

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

Лабораторная работа №3  
По дисциплине: «Операционные системы и системное программирование»  
Тема: «BASH: потоки данных. Программирование»

Подготовил:  
Студент 2 курса  
Группы ПО-3(2)  
Огиевич Е.А.  
Проверила:  
Давидюк Ю.И.

Брест, 2020

## Лабораторная работа №3

Цель: изучить потоки данных.

Задание:

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.

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.

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

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

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.

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

Индивидуальное задание:

20. Реализовать командный файл, реализующий символьное меню (в цикле):

1) поиск файлов в заданном каталоге, размер которых не превосходит заданного параметра;

2) поиск файлов в заданном каталоге по заданному времени создания.

3) завершение.

Ход работы:

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.

```
natefoust@natefoust-X550LC:~/OS$ echo text1 > noFile
natefoust@natefoust-X550LC:~/OS$ echo text2 >> noFile1
natefoust@natefoust-X550LC:~/OS$ cat noFile
text1
natefoust@natefoust-X550LC:~/OS$ cat noFile1
text2
natefoust@natefoust-X550LC:~/OS$ echo text3 > noFile
natefoust@natefoust-X550LC:~/OS$ echo text4 >> noFile
natefoust@natefoust-X550LC:~/OS$ cat noFile
text3
text4
```

Все команды сработали безошибочно. Первая команда создала файл noFile и добавила туда text1. Вторая команда тоже создала новый файл noFile и добавила туда text2. С помощью третьей команды мы пересоздали файл noFile, теперь там текст text3. И с помощью четвертый добавили в конец этого файла text4.

2. Переадресовать стандартный ввод для команды cat на файл.

```
natefoust@natefoust-X550LC:~/OS$ cat > OS3.txt
TEST
TEST
^Z
[2]+  Остановлен   cat > OS3.txt
natefoust@natefoust-X550LC:~/OS$ cat < OS3.txt
TEST
TEST
```

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.

1) без перенаправления (sh myscript)

```
natefoust@natefoust-X550LC:~/0S$ touch myscript
natefoust@natefoust-X550LC:~/0S$ sh myscript
stdout
stderr
```

Мы просто выполнили скрипт, не используя перенаправление, поэтому вывело 2 сообщения.

2) перенаправив стандартный вывод в файл, посмотреть содержимое файла (sh myscript > file1)

```
natefoust@natefoust-X550LC:~/0S$ sh myscript > file1
stderr
natefoust@natefoust-X550LC:~/0S$ cat file1
stdout
```

В этом случае мы перенаправили стандартный вывод в файл, поэтому записалось stdout.

3) перенаправить стандартный канал ошибок в существующий и несуществующий файлы с помощью символов > и >>

В существующий:

```
natefoust@natefoust-X550LC:~/0S$ touch myscript
natefoust@natefoust-X550LC:~/0S$ touch file2
natefoust@natefoust-X550LC:~/0S$ sh myscript 2> file2
stdout
natefoust@natefoust-X550LC:~/0S$ cat file2
stderr
natefoust@natefoust-X550LC:~/0S$ sh myscript 2>> file2
stdout
natefoust@natefoust-X550LC:~/0S$ cat file2
stderr
stderr
```

2 раза записалось stderr т. к. в первом случае перенаправили стандартный поток ошибок в файл с помощью символа 2, потом дополнили файл тем же с помощью 2>>.

В несуществующий:

```
natefoust@natefoust-X550LC:~/0S$ sh myscript 2> file3
stdout
natefoust@natefoust-X550LC:~/0S$ sh myscript 2>> file4
stdout
natefoust@natefoust-X550LC:~/0S$ cat file3
stderr
natefoust@natefoust-X550LC:~/0S$ cat file4
stderr
```

Практически такой же результат, только мы перенаправляли в разные файлы. Сработало в обоих случаях т. к. и 2> и 2>> создают новый файл.

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

```
natefoust@natefoust-X550LC:~/OS$ sh myscript 1>file5
stderr
natefoust@natefoust-X550LC:~/OS$ sh myscript 2>file6
stdout
natefoust@natefoust-X550LC:~/OS$ cat file5
stdout
natefoust@natefoust-X550LC:~/OS$ cat file6
stderr
```

Теперь мы с помощью 1> перенаправили вывод в файл, записалось stdout — всё верно. И поток ошибок с помощью 2>. (1> по стандарту отвечает за вывод)

5) перенаправив стандартный вывод и стандартный канал ошибок в файл 3

```
natefoust@natefoust-X550LC:~/OS$ sh myscript 1>file8
stderr
natefoust@natefoust-X550LC:~/OS$ sh myscript 2>file8
stdout
natefoust@natefoust-X550LC:~/OS$ cat file8
stderr
```

Сначала записалось stdout, потом файл пересоздался и записалось stderr т. к. во втором случае использовали >.

6) перенаправив стандартный вывод в файл 4 с помощью символа >, а стандартный канал ошибок в файл 4 с помощью символа >>

```
natefoust@natefoust-X550LC:~/OS$ sh myscript 1>file9
stderr
natefoust@natefoust-X550LC:~/OS$ sh myscript 2>>file9
stdout
natefoust@natefoust-X550LC:~/OS$ cat file9
stdout
stderr
```

Похожая на предыдущий пункт ситуация, только мы когда перенаправляли поток ошибок использовали 2>>, а этот оператор файл добавляет в конец, а не пересоздает файл.

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

```
natefoust@natefoust-X550LC:~/OS$ cat /etc/group | sort -r | tail | head -n 3 | tail -n 1
cdrom:x:24:natefoust
natefoust@natefoust-X550LC:~/OS$
```

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

```
natefoust@natefoust-X550LC:~/OS$ ls -l /dev | grep ^b | wc -l
21
```

```
natefoust@natefoust-X550LC:~/OS$ ls -l /dev | grep ^c | wc -l
178
```

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.

```
#!/bin/bash
count=1
for i in $@
do echo "$count) $i"
count=`expr $count + 1`
done
```

```
natefoust@natefoust-X550LC:~/OS$ sh script6 1 2 3
1) 1
2) 2
3) 3
```

```
natefoust@natefoust-X550LC:~/OS$ chmod u+rx script6
natefoust@natefoust-X550LC:~/OS$ ./script6 asd wsd lsd
1) asd
2) wsd
3) lsd
```

7. Реализовать командный файл, реализующий символьное меню (в цикле):

- 1) поиск файлов в заданном каталоге, размер которых не превосходит заданного параметра;
- 2) поиск файлов в заданном каталоге по заданному времени создания.
- 3) завершение

```
#!/bin/bash
```

```
echo "1 - поиск файлов в заданном каталоге, размер которых не превосходит заданного параметра;"
```

```
echo "2 - поиск файлов в заданном каталоге по заданному времени создания;"
```

```
echo "3 - выйти."
```

```
read case
```

```
case "$case" in "1")
```

```
echo "Введите каталог: "
```

```
read dir
```

```
echo "Введите максимальный размер: "
```

```
read size
```

```
find $dir -size -$size
```

```
;;
```

```

"2")
echo "Введите каталог: "
read dir
echo "год, месяц, день: "
read year
read month
read day
day1=`expr $day + 1`
find $dir -newerct $year-$month-$day ! -newerct $year-$month-$day1
;;

"3")
exit 0
;;

esac

```

```

natefoust@natefoust-X550LC:~/OS$ sh script7
1 - поиск файлов в заданном каталоге, размер которых не превосходит заданного параметра;
2 - поиск файлов в заданном каталоге по заданному времени создания;
3 - выйти.
1
Введите каталог:
.
Введите максимальный размер:
1М
./file7
./file.txt
./2.txt
./1.txt
./5.txt
./3.txt
./link1
./new1.txt
./4.txt
natefoust@natefoust-X550LC:~/OS$ sh script7
1 - поиск файлов в заданном каталоге, размер которых не превосходит заданного параметра;
2 - поиск файлов в заданном каталоге по заданному времени создания;
3 - выйти.
2
Введите каталог:
.
год, месяц, день:
2020
04
08
./file6
./file3
./file7
./myscript
./OS3.txt
./script7
./file8
./file9

```

Вывод: изучил переадресацию, работу некоторых команд, научился писать скрипты.