

# Отчёт по лабораторной работе №13

Дмитрий Алексеевич Митяков

## Содержание

Цель работы .....	1
Выполнение лабораторной работы .....	1
Контрольные вопросы.....	4
Выводы.....	5
Список литературы.....	5

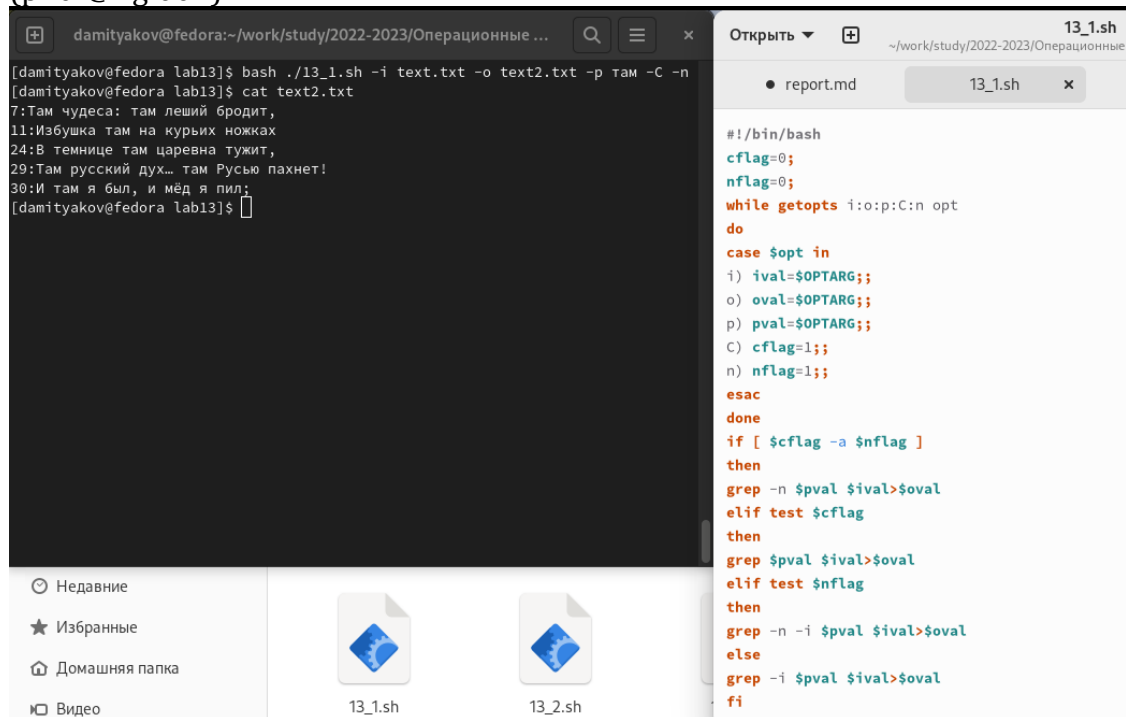
## Цель работы

Изучить основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

## Выполнение лабораторной работы

Используя команды `getopts` `grep` напишем командный файл, который анализирует командную строку с ключами и выполним его: `-i inputfile` — прочитать данные из указанного файла; `-o outputfile` — вывести данные в указанный файл; `-r шаблон` — указать шаблон для поиска; `-C` — различать большие и малые буквы; `-n` — выдавать номера строк; а затем ищет в указанном файле нужные строки

(рис. @fig:001).



The image shows a terminal window and a file manager. The terminal window displays the execution of a script `13_1.sh` with the following output:

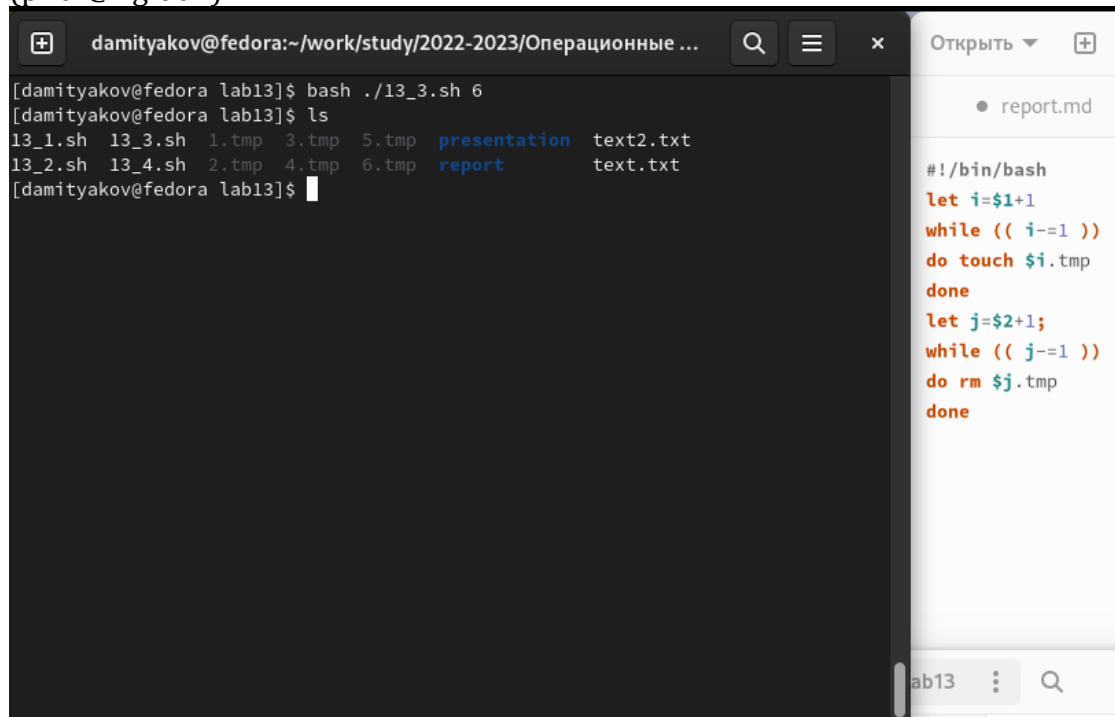
```
[damityakov@fedora lab13]$ bash ./13_1.sh -i text.txt -o text2.txt -p там -C -n
[damityakov@fedora lab13]$ cat text2.txt
7:Там чудеса: там леший бродит,
11:Избушка там на курьих ножках
24:В темнице там царевна тужит,
29:Там русский дух... там Русью пахнет!
30:И там я был, и мёд я пил:
[damityakov@fedora lab13]$
```

The file manager shows two files, `13_1.sh` and `13_2.sh`. The content of `13_1.sh` is as follows:

```
#!/bin/bash
cflag=0;
nflag=0;
while getopts i:o:p:C:n opt
do
case $opt in
i) ival=$OPTARG;;
o) oval=$OPTARG;;
p) pval=$OPTARG;;
C) cflag=1;;
n) nflag=1;;
esac
done
if [ $cflag -a $nflag ]
then
grep -n $pval $ival>$oval
elif test $cflag
then
grep $pval $ival>$oval
elif test $nflag
then
grep -n -i $pval $ival>$oval
else
grep -i $pval $ival>$oval
fi
```

Напишем сначала на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем завершим программу при помощи функции `exit(n)`, передавая информацию о коде завершения в оболочку. Командный файл вызовет эту программу и, проанализировав с помощью команды `$?`, выдаст сообщение о том, какое число было введено

(рис. @fig:002).



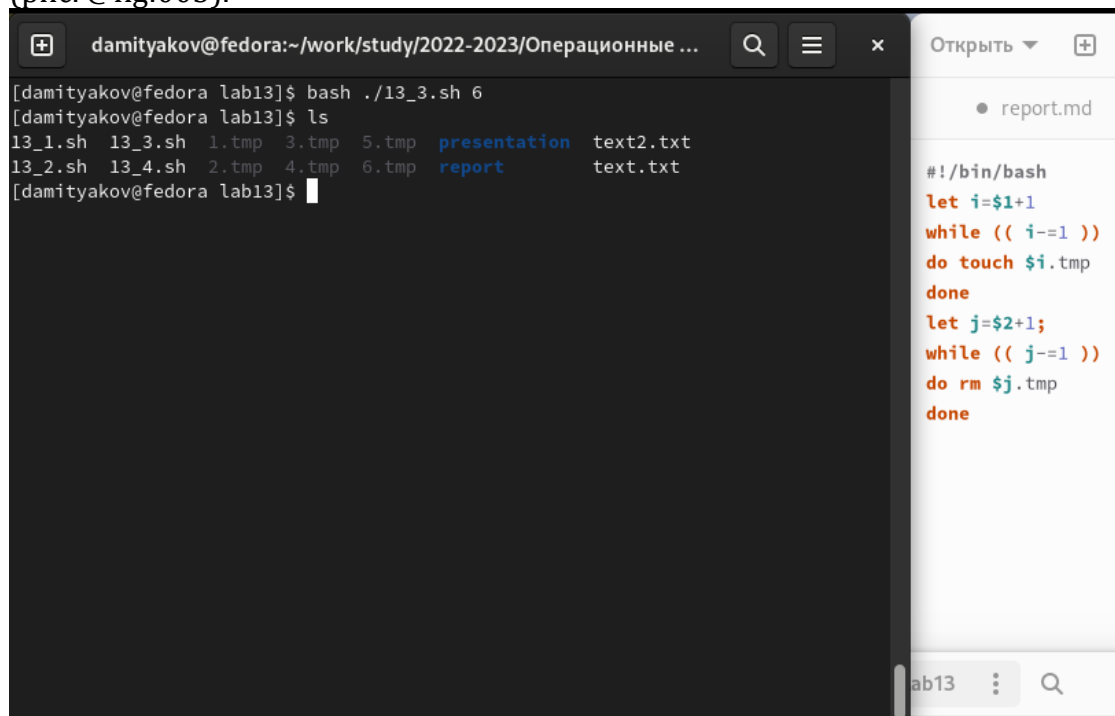
```
damityakov@fedora:~/work/study/2022-2023/Операционные ...
[damityakov@fedora lab13]$ bash ./13_3.sh 6
[damityakov@fedora lab13]$ ls
13_1.sh 13_3.sh 1.tmp 3.tmp 5.tmp presentation text2.txt
13_2.sh 13_4.sh 2.tmp 4.tmp 6.tmp report text.txt
[damityakov@fedora lab13]$
```

The terminal window shows the execution of a script `./13_3.sh 6`. After running `ls`, the output lists files including `13_1.sh`, `13_3.sh`, `1.tmp`, `3.tmp`, `5.tmp`, `presentation`, `text2.txt`, `13_2.sh`, `13_4.sh`, `2.tmp`, `4.tmp`, `6.tmp`, `report`, and `text.txt`. A file explorer on the right shows `report.md` with its contents:

```
#!/bin/bash
let i=$1+1
while (( i--=1 ))
do touch $i.tmp
done
let j=$2+1;
while (( j--=1 ))
do rm $j.tmp
done
```

Напишем командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N

(рис. @fig:003).



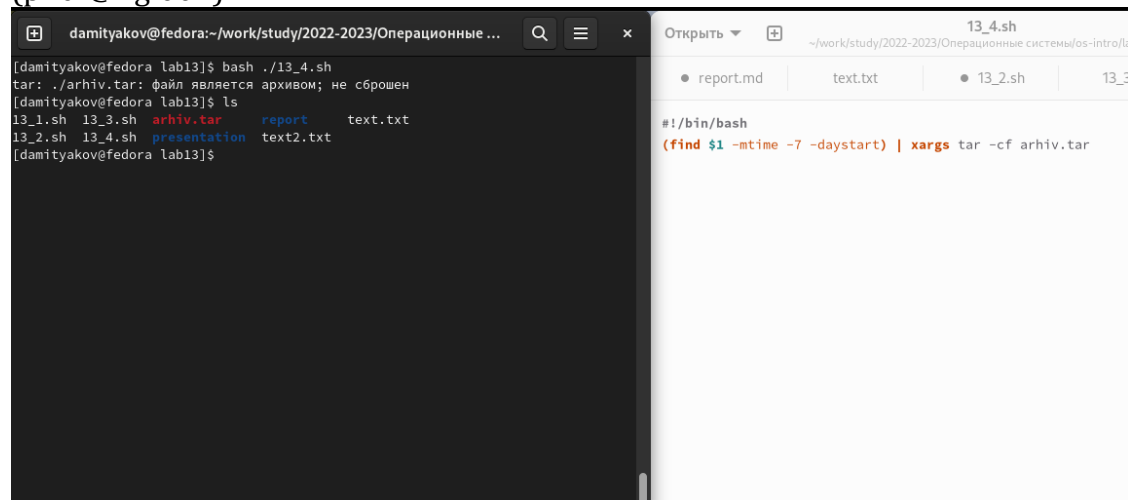
```
damityakov@fedora:~/work/study/2022-2023/Операционные ...
[damityakov@fedora lab13]$ bash ./13_3.sh 6
[damityakov@fedora lab13]$ ls
13_1.sh 13_3.sh 1.tmp 3.tmp 5.tmp presentation text2.txt
13_2.sh 13_4.sh 2.tmp 4.tmp 6.tmp report text.txt
[damityakov@fedora lab13]$
```

The terminal window shows the execution of a script `./13_3.sh 6`. After running `ls`, the output lists files including `13_1.sh`, `13_3.sh`, `1.tmp`, `3.tmp`, `5.tmp`, `presentation`, `text2.txt`, `13_2.sh`, `13_4.sh`, `2.tmp`, `4.tmp`, `6.tmp`, `report`, and `text.txt`. A file explorer on the right shows `report.md` with its contents:

```
#!/bin/bash
let i=$1+1
while (( i--=1 ))
do touch $i.tmp
done
let j=$2+1;
while (( j--=1 ))
do rm $j.tmp
done
```

Напишем командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории. Модифицируем его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад.

(рис. @fig:004).



The image shows a terminal window on the left and a file editor on the right. The terminal window displays the following commands and output:

```
[damityakov@fedora lab13]$ bash ./13_4.sh
tar: ./arhiv.tar: файл является архивом; не сброшен
[damityakov@fedora lab13]$ ls
13_1.sh 13_3.sh arhiv.tar report text.txt
13_2.sh 13_4.sh presentation text2.txt
[damityakov@fedora lab13]$
```

The file editor on the right shows the content of the file 13\_4.sh:

```
#!/bin/bash
(find $1 -mtime -7 -daystart) | xargs tar -cf arhiv.tar
```

## Контрольные вопросы

1. Каково предназначение команды getopts?
  - Создание по пользовательским аргументам.
2. Какое отношение метасимволы имеют к генерации имён файлов?
  - Используют как файлы так и аргументы.
3. Какие операторы управления действиями вы знаете?
  - If, else, elif, fi, while, do, done, until, do, done, for, in, do, done, case, in, esac
4. Какие операторы используются для прерывания цикла?
  - - a) for – будет выполнять действие до тех пор, пока есть объекты для выполнения. b) while – выполняет действие до тех пор, пока условие является истинным.
    - c) until – будет выполняться пока условие не станет правдиво.
5. Для чего нужны команды false и true?
  - Эти команды указывают, ложные и правдивые утверждения
6. Что означает строка if test -f mans/i.\$s, встреченная в командном файле?
  - Проверяет если существует файл его размерность и тип с двумя разными расширениями, заменяя через переменные.
7. Объясните различия между конструкциями while и until.
  - while – выполняет действие до тех пор, пока условие является истинным. until – будет выполняться до тех пор, пока условие не станет истинным, т.е. пока оно false.

## Выводы

Я изучил основы программирования в оболочке ОС UNIX. Научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

## Список литературы