

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

Лабораторная работа №1
по теме «BASH: Потоки данных. Программирование»

Выполнил:
студентка 3 курса
группы АС-51
Усач С.В.
Проверила:
Давидюк Ю.И.

Брест 2020

Задания лабораторной работы:

1. Вывести любое сообщение с помощью команды echo перенаправив вывод :

- в несуществующий файл с помощью символа >;

```
anteli@anteli-VB:~$ echo Hello world > 1
```

```
anteli@anteli-VB:~$ cat 1
```

```
Hello world
```

- в несуществующий файл с помощью символа >>;

```
anteli@anteli-VB:~$ echo Hello world >> 2
```

```
anteli@anteli-VB:~$ cat 2
```

```
Hello world
```

- в существующий файл с помощью символа >; anteli@anteli-

```
VB:~$ echo world is great > 1 anteli@anteli-VB:~$ cat 1
```

```
world is great
```

- в существующий файл с помощью символа >>; anteli@anteli-

```
VB:~$ echo world is great >> 2
```

```
anteli@anteli-VB:~$ cat 2
```

```
Hello world
```

```
world is great
```

По результатам выполнения видим:

Заполнение через «>» заменяет хранимые данные на новые, если папка уже существует. В связи с этим, когда мы заполнили существующий 1 новым значением, то предыдущее было удалено. Способ заполнения через «>>» может просто дополнить существующие данные новыми.

2. Переадресовать стандартный ввод для команды cat на файл.

```
anteli@anteli-VB:~$ cat << end
```

```
> Гm
```

```
> angel
```

```
> with
```

```
> a
```

```
> shotgun
```

```
> end
```

```
Гm angel
```

```
with
```

```
a shotgun
```

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); anteli@anteli-VB:~\$

```
sh myscript
```

```
stdout stderr
-
перенаправ
ив
стандартны
й вывод в
файл,
просмотрет
ь
содержимо
е файла (sh
myscript >
file
1);
anteli@anteli-VB:~$ sh myscript >> 1 stderr
anteli@anteli-VB:~$ cat 1
world is great
stdout
```

-перенаправить стандартный канал ошибок в существующий и не существующий файлы с помощью символов > и >> (а тут и дальше уже сами :)); не существующий:

```
anteli@anteli-VB:~$ sh myscript 2> File_1 stdout
anteli@anteli-VB:~$ cat File_1
stderr
anteli@anteli-VB:~$ sh myscript 2>> File_2 stdout
anteli@anteli-VB:~$ cat File_2
stderr
```

существующий: anteli@anteli-VB:~\$ sh
myscript 2> File_1 stdout
anteli@anteli-VB:~\$ sh myscript 2>> File_2
stdout
anteli@anteli-VB:~\$ cat File_1 stderr
anteli@anteli-VB:~\$ cat File_2
stderr stderr

-перенаправив стандартный вывод в файл 1, стандартный канал ошибок — в файл 2;

```
anteli@anteli-VB:~$ sh myscript > File_1 & sh myscript 2> File_2
[1] 2051
stderr stdout
[1]+  Завершён      sh myscript > File_1 anteli@anteli-VB:~$
cat File_1 & File_2
[1] 2053 stdout
File_2: команда не найдена
[1]+  Завершён      cat File_1
```

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

```
anteli@anteli-VB:~$ sh myscript >> File_3 & sh myscript 2>> File_3
[1] 2085
stderr stdout
[1]+
Завершён
sh myscript
>> File_3
anteli@ante
li-VB:~$
cat File_3
stdout
stderr
```

-перенаправив стандартный вывод в файл 4 с помощью символа >, а стандартный канал ошибок в файл 4 с помощью символа >>;

```
anteli@anteli-VB:~$ sh myscript > File_4 & sh myscript 2>> File_4
[1] 2088
stderr stdout
[1]+ Завершён      sh myscript > File_4 anteli@anteli-VB:~$
cat File_4
stdout
stderr
```

Объяснить результаты.

4. Вывести третью строку из последних десяти строк отсортированного в обратном порядке файла etc/group.

```
anteli@anteli-VB:~$ cat /etc/group | sort -r | tail -10 | head -3 | tail -1 cdrom:x:24:anteli
```

5. Подсчитать при помощи конвейера команд количество блочных устройств ввода вывода, доступных в системе. anteli@anteli-VB:~\$ ls -l /dev | grep '^b' | wc -l
18

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.

Script:

```
#!/bin/sh for
i in "$*" do
echo "$i"
done
exit 0
```

Мы выводим каждый введенный пользователем параметр. Параметры просматриваются через цикл фор для каждого существующего параметра Command line:

```
anteli@anteli-VB:~$ chmod ugo+x myscript anteli@anteli-VB:~$  
./myscript 1 22 333 4444  
1 22 333 4444  
anteli@anteli-VB:~$ bash myscript 1 22 333 4444  
1 22 333 4444  
anteli@anteli-VB:~$ ~/myscript 1 22 333 4444  
1 22 333 4444
```

7. Написать скрипт согласно индивидуальному заданию. Номер варианта согласовать с преподавателем.

Вариант 6. реализовать меню из двух пунктов: 1-й пункт — определить количество запущенных данным пользователем процессов bash(предусмотреть ручной ввод имени пользователя);

2-ой пункт — завершить все процессы bash данного пользователя.

Script:

```
#!/bin/sh echo "Menu" echo "1) Show the user  
bash processes" echo "2) Kill bash processes  
of the such user"  
read -p "Select 1 of 2 variants: " var case  
$var in  
1)read -p "Enter user name: " username  
ps -u "$username"|grep 'bash$';; 2) read  
-p "Enter user name: " username pkill -9  
-u "$username" 'bash$';;  
*) echo "Unknown command! Proshyaite";;  
esac  
exit 0
```

Results: anteli@anteli-VB:~\$

./myscript

Menu

1) Show the user bash processes

2) Kill bash processes of the such user

Select 1 of 2 variants: 1

Enter user name: anteli

1714 pts/0 00:00:00 bash