

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

Лабораторная работа №3
за 4 семестр
По дисциплине: «ОСиСП»
Тема: «Bash: потоки данных. Программирование»

Выполнил:
Студент 2 курса
Группы ПО-6(1)
Лавренчик Д.О.
Проверил:
Давидюк Ю. И.

Лабораторная работа №3

Bash: потоки данных. Программирование

Цель работы: изучить работу с потоками данных в ОС Linux.

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 >>

```
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ echo "Hello word" > lab1.txt
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ cat lab1.txt
Hello word
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ echo "HI" >> lab2.txt
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ cat lab2.txt
HI
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ echo "HI word"> lab1.txt
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ cat lab1.txt
HI word
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ echo "GL HF">> lab2.txt
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ cat lab2.txt
HI
GL HF
```

Объяснить результаты.

Если файла не существует, то он создается. При использовании символа > файл перезаписывается заново, при использовании >> происходит дозапись в файл.

2. Переадресовать стандартный ввод для команды cat на файл.

```
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ cat < lab2.txt
HI
GL HF
```

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);

```
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ sh myscript
stdout
stderr
```

- перенаправив стандартный вывод в файл, просмотреть содержимое файла (sh myscript > file1);

```
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ sh myscript >1_file
stderr
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ cat 1_file
stdout
```

- перенаправить стандартный канал ошибок в существующий и несуществующий файлы с помощью символов > и >> ;

Существующий файл:

```
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ sh myscript >1_file
stderr
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ cat 1_file
stdout
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ sh myscript >file1.txt
stderr
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ cat file1.txt
stdout
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ sh myscript >>file1.txt
stderr
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ cat file1.txt
stdout
stdout
```

Несуществующие файлы:

```
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ sh myscript >> file3.txt
stderr
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ cat file3.txt
stdout
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ sh myscript> file4.txt
stderr
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ cat file4.txt
stdout
```

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

```
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ sh myscript 1>file1
stderr
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ sh myscript 2>file2
stdout
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ cat file1
stdout
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ cat file2
stderr
```

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

```
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ sh myscript 1>>file_3 2>>file_3
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ cat file_3
stdout
stderr
```

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

```
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ sh myscript 1>file4 2>>file4
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ cat file4
stdout
stderr
```

Объяснить результаты.

При перенаправлении стандартного вывода выводится строка `echo "stdout"`, а при перенаправлении канала ошибок – `echo "stderr"`, т.к. `>&2` в скрипте перенаправляет в стандартный поток ошибок

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

```
derelya@derelya-VirtualBox:~$ sort -r /etc/group | tail -15 | sed -n -e '3p' -e '6p'
dialout:x:20:
crontab:x:105:
```

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

```
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ ls -l /dev | grep '^b' | wc -l
14
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ ls -l /dev | grep '^b'
brw-rw---- 1 root disk 7, 0 кра 19 11:13 loop0
brw-rw---- 1 root disk 7, 1 кра 19 11:18 loop1
brw-rw---- 1 root disk 7, 2 кра 19 11:13 loop2
brw-rw---- 1 root disk 7, 3 кра 19 11:13 loop3
brw-rw---- 1 root disk 7, 4 кра 19 11:13 loop4
brw-rw---- 1 root disk 7, 5 кра 19 11:13 loop5
brw-rw---- 1 root disk 7, 6 кра 19 11:13 loop6
brw-rw---- 1 root disk 7, 7 кра 19 11:13 loop7
brw-rw---- 1 root disk 7, 8 кра 19 11:18 loop8
brw-rw---- 1 root disk 8, 0 кра 19 11:13 sda
brw-rw---- 1 root disk 8, 1 кра 19 11:13 sda1
brw-rw---- 1 root disk 8, 2 кра 19 11:13 sda2
brw-rw---- 1 root disk 8, 5 кра 19 11:13 sda5
brw-rw----+ 1 root cdrom 11, 0 кра 19 11:13 sr0

derelya@derelya-VirtualBox:~$ ls -l /dev | grep '^c' | wc -l
161
```

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.

```
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ touch script1
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ nano
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ cat script1
#!/bin/bash
for i in "$@"
do
echo "$i"
done
exit 0
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ sh script1 arg1 arg2 arg3 arg4
arg1
arg2
arg3
arg4
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ chmod +x script1
derelya@derelya-VirtualBox:~/lavrenchik/LABS$ ./script1 arg1 arg2
arg1
arg2
```

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

Вариант 7

7. Реализовать командный файл который при старте выводит информацию о системе, информацию о пользователе, запустившем данный командный файл, далее

в цикле выводит список активных пользователей в системе – запрашивает имя пользователя и выводят список всех процессов bash запущенных данным пользователем.

```
1 #!/bin/bash
2
3 info=`uname -a`
4 echo "Информация о системе:\n $info\n"
5
6 infouser=`whoami`
7 echo "Информация о пользователе, запустивший файл:\n $infouser\n"
8
9 infousers=`who|cut -d' ' -f1 |sort|uniq`
10 echo "Список активных пользователей:\n $infousers\n"
11
12 while read name && [ "$name" != "exit" ]
13 do
14     if [ $name != $infousers ]
15     then echo "Нет такого пользователя\n";
16
17     fi
18     ps -u "$name"|grep 'bash$';
19 done
```

Пример работы:

```
derelya@derelya-VirtualBox:~/lavrenchik/LAB5$ sh script
Информация о системе:
Linux derelya-VirtualBox 5.13.0-30-generic #33-20.04.1-Ubuntu SMP Mon Feb 7 14:25:10 UTC 2022 x86_64 x86_64 x86_64 GNU/Linux

Информация о пользователе, запустивший файл:
derelya

Список активных пользователей:
derelya

derelya
 2949 pts/0    00:00:00 bash
exit
derelya@derelya-VirtualBox:~/lavrenchik/LAB5$
```

Вывод: изучил работу с потоками данных в ОС Linux.

