

Министерство Образования Республики Беларусь
Учреждение образования
Брестский Государственный Технический Университет

Кафедра ИИТ

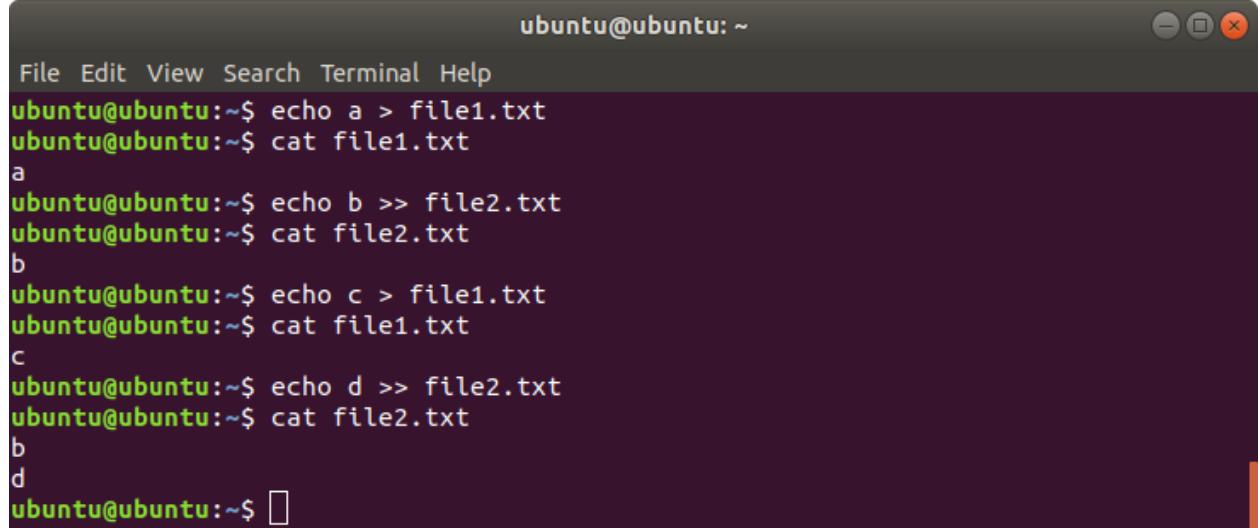
Лабораторная работа № 7
по дисциплине «СПО»
Тема: «BASH: ПОТОКИ ДАННЫХ. ПРОГРАММИРОВАНИЕ»

Выполнил:
Студент 3 курса
Факультета ЭИС
Группы АС-51
Сац А.В
Проверила:
Давидюк Ю. И.

Брест 2019

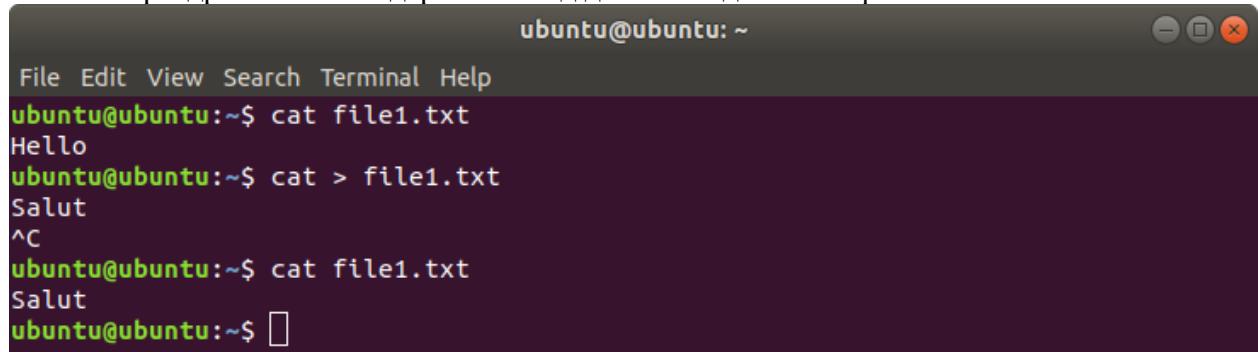
Задание для выполнения:

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 >>;



ubuntu@ubuntu:~\$ echo a > file1.txt
ubuntu@ubuntu:~\$ cat file1.txt
a
ubuntu@ubuntu:~\$ echo b >> file2.txt
ubuntu@ubuntu:~\$ cat file2.txt
b
ubuntu@ubuntu:~\$ echo c > file1.txt
ubuntu@ubuntu:~\$ cat file1.txt
c
ubuntu@ubuntu:~\$ echo d >> file2.txt
ubuntu@ubuntu:~\$ cat file2.txt
b
d
ubuntu@ubuntu:~\$

2. Переадресовать стандартный ввод для команды *cat* на файл:



ubuntu@ubuntu:~\$ cat file1.txt
Hello
ubuntu@ubuntu:~\$ cat > file1.txt
Salut
^C
ubuntu@ubuntu:~\$ cat file1.txt
Salut
ubuntu@ubuntu:~\$

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 >> (а тут и дальше уже сами :));
- 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 >>;

```
ubuntu@ubuntu:~  
File Edit View Search Terminal Help  
ubuntu@ubuntu:~$ sh myscript  
stdout  
stderr  
ubuntu@ubuntu:~$ 
```

```
ubuntu@ubuntu:~  
File Edit View Search Terminal Help  
ubuntu@ubuntu:~$ sh myscript  
stdout  
stderr  
ubuntu@ubuntu:~$ sh myscript > file1.txt  
stdout  
stderr  
ubuntu@ubuntu:~$ cat file1.txt  
stdout  
ubuntu@ubuntu:~$ 
```

```
ubuntu@ubuntu:~  
File Edit View Search Terminal Help  
ubuntu@ubuntu:~$ sh myscript 2> file1.txt  
stdout  
ubuntu@ubuntu:~$ cat file1.txt  
stderr  
ubuntu@ubuntu:~$ sh myscript 2>> file2.txt  
stdout  
ubuntu@ubuntu:~$ cat file2.txt  
stderr  
ubuntu@ubuntu:~$ sh myscript 2> file1.txt  
stdout  
ubuntu@ubuntu:~$ cat file1.txt  
stderr  
ubuntu@ubuntu:~$ sh myscript 2>> file2.txt  
stdout  
ubuntu@ubuntu:~$ cat file2.txt  
stderr  
ubuntu@ubuntu:~$ 
```

```
ubuntu@ubuntu:~  
File Edit View Search Terminal Help  
ubuntu@ubuntu:~$ sh myscript 1> file1.txt  
stderr  
ubuntu@ubuntu:~$ sh myscript 2> file2.txt  
stdout  
ubuntu@ubuntu:~$ cat file1.txt  
stdout  
ubuntu@ubuntu:~$ cat file2.txt  
stderr  
ubuntu@ubuntu:~$ 
```

```
ubuntu@ubuntu: ~
File Edit View Search Terminal Help
ubuntu@ubuntu:~$ sh myscript 1> file1.txt
stderr
ubuntu@ubuntu:~$ sh myscript 2>> file1.txt
stdout
ubuntu@ubuntu:~$ cat file1.txt
stdout
stderr
ubuntu@ubuntu:~$
```

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

```
ubuntu@ubuntu: /etc
File Edit View Search Terminal Help
messagebus:x:107:
netdev:x:108:
mlocate:x:109:
ssl-cert:x:110:
uuidd:x:111:
avahi-autoipd:x:112:
bluetooth:x:113:
rtkit:x:114:
ssh:x:115:
lpadmin:x:116:ubuntu
whoopsie:x:117:
scanner:x:118:saned
saned:x:119:
pulse:x:120:
pulse-access:x:121:
avahi:x:122:
colord:x:123:
geoclue:x:124:
gdm:x:125:
ubuntu:x:999:
sambashare:x:126:ubuntu
ubuntu@ubuntu:/etc$ sort group -r | tail -n 10 | head -n 3 | tail -n 1
cdrom:x:24:ubuntu
ubuntu@ubuntu:/etc$
```

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

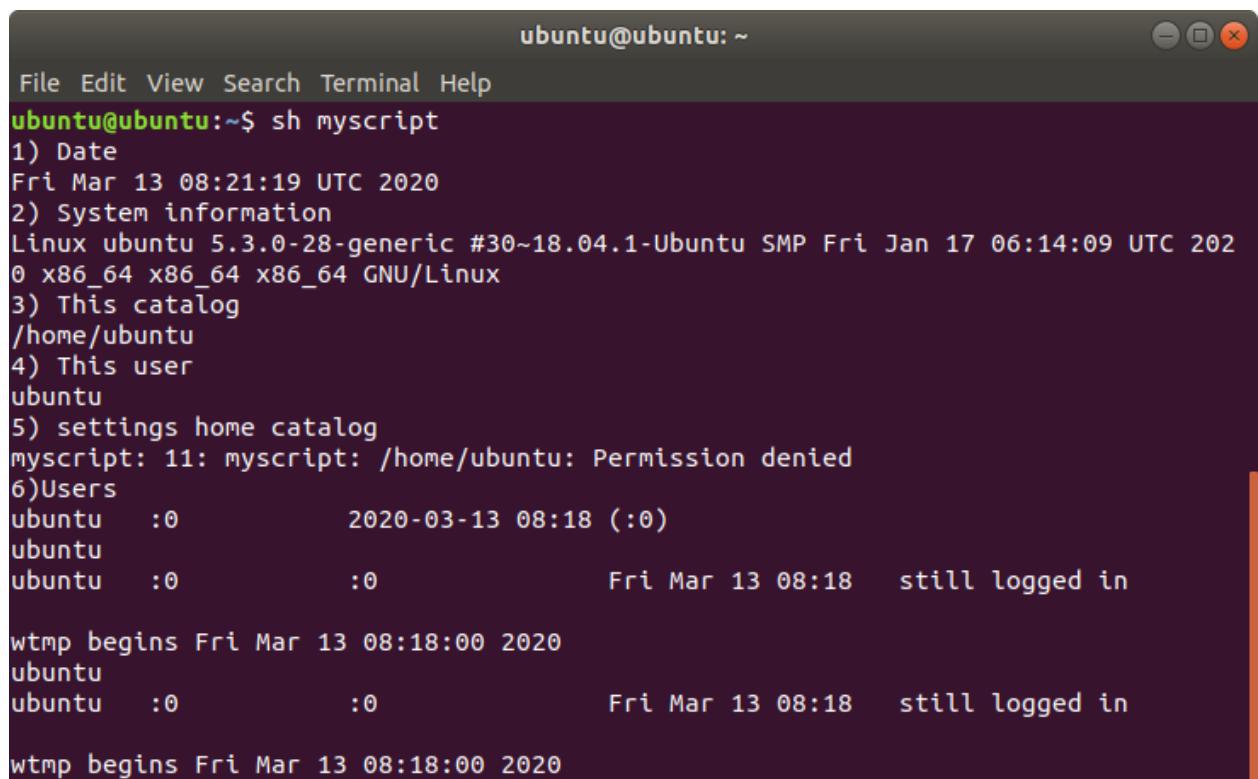
```
ubuntu@ubuntu: /etc
File Edit View Search Terminal Help
ubuntu@ubuntu:/etc$ ls -l /dev | grep '^b' | wc -l
14
ubuntu@ubuntu:/etc$
```

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/sh
While (test -n $1) do
echo $1
shift
done
```

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

```
#!/bin/bash
echo "1) Date"
date
echo "2) System information"
uname -a
echo "3) This catalog"
pwd
echo "4) This user"
whoami
echo "5) settings home catalog"
$HOME
echo "6)Users"
who
while [ 1 = 1 ]
do
    read user
    last $user
done
```



```
ubuntu@ubuntu:~$ sh myscript
1) Date
Fri Mar 13 08:21:19 UTC 2020
2) System information
Linux ubuntu 5.3.0-28-generic #30~18.04.1-Ubuntu SMP Fri Jan 17 06:14:09 UTC 202
0 x86_64 x86_64 x86_64 GNU/Linux
3) This catalog
/home/ubuntu
4) This user
ubuntu
5) settings home catalog
myscript: 11: myscript: /home/ubuntu: Permission denied
6)Users
ubuntu :0          2020-03-13 08:18 (:0)
ubuntu :0          Fri Mar 13 08:18 still logged in

wtmp begins Fri Mar 13 08:18:00 2020
ubuntu :0          :0          Fri Mar 13 08:18 still logged in

wtmp begins Fri Mar 13 08:18:00 2020
```