МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

КАФЕДРА ИИТ

ОТЧЁТ

по лабораторной работе №3

**«BASH: Потоки данных.**

**Программирование»**

Выполнил:

Студент 2 курса

группы ПО-9

Мисиюк Алексей Сергеевич

(№ зач. книги 210664)

Проверила:

Давидюк Ю. И.

Брест 2022

**Цель работы:** ознакомиться иизучить потоки данных (в частности ввода, вывода) в терминале bash, изучить и опробовать программирование с помощью командных файлов.

**Ход работы**

**Вариант индивидуального задания №18.**

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.  
    ~$ echo hi >1t.txt

~$ echo hi >>2t.txt

~$ echo hi >1t.txt

~$ echo hi >>2t.txt

~$ cat 1t.txt

hi

~$ cat 2t.txt

hi

hi

Перенаправление в несуществующий файл создает файл. > перезаписывает файл, >> дописывает в файл, поэтому в 2-ом файле 2 строки, а в 1-ом одна.

1. Переадресовать стандартный ввод для команды cat на файл.  
    ~$ cat <2t.txt

hi

hi

1. 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.  
    ~$ echo HIerror >&2

HIerror

~$ cat >myscript

#!/bin/sh

echo stdout

echo stderr>&2

exit 0

^C

~$ sh myscript

stdout

stderr

//Поток вывода и поток ошибок выводятся на экран

~$ sh myscript >file1

stderr

~$ cat file1

stdout

//Поток ошибок выводится на экран, поток вывода в файле

~$ sh myscript 2>file1

stdout

~$ sh myscript 2>>file2

stdout

~$ sh myscript 2>file1

stdout

~$ sh myscript 2>>file2

stdout

~$ cat file1

stderr

~$ cat file2

stderr

stderr

//Перенаправление в несуществующий файл создает файл. > перезаписывает //файл, >> дописывает в файл, поэтому в 2-ом файле 2 строки, а в 1-ом одна.

~$ sh myscript >file1 2>file2

~$ cat file1

stdout

~$ cat file2

stderr

//Потоки направлены в 2 разных файла

~$ sh myscript >>file3 2>>file3

~$ cat file3

stdout

stderr

//Потоки направлены в 1 файл

~$ sh myscript >file4 2>>file4

~$ cat file4

stdout

stderr

//Потоки направлены в 1 файл, но поток вывода перезаписывает файл, а поток //ошибок дописывает в файл

1. Вывести третью и шестую строку из последних пятнадцати строк отсортированного в обратном порядке файла /etc/group.

**~$ tail -15 /etc/group | sort -r | cat -n | egrep "^\s+[3,6]\s+" | cut -f2**

// ( вспомнив про существование cut, почувствовал себя гением )

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

**~$ ls -l /dev | egrep "^[b,c]" | wc -l**

1. 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.

~$ cat >myscript

**#!/bin/sh**

**echo "$@"**

**exit 0**

^C

~$ chmod +x myscript

~$ sh myscript we are great

we are great

~$ ./myscript we are great

we are great

1. Написать скрипт согласно индивидуальному заданию. Номер варианта согласовать с преподавателем.  
    № 18. Реализовать командный файл, реализующий символьное меню (в цикле):   
    1) копирование файлов: в этом пункте выводится информация о содержимом текущего каталога, далее предлагается интерфейс копирования файла: ввод имени файла и ввод каталога для копирования. По выполнению пункта выводится содержимое каталога, куда был скопирован файл и выводится содержимое скопированного файла;  
    2) завершение.

#!/bin/sh

Resp=1

while [ $Resp != 2 ]

do

echo "Copy - 1, exit - 2:"

read Resp

if [ $Resp = 1 ]

then

echo "File to copy:"

read File

echo "Directory, where copy will be:"

read Directory

cp $File $Directory"/"

ls -l $Directory"/"

cat $Directory"/"$File

fi

done

exit 0

**Вывод:** для работы в терминале bash можно и следует при необходимости использовать потоки данных, их всевозможные перенаправления, а также конвейеры для передачи вывода 1-ой команды сразу же в следующую без очевидного посредника. Кроме этого, bash можно использовать для создания скриптов, что может оказаться удобнее программирования на полноценных языках программирования.