МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования

«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Интеллектуальные информационные технологии»

ОТЧЕТ

По лабораторной работе №3

«BASH: ПОТОКИ ДАННЫХ. ПРОГРАММИРОВАНИЕ»

**Выполнил:** Антонюк Н.А.

**Номер зачетной книжки 220203**

**Группа:** ПО-11

**Проверил:** Давидюк Ю.И.

Брест 2023

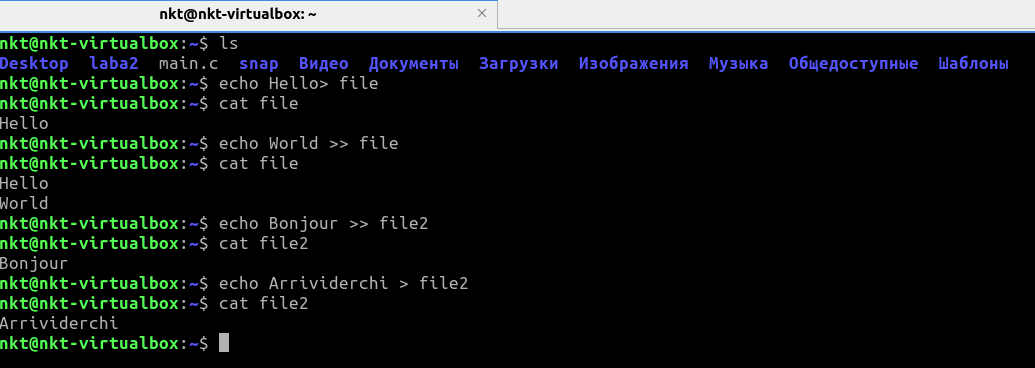
**№1.** Вывести любое сообщение с помощью команды echo перенаправив вывод: - в несуществующий файл с помощью символа >; - в несуществующий файл с помощью символа >>; - в существующий файл с помощью символа >; - в существующий файл с помощью символа >>; Объяснить результаты.

С помощью операции переадресации создается новый файл-адресат. Если он уже

существует, то система заменяет его содержимое данными стандартного вывода.

Для добавления стандартного вывода к существующему файлу служит оператор

переадресации >>.



- В несуществующий файл с помощью символа >:

echo Hello > file

Результат: Создастся новый файл file и в него будет записано сообщение “Hello”.

- В несуществующий файл с помощью символа >>:

echo Bonjour >> file

Результат: Создастся новый файл file и в него будет дописано сообщение “Bonjour”.

- В существующий файл с помощью символа >:

echo Arrividerchi >> file2

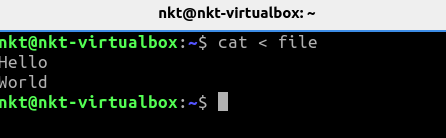
Результат: Содержимое файла file будет заменено на новое сообщение “Arrividerchi”.

- В существующий файл с помощью символа >>:

echo World > file

Результат: Сообщение “World” будет добавлено в конец файла file.

**№2**. Переадресовать стандартный ввод для команды cat на файл.



**№3**

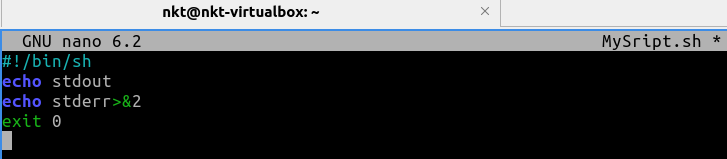
Вывести сообщение с помощью команды echo в канал ошибок. Создать файл myscript: #!/bin/sh

echo stdout

echo stderr>&2

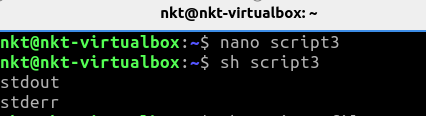
exit 0

Запустить его: - без перенаправления (sh myscript); - перенаправив стандартный вывод в файл, просмотреть содержимое файла (sh myscript > file1); - перенаправить стандартный канал ошибок в существующий и несуществующий файлы с помощью символов > и >> ; - перенаправив стандартный вывод в файл 1, стандартный канал ошибок - в файл 2; - перенаправив стандартный вывод и стандартный канал ошибок в файл 3; - перенаправив стандартный вывод в файл 4 с помощью символа >, а стандартный канал ошибок в файл 4 с помощью символа >>; Объяснить результаты.



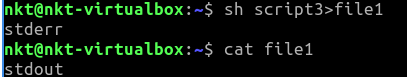
Запустить его:

− без перенаправления (sh script3);



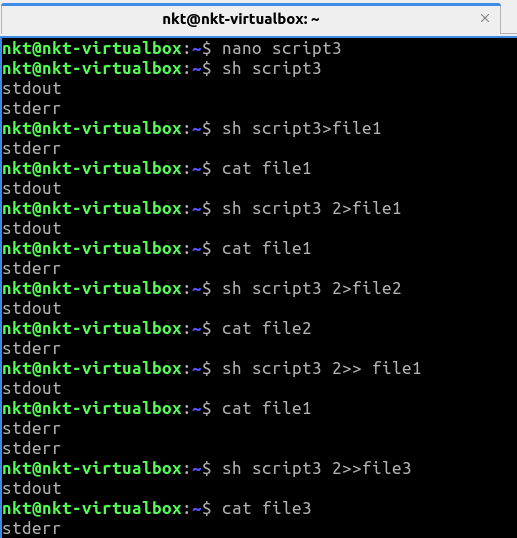
− перенаправив стандартный вывод в файл, просмотреть содержимое файла

(sh script3 > file1);

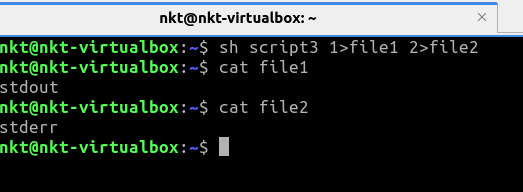


− перенаправив стандартный канал ошибок в существующий и

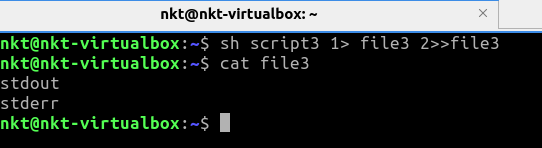
несуществующий файлы с помощью символов > и >>;



− перенаправив стандартный вывод в файл 1, стандартный канал ошибок — в файл2;

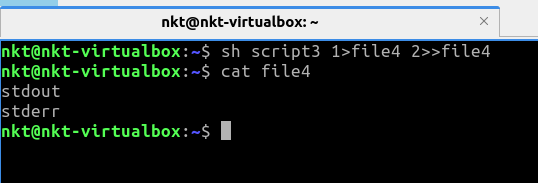


− перенаправив стандартный вывод и стандартный канал ошибок в файл 3;



− перенаправив стандартный вывод в файл 4 с помощью символа >, а

стандартный канал ошибок в файл 4 с помощью символа >>.



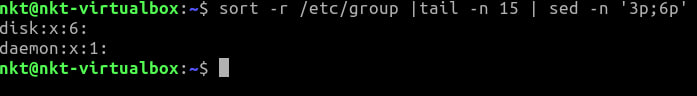
При перенаправлении стандартного вывода в файл выводится строка «stdout», а

при перенаправлении стандартного потока ошибок — строка «stderr».

**№4** Вывести третью и шестую строку из последних пятнадцати строк отсортированного в обратном порядке файла /etc/group.

sort -r /etc/group | tail -n 15 | sed -n '3p;6p'

sott -r сортирует в обратном порядке. tail -n 15 /etc/group, чтобы получить последние пятнадцать строк файла /etc/group. Используется sed -n '3p;6p', чтобы выбрать третью и шестую строку для вывода.



**№5** Подсчитать при помощи конвейера команд количество блочных и количество символьных устройств ввода-вывода, доступных в системе.

Подсчитать при помощи конвейера команд количество блочных и количество символьных устройств ввода-вывода, доступных в системе.

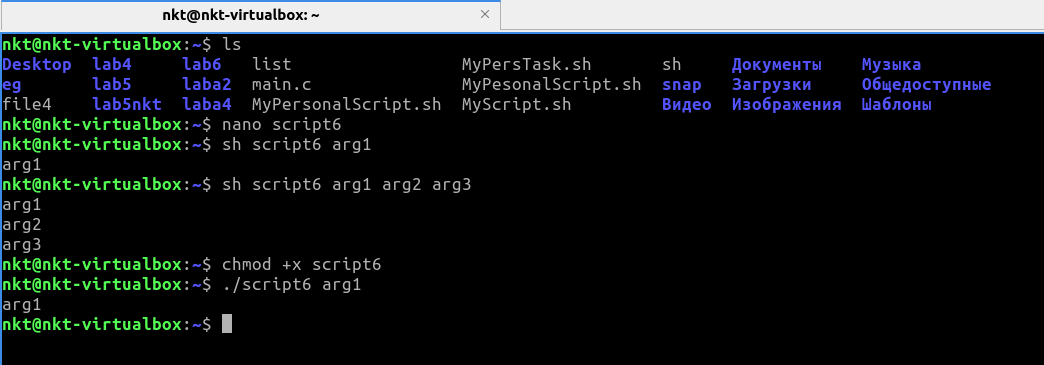
ls -l /dev | grep “^b” | wc -l (блочных устройств)

ls -l /dev | grep “^c” | wc -l (символьных устройств)



**№6** Написать скрипт, выводящий на консоль все аргументы командной строки, переданные данному скрипту. Привести различные варианты запуска данного скрипта, в том числе без непосредственного вызова интерпретатора в командной строке.

1. bash script.sh arg1 arg2 arg3
2. sh script.sh arg1 arg2 arg3
3. script.sh arg1 arg2 arg3



**Скрипт:**

#!/bin/bash

for arg in "$@"

do

echo $arg

done

**№7** Написать скрипт согласно индивидуальному заданию. Номер варианта согласовать с преподавателем.

**Вариант 19**. Реализовать командный файл, реализующий символьное меню (в цикле):

1) поиск всех файлов в заданном каталоге, у которых нет прав на

редактирование;

2) поиск файлов в заданном каталоге по заданным правам;

3) завершение.

**Скрипт:**

#!/bin/bash

while :

do

# Выводим меню

echo "Символьное меню:"

echo "1) Поиск файлов без прав на редактирование в заданном каталоге"

echo "2) Поиск файлов в заданном каталоге по заданным правам"

echo "3) Завершение"

# Запрос выбора пользователя

echo -n "Выберите действие (1-3): "

read item

case $item in

1)

# Поиск файлов без прав на редактирование

echo -n "Введите путь к каталогу: "

read directory

find "$directory" -maxdepth 1 -type f ! -writable

;;

2)

# Поиск файлов по заданным правам

echo -n "Введите путь к каталогу: "

read directory

echo -n "Введите права для поиска (например, 644): "

read permissions

find "$directory" -maxdepth 1 -type f -perm "$permissions"

;;

3)

# Завершение программы

echo "Программа завершена."

exit 0

;;

\*)

# Обработка некорректного выбора

echo "Некорректный ввод. Пожалуйста, выберите действие от 1 до 3."

;;

esac

# Запрос на продолжение выполнения программы

echo -n "Хотите продолжить? (y/n): "

read continue\_execution

if [ $continue\_execution != y ]

then

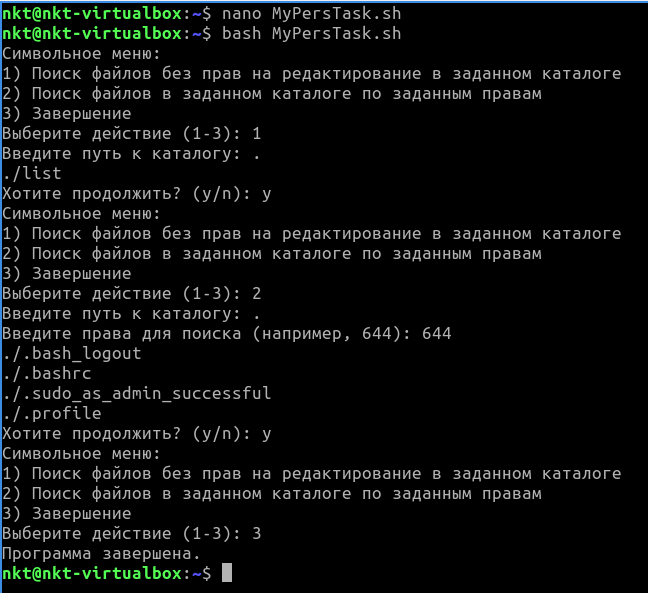
echo "Программа завершена."

exit 0

fi

done

**Вывод в консоли:**



Для работы с 1 заданием в скрипте

Предварительно создал каталог list с помощью “touch”

Присвоил ему права с помощью команды “chmod”