

# **Отчёт по лабораторной работе №13**

**Дисциплина: операционные системы**

Казаазев Даниил Михайлович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
3.1	Задание 1 . . . . .	7
<b>4</b>	<b>Задание 2</b>	<b>11</b>
<b>5</b>	<b>Задание 3</b>	<b>14</b>
<b>6</b>	<b>Задание 4</b>	<b>16</b>
<b>7</b>	<b>Выводы</b>	<b>17</b>

# Список иллюстраций

3.1	Создание каталога и файлов для заданий . . . . .	7
3.2	Перенесенный текст . . . . .	8
3.3	Код программы . . . . .	9
3.4	Запуск программы . . . . .	10
3.5	Результат программы . . . . .	10
4.1	Код программы на языке C . . . . .	11
4.2	Код программы на языке Bash . . . . .	12
4.3	Результат выполнения программы . . . . .	13
5.1	Код третьей программы . . . . .	14
5.2	Результат программы . . . . .	15
5.3	Результат программы . . . . .	15
6.1	Код четвертой программы . . . . .	16
6.2	Результат выполнения . . . . .	16

## **Список таблиц**

# 1 Цель работы

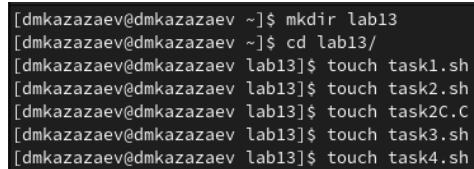
Изучить основы программирования в оболочке ОС UNIX, научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

## **2 Задание**

1. Выполнить задания лабораторной работы

## 3 Выполнение лабораторной работы

Создаю каталог и файлы для заданий. (рис. 3.1)



```
[dmkazazaev@dmkazazaev ~]$ mkdir lab13  
[dmkazazaev@dmkazazaev ~]$ cd lab13/  
[dmkazazaev@dmkazazaev lab13]$ touch task1.sh  
[dmkazazaev@dmkazazaev lab13]$ touch task2.sh  
[dmkazazaev@dmkazazaev lab13]$ touch task2C.C  
[dmkazazaev@dmkazazaev lab13]$ touch task3.sh  
[dmkazazaev@dmkazazaev lab13]$ touch task4.sh
```

Рис. 3.1: Создание каталога и файлов для заданий

### 3.1 Задание 1

Создаю еще два файла для первого задания и переношу в один из них текст.  
(рис. 3.2)

```
У лукоморья дуб зелёный;  
Златая цепь на дубе том:  
И днём и ночью кот учёный  
Всё ходит по цепи кругом;  
Идёт направо — песнь заводит,  
Налево — сказку говорит.  
Там чудеса: там леший бродит,  
Русалка на ветвях сидит;  
Там на неведомых дорожках  
Следы невиданных зверей;  
Избушка там на курьих ножках  
Стоит без окон, без дверей;  
Там лес и дол видений полны;  
Там о заре прихлынут волны  
На брег песчаный и пустой,  
И тридцать витязей прекрасных  
Чредой из вод выходят ясных,  
И с ними дядька их морской;  
Там королевич мимоходом
```

Рис. 3.2: Перенесенный текст

После переноса текста перехожу в файл с расширением .sh и пишу программу, которая соответствует заданию. (рис. 3.3)



```

1  #! /bin/bash
2
3  while getopts i:o:p:cn optionletters
4  do
5      case $optionletters in
6          i) iflag=1; ival=$OPTARG;;
7          o) oflag=1; oval=$OPTARG;;
8          p) pflag=1; pval=$OPTARG;;
9          c) cflag=1;;
10         n) nflag=1;;
11     esac
12 done
13
14 if ! test $cflag
15 then
16     cf=-i
17 fi
18
19 if test $nflag
20 then
21     nf=-n
22 fi
23
24 grep $cf $nf $pval $ival >> $oval
25

```

Рис. 3.3: Код программы

Листинг первой программы:

```

#! /bin/bash

while getopts i:o:p:cn optionletters
do
    case $optionletters in
        i) iflag=1; ival=$OPTARG;;
        o) oflag=1; oval=$OPTARG;;
        p) pflag=1; pval=$OPTARG;;
        c) cflag=1;;
        n) nflag=1;;
    esac
done

```

```

if ! test $cflag
    then
        cf=-i
    fi

if test $nflag
    then
        nf=-n
    fi

grep $cf $nf $pval $ival >> $oval

```

После чего запускаю программу. (рис. 3.4)

```

[dmkazazaev@dmkazazaev lab13]$ bash task1.sh -p дуб -i inputTest.txt -o outputTest.txt -c -n
[dmkazazaev@dmkazazaev lab13]$ kwrite outputTest.txt

```

Рис. 3.4: Запуск программы

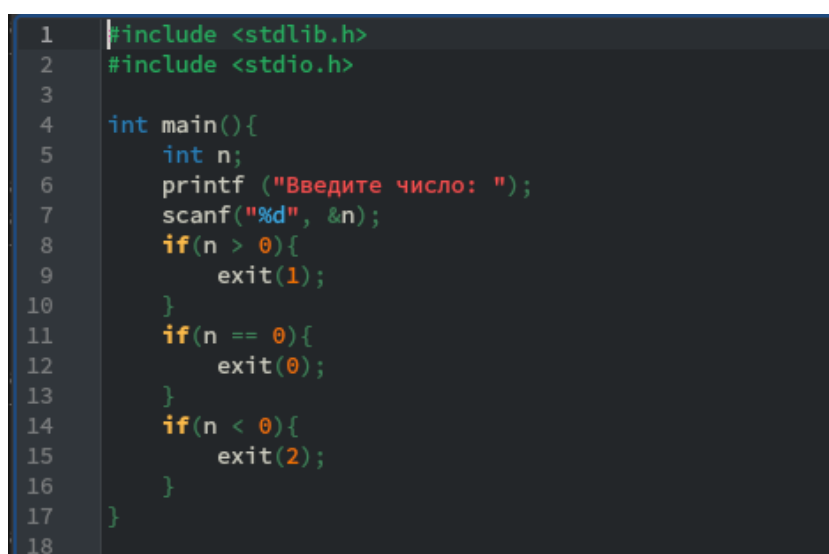
Открываю второй файл, чтобы проверить выполнение программы. (рис. 3.5)

1	1:У лукоморья дуб зелёный;
2	2:Златая цепь на дубе том:
3	31:У моря видел дуб зелёный;
4	

Рис. 3.5: Результат программы

## 4 Задание 2

В файле с расширением .c пишу программу, которая сравнивает число с нулем.  
(рис. 4.1)



```
1  #include <stdlib.h>
2  #include <stdio.h>
3
4  int main(){
5      int n;
6      printf ("Введите число: ");
7      scanf("%d", &n);
8      if(n > 0){
9          exit(1);
10     }
11     if(n == 0){
12         exit(0);
13     }
14     if(n < 0){
15         exit(2);
16     }
17 }
18
```

Рис. 4.1: Код программы на языке C

Листинг программы на языке C:

```
#include <stdlib.h>
#include <stdio.h>

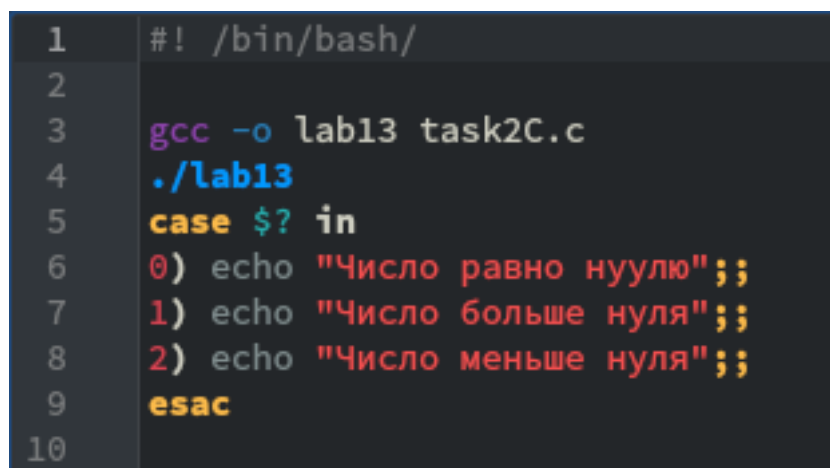
int main(){
    int n;
    printf ("Введите число: ");
    scanf("%d", &n);
```

```

if(n > 0){
    exit(1);
}
if(n == 0){
    exit(0);
}
if(n < 0){
    exit(2);
}
}

```

После чего перехожу в файл с расширением .sh и пишу программу, которая будет вызывать программу на языке С и, проанализировав с помощью \$?, вывести на экран результат сравнения. (рис. 4.2)



```

1  #! /bin/bash/
2
3  gcc -o lab13 task2C.c
4  ./lab13
5  case $? in
6  0) echo "Число равно нулю";;
7  1) echo "Число больше нуля";;
8  2) echo "Число меньше нуля";;
9  esac
10

```

Рис. 4.2: Код программы на языке Bash

Листинг программы на языке Bash:

```

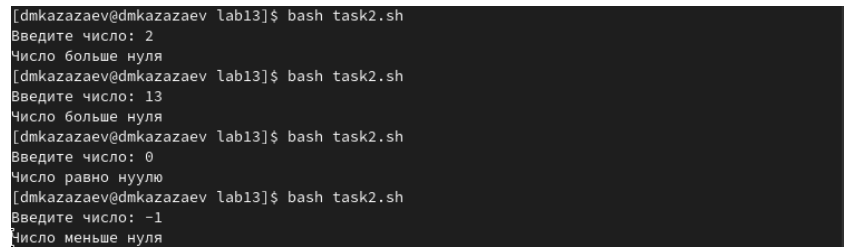
#!/bin/bash/

gcc -o lab13 task2C.c
./lab13

```

```
case $? in
0) echo "Число равно нулю";;
1) echo "Число больше нуля";;
2) echo "Число меньше нуля";;
esac
```

Проверяю работу программы. (рис. 4.3)

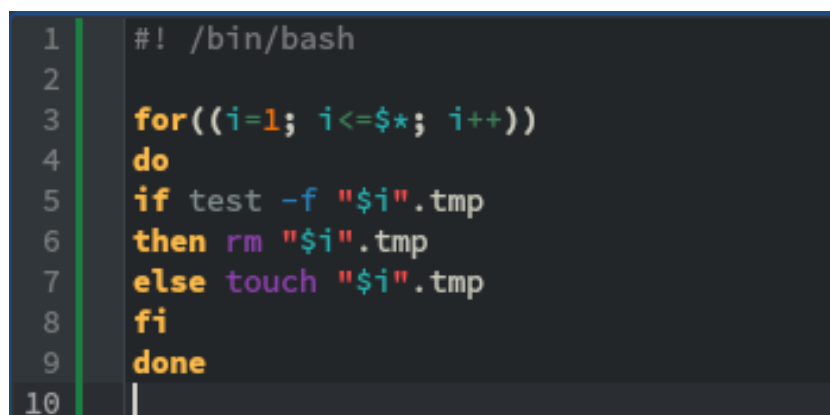


```
[dmkazazaev@dmkazazaev lab13]$ bash task2.sh
Введите число: 2
Число больше нуля
[dmkazazaev@dmkazazaev lab13]$ bash task2.sh
Введите число: 13
Число больше нуля
[dmkazazaev@dmkazazaev lab13]$ bash task2.sh
Введите число: 0
Число равно нулю
[dmkazazaev@dmkazazaev lab13]$ bash task2.sh
Введите число: -1
Число меньше нуля
```

Рис. 4.3: Результат выполнения программы

## 5 Задание 3

Перехожу в третий файл и пишу программу, которая будет создавать n-ное число временных файлов, а если они уже существуют, что удалять их. (рис. 5.1)



```
1  #!/bin/bash
2
3  for((i=1; i<=$*; i++))
4  do
5  if test -f "$i".tmp
6  then rm "$i".tmp
7  else touch "$i".tmp
8  fi
9  done
10
```

Рис. 5.1: Код третьей программы

Листинг третьей программы:

```
#!/bin/bash

for((i=1; i<=$*; i++))
do
if test -f "$i".tmp
then rm "$i".tmp
else touch "$i".tmp
fi
done
```

Запускаю парогрaмму и проверяю, создались ли файлы. (рис. 5.2)

```
[dmkazazaev@dmkazazaev lab13]$ bash task3.sh 3
[dmkazazaev@dmkazazaev lab13]$ ls
1.tmp  3.tmp  lab13  task1.sh  task2.sh  task4.sh
2.tmp  inputTest.txt  outputTest.txt  task2C.c  task3.sh
```

Рис. 5.2: Результат программы

Запускаю парогрaмму еще раз и проверяю, удалились ли файлы. (рис. 5.3)

```
[dmkazazaev@dmkazazaev lab13]$ bash task3.sh 3
[dmkazazaev@dmkazazaev lab13]$ ls
inputTest.txt  lab13  outputTest.txt  task1.sh  task2C.c  task2.sh  task3.sh  task4.sh
```

Рис. 5.3: Результат программы

## 6 Задание 4

Перехожу в четвертый файл и пишу программу, которая будет архивировать файлы, которые были изменены менее недели назад. (рис. 6.1)

```
#!/bin/bash

find $* -mtime -7 -mtime +0 -type f > files.txt
tar -cf test4.tar -T files.txt
```

Рис. 6.1: Код четвертой программы

Листинг четвертой программы:

```
#!/bin/bash

find $* -mtime -7 -mtime +0 -type f > files.txt
tar -cf test4.tar -T files.txt
```

Запускаю программу и проверяю результат выполнения. (рис. 6.2)

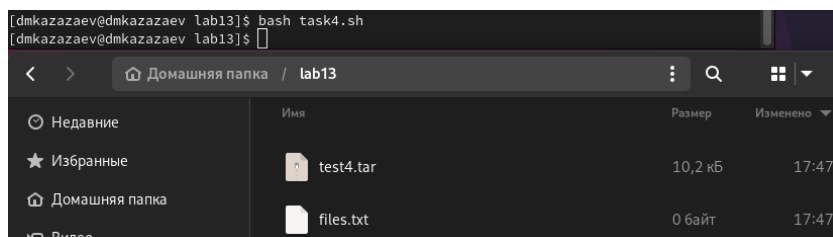


Рис. 6.2: Результат выполнения



## 7 Выводы

В результате выполнения лабораторной работы я изучил основы программирования в оболочке ОС UNIX, научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.