МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет информационных технологий и робототехники

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

О Т Ч Е Т

ПО ЛАБОРАТОРНОЙ РАБОТЕ №**2.1**

«*Командный язык и скрипты Shell. Часть 1 – Скрипты и перенаправление ввода/вывода*»

по курсу: "Системное программирование"

Выполнил: Кабыш Я.А. 10702221

Проверил: Давыденко Н

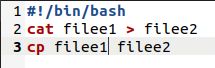
**Цель работы :**

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

**Задание :**

**Задание 1**

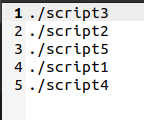
1. Используя скрипт, который записывает содержимое одного файла в другой и при этом оба файла существуют до операции. Дать два разных варианта выполнения.



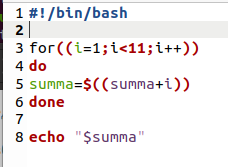


1. Найти все файлы, начинающиеся на слово script и записать их имена в файл list.txt.





1. Написать скрипт, который проверяет, содержится ли скриптовый файл в директории, если да, то выполнить его.
2. Написать скрипт, подсчитывающий сумму от 1 до 10.



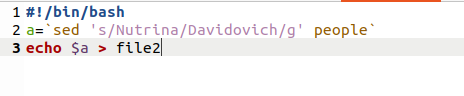


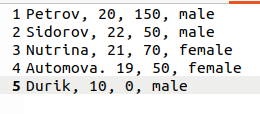
1. А какие командные интерпретаторы установлены в вашей системе? Напишите скрипт поиска местонахождения командных интерпретаторов вашей системы.

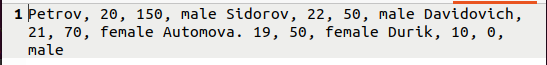


**Задание 2**

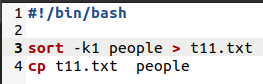
1. Используйте текст первого примера, заменить некоторые фамилии на фамилии членов вашей команды.

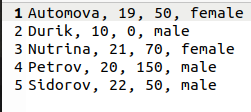






1. Отсортируйте данные файла так, чтобы он не поменял свое собственное имя (Решение. Нужно предварительно создать промежуточный файл t11.txt. Затем выполнить нужные команды).



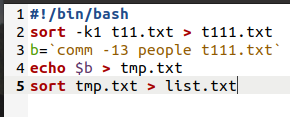


1. Подсчитайте число одинаковых слов в обоих файлов. Для подсчета числа слов в файле используйте команду: wc -w file1.



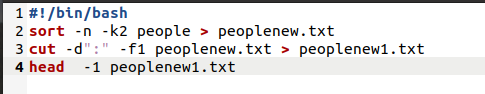


1. Написать скрипт, который создает отсортированный файл, содержащий слова из двух файлов, исключая их общую часть одинаковых слов.



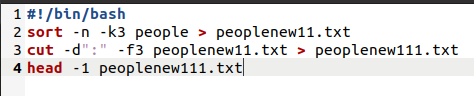
1. Вывести фамилию самого молодого человека (в файле people – это второй столбец).



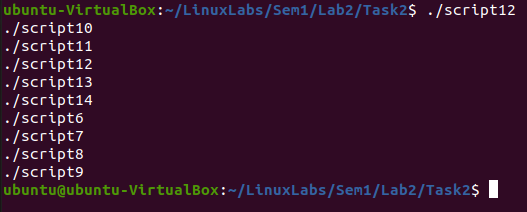


1. Вывести зарплату самого молодого человека (зарплата – третий столбец).



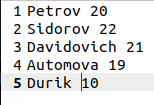


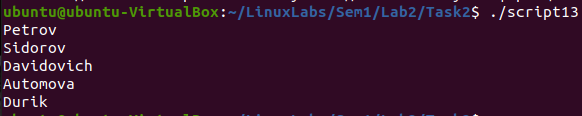
1. Вывести отсортированный список имен файлов, начальная часть имени есть script.





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





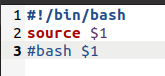


1. Вывести упорядоченный список имен файлов, в именах которых содержится символ подчеркивания, например, script\_sort.



**Задание 3**

1. Напишите скрипт запуска другого скрипта.



1. Расположите скрипт в одной из стандартных директорий системы, и запустите его из домашней директории пользователя.

**Вывод :** в ходе выполнения лабораторной работы я закрепила на практике основы работы со сценариями командной оболочки и перенаправление ввода/вывода.

**Контрольные вопросы:**

Файлы скриптов (или исполняемые файлы) должны иметь соответствующие права доступа, чтобы быть выполненными в системе. Вот общие рекомендации по правам доступа для скриптов:

1. **Исполняемый бит (Execute):**
   * Для того чтобы скрипт можно было выполнить, он должен иметь установленный исполняемый бит. Это делается с помощью команды **chmod** или другими средствами управления правами доступа.
   * Установка исполняемого бита: **chmod +x имя\_скрипта.sh**
   * Проверка наличия исполняемого бита: **ls -l имя\_скрипта.sh**. В выводе должно быть что-то вроде **-rwxr-xr-x**, где буква **x** указывает на установленный исполняемый бит.
2. **Чтение и запись:**
   * В зависимости от ваших потребностей, скрипт может также требовать прав на чтение и запись файлов, с которыми он взаимодействует.
   * Права на чтение и запись могут быть установлены с помощью команды **chmod**, например: **chmod +rw имя\_файла**.
3. **Права доступа к директории:**
   * Если скрипт выполняется в директории, важно также убедиться, что у вас есть соответствующие права доступа к этой директории, чтобы можно было выполнять скрипт из нее.
   * Для выполнения скрипта из директории, убедитесь, что у вас есть права на чтение и выполнение этой директории.

**Потоки данных:**

* Потоки данных - это механизм передачи информации между программами и операционной системой или между разными частями программы. Они представляют собой абстракцию для ввода и вывода данных.
* Стандартные потоки данных:
  + **stdin (стандартный ввод)**: Этот поток данных предназначен для ввода данных в программу. Обычно он связан с клавиатурой.
  + **stdout (стандартный вывод)**: Этот поток данных предназначен для вывода данных, произведенных программой. Обычно он связан с экраном (терминалом).
  + **stderr (стандартный вывод ошибок)**: Этот поток данных предназначен для вывода сообщений об ошибках и диагностических сообщений программой. Он также связан с экраном.
* Файлы потоков данных находятся в системной директории **/dev/**, например:
  + **/dev/stdin** - стандартный ввод (stdin)
  + **/dev/stdout** - стандартный вывод (stdout)
  + **/dev/stderr** - стандартный вывод ошибок (stderr)

Стандартные потоки данных могут быть перенаправлены, чтобы изменить их направление. Например, вы можете перенаправить вывод программы в файл, используя символы **>** или **>>**, или перенаправить ввод из файла с помощью **<**.