

*#!/bin/bash*

*echo "Введите название скрипта: "*

*read scriptName*

*if [ -f "$scriptName" ];*

*then*

*bash "$scriptName"*

*else*

*echo "Такой файл или не существует или написан с расширением"*

*fi*

***Скрипт 2:***

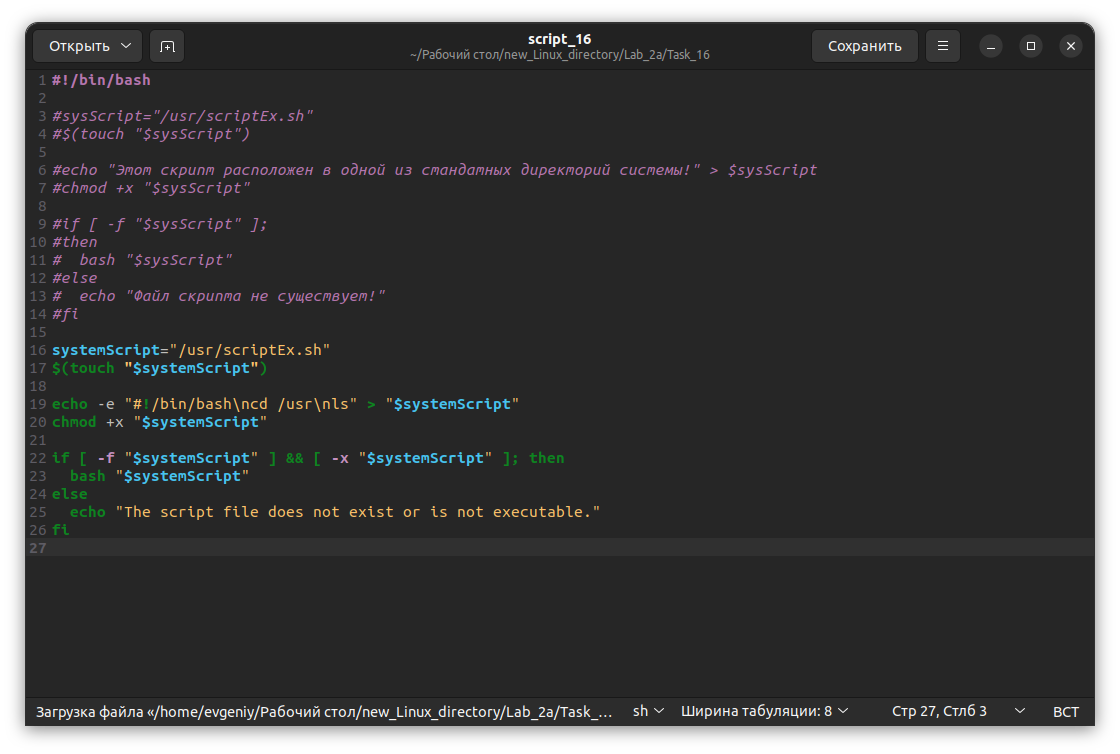
*#!/bin/bash*

*echo "Этот скрипт запущен с помощью другого скрипта!"*

***Задание 3:***

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

запустите его из домашней директории пользователя.

****

*Рис. 2. Решение 2-го подзадания 3-го задания*

***Скрипт:***

#!/bin/bash

#sysScript="/usr/scriptEx.sh"

#$(touch "$sysScript")

#echo "Этот скрипт расположен в одной из стандатных директорий системы!" > $sysScript

#chmod +x "$sysScript"

#if [ -f "$sysScript" ];

#then

# bash "$sysScript"

#else

# echo "Файл скрипта не существует!"

#fi

systemScript="/usr/scriptEx.sh"

$(touch "$systemScript")

echo -e "#!/bin/bash\ncd /usr\nls" > "$systemScript"

chmod +x "$systemScript"

if [ -f "$systemScript" ] && [ -x "$systemScript" ]; then

bash "$systemScript"

else

echo "The script file does not exist or is not executable."

fi

Вывод*:* в ходе выполнения лабораторной работы были изучены принципы написания и особенности создания bash-скриптов, а также изучены синтаксические и семантические особенности кода, интерпретируемого башем.

**Контрольные вопросы:**

1. Какие способы запуска выполнения скриптов вы знаете? Как запустить скрипт на который отсутствуют права исполнения (запуска)?

*Существуют несколько способов запуска выполнения скриптов в Ubuntu. Основные способы включают:*

*1. Через команду bash : Вы можете запустить скрипт, указав его имя после команды bash в терминале. Например, bash script.sh . Этот способ не требует наличия прав исполнения у скрипта.*

*2. Через команду sh : Аналогично, вы можете использовать команду sh для запуска скрипта, например, sh script.sh . Она также не требует наличия прав исполнения у скрипта.*

*3. Добавление прав исполнения: Если у вас нет прав исполнения для скрипта, вы можете добавить их с помощью команды chmod . Например, chmod +x script.sh добавит права исполнения для скрипта. Затем вы сможете запустить его, указав его имя после команды ./ в терминале, например, ./script.sh .*

*Помните, что при использовании команд bash или sh для запуска скриптов, они должны находиться в текущем рабочем каталоге или указываться с полным путем к файлу. Если скрипт находится в другом каталоге, вам нужно указать полный путь к нему при запуске командой bash или sh .*

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

*Три фундаментальные концепции операционных систем и системного программирования:*

*1. Управление процессами. Управление процессами включает в себя создание, выполнение и завершение процессов. Оно включает в себя распределение ресурсов, планирование процессов и управление межпроцессным взаимодействием.*

*2. Управление памятью. Управление памятью отвечает за распределение и управление ресурсами памяти системы. Оно включает в себя отслеживание использования памяти, выделение памяти процессам и обеспечение эффективного использования памяти.*

*3. Файловая система. Файловая система обеспечивает структурированный способ хранения, организации и извлечения данных на устройстве хранения. Она управляет файлами, каталогами и разрешениями на доступ к файлам, позволяя пользователям и приложениям взаимодействовать с сохраненными данными.*

*Эти концепции составляют основу операционных систем и системного программирования, обеспечивая эффективное выполнение программ и управление системными ресурсами****.***