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* Яковец И.А

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

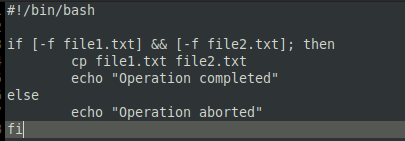
**1.** Написать скрипт, который записывает содержимое одного файла в другой и при этом оба файла существуют до операции. Дать два разных варианта выполнения.

1. Вызываем текстовый редактор



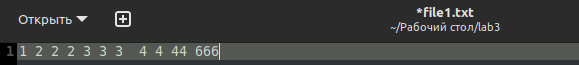
1. Пишем код сценария и сохраняем файл

1) Используя команду `cp` для копирования файла:



1. Делаем скрипт выполняемым файлом

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



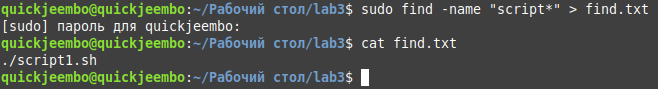
5. Запускаем скрипт



6. Проверяем наш второй файл:

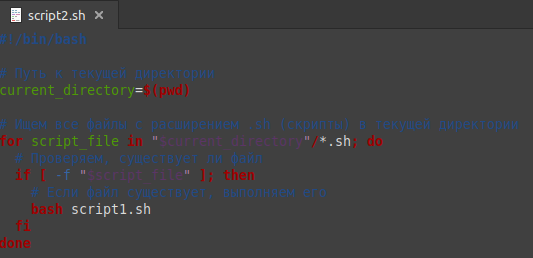


**2.** Найти все файлы, начинающиеся на слово script и записать их имена в файл list.txt.



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

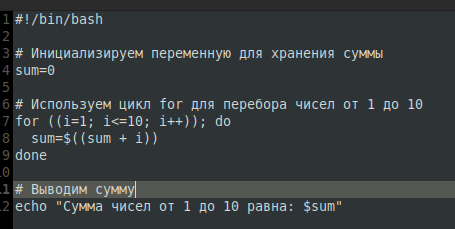
1. Пишем скрипт.



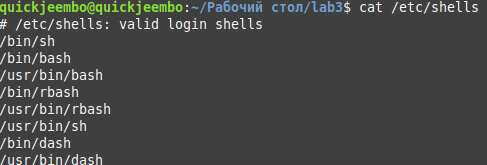
2. Запускаем и проверяем работу



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

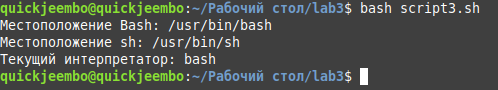
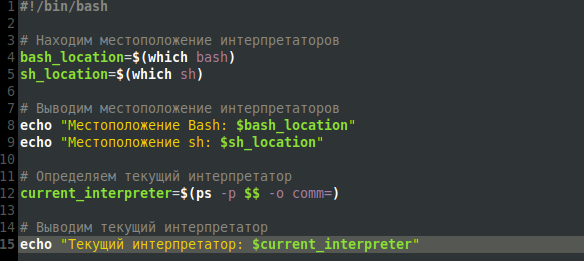


**5** А какие командные интерпретаторы установлены в вашей системе?



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.

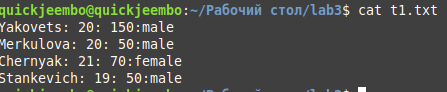
Должны получить такой файл:

Privet year

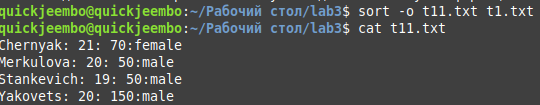
Good bye year

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

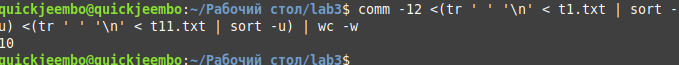
**6.** Используйте текст первого примера, заменить некоторые фамилии на фамилии членов вашей команды (например, Nutrina заменить на Stankewich).



**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).

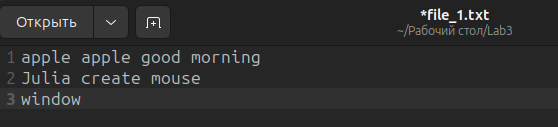


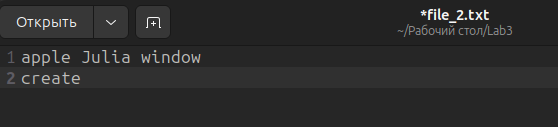
**8.** Подсчитайте число одинаковых слов в обоих файлах. Для подсчета числа слов в файле используйте команду: wc -w file1.



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

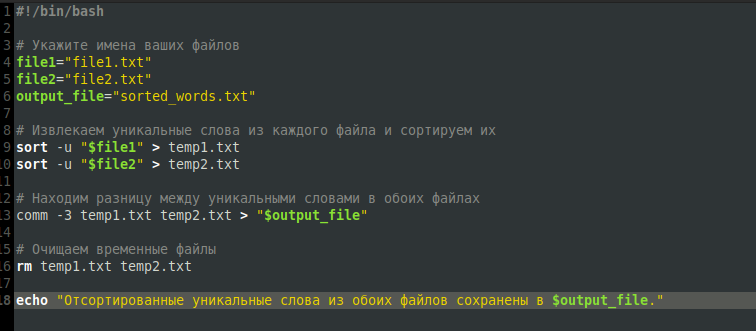
1. Создаем два файла и заполняем их.



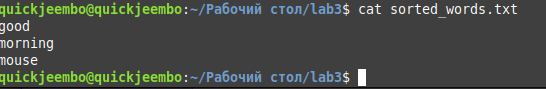


2. Пишем скрипт.

Здесь мы при помощи tr -s разделяем заменяем пробелы, но 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.



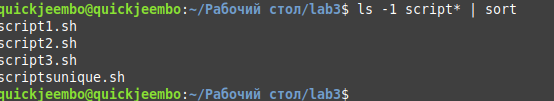
**10.** Вывести фамилию самого молодого человека



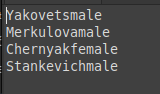
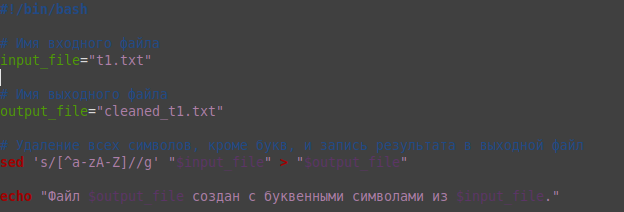
**11.** Вывести зарплату самого молодого человека (зарплата – третий столбец).



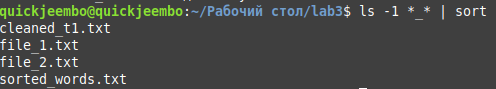
**12.** Вывести отсортированный список имен файлов, начальная часть имени есть script.



**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.

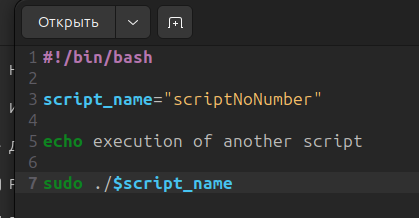


**Задание 3.**

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.

**Какие стандартные каталоги ситемы Linux вы знаете, и для скрипты и программ каких пользователей эти каталоги предназначены?**

В системе Linux существует множество стандартных каталогов, предназначенных для различных целей и разных категорий пользователей. Вот несколько основных стандартных каталогов:

1. **/bin**: Этот каталог содержит исполняемые файлы (команды), которые нужны для работы системы даже в случае, когда файловая система монтируется только для чтения. Он предназначен для всех пользователей.
2. **/sbin**: Этот каталог также содержит исполняемые файлы, но они обычно предназначены для системных администраторов и нужны для выполнения системных задач.
3. **/usr/bin**: Этот каталог содержит исполняемые файлы, которые предназначены для всех пользователей и используются во время обычной работы.
4. **/usr/sbin**: Этот каталог содержит исполняемые файлы, предназначенные для системных администраторов и используемые для выполнения системных задач.
5. **/usr/local/bin**: Этот каталог содержит исполняемые файлы, установленные на уровне локального пользователя или системного администратора.
6. **/home**: Этот каталог содержит домашние директории пользователей.
7. **/root**: Домашний каталог суперпользователя (администратора системы).
8. **/var**: В этом каталоге хранятся переменные данные, такие как журналы, временные файлы и другие изменяемые данные.
9. **/tmp**: Этот каталог используется для хранения временных файлов.
10. **/etc**: В этом каталоге хранятся конфигурационные файлы системы.
11. **/opt**: Этот каталог предназначен для установки дополнительного программного обеспечения.
12. **/dev**: В этом каталоге представлены файлы, представляющие устройства.
13. **/proc**: Этот каталог содержит виртуальную файловую систему, предоставляющую информацию о работающих процессах и системе.
14. **/mnt**: Этот каталог предназначен для временного монтирования дополнительных файловых систем.
15. **/lib** и **/lib64**: Эти каталоги содержат библиотеки, используемые исполняемыми файлами.
16. **/media**: Этот каталог используется для автоматического монтирования съемных носителей, таких как USB-флешки и CD/DVD.

Каждый из этих каталогов имеет свою специфическую цель и назначение в структуре файловой системы Linux.

**Для чего используются команды-фильтры? Назовите некоторые из них.**

Команды-фильтры в Unix и Linux предназначены для обработки и манипуляции текстовыми данными в конвейере команд или скриптах. Они принимают входные данные, выполняют операции и отправляют результат в выходной поток. Некоторые из популярных команд-фильтров включают **grep** для поиска текста, **sed** для замены и манипуляций, **awk** для сложных операций на текстовых строках, **cut** для извлечения данных, **sort** для сортировки, **uniq** для удаления повторяющихся строк и другие. Фильтры полезны для обработки и анализа текстовых данных в Unix-среде.