

# **Лабораторная 11**

**Программирование в программном процессоре ОС UNIX**

Осокин Георгий Иванович

# Содержание

<b>1</b>	<b>НАЧАЛО ☒</b>	<b>5</b>
1.1	Цель работы . . . . .	5
<b>2</b>	<b>Выполнение работы</b>	<b>6</b>
2.1	Программа с getops . . . . .	6
2.2	Программа на С определяющая положительность числа . . . . .	8
2.3	Программа нумерующая файлы . . . . .	10
2.4	Программа сжимающая файлы . . . . .	12
<b>3</b>	<b>Заключение</b>	<b>13</b>
3.1	Выводы . . . . .	13

# Список иллюстраций

2.1	Бараш.sh	6
2.2	Бараш2.sh	8
2.3	Бараш3.sh	10
2.4	Барашбараш4.sh	12

## **Список таблиц**

# 1 НАЧАЛО ☒

## 1.1 Цель работы

Изучить основы программирования в оболочке ОС UNIX. Научиться писать более сложные командные файлы.

## 2 Выполнение работы

### 2.1 Программа с getops

```
## About frontend clients
The purpose of "hole project" is to write independent frontend clients for each platform,
as a hole-core is just a logic part, which should be multiplatform, **frontend shouldn't*
k", etc...

-----

# TODO
- [x] Basic sending and receiving messages
- [ ] Add RSA or NTRU encryption
- [ ] Write tests
- [ ] Make more verbose documentation
- [ ] Create a hole-daemon

→ code git:(master) X ./task1.sh -p "-" -i /tmp/kek.md -o /tmp/out.md
→ code git:(master) X cat /tmp/out.md
1. Clone [this](https://github.com/hole-chat/core) repo
3. Install and launch client (for example [rabbit-hole](https://github.com/hole-chat/rabb
-----
As a hole-core is just a logic part, which should be multiplatform, **frontend shouldn't*
k", etc...

-----

- [x] Basic sending and receiving messages
- [ ] Add RSA or NTRU encryption
- [ ] Write tests
- [ ] Make more verbose documentation
- [ ] Create a hole-daemon

→ code git:(master) X ./task1.sh -p "-" -i /tmp/kek.md -o /tmp/out.md
```

Рис. 2.1: Барах.sh

Код:

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

```
file=''
```

```
filecontent=''
```

```
case_sensitive=0
```

```
line_numbers=0
```

```

template='0'

get-template () {
    template=$1
}

get-input () {
    file=$1
}

put-input () {
    nums=$( [ "$line_numbers" = "1" ] && echo "-n")
    lines=$( [ "$case_sensitive" = "0" ] && echo "-i" )
    echo - $nums $lines
    grep $template $file $nums $lines > $1
}

helpme () {
    echo "-p to specify the template"
    echo "-C case sensitivity ( 0 or 1 )"
    echo "-n show line numbers"
    echo "-p template string"
    echo "-i input file"
    echo "-o output file"
}

while getopts 'p:n:i:o:C:' opt; do
    case $opt in
        p) get-template $OPTARG ;;

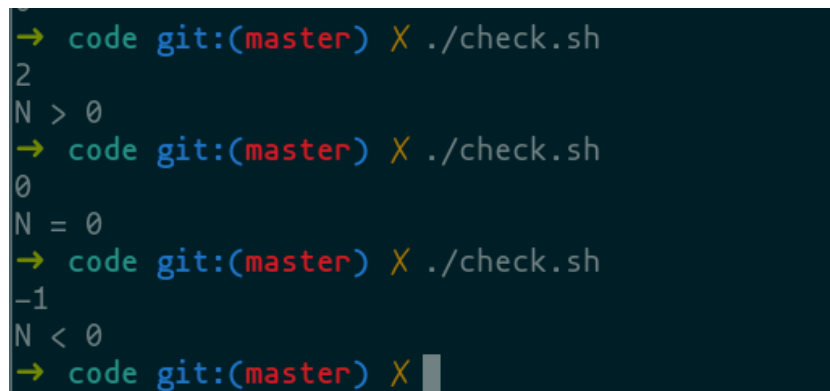
```

```

C) case_sensitive=1 ;;
n) line_numbers=1 ;;
i) get-input "$OPTARG" ;;
o) put-input "$OPTARG" ;;
?) echo "YOU'RE WRONG ☒"; helpme ; exit 1;
esac
done

```

## 2.2 Программа на С определяющая положительность числа



```

→ code git:(master) X ./check.sh
2
N > 0
→ code git:(master) X ./check.sh
0
N = 0
→ code git:(master) X ./check.sh
-1
N < 0
→ code git:(master) X

```

Рис. 2.2: Бараш2.sh

Код:

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

```
/*
```

Написать на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем программа завершается с помощью функции `exit(n)`, передавая информацию в о коде завершения в оболочку. Командный ф

```
*/
```



```

int main () {
    int num = 0;
    scanf("%d", &num);
    if(num > 0 ) {
        exit(0);
    } else if (num < 0) {
        exit(1);
    } else {
        exit(2);
    }

    return 0;
}

```

check.sh

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

```
./МАМА-Я_ОБ_ЭТОМ_НЕ_ЗНАЛ-НО_НА_ПАРАХ_МЫ_УЧИЛИ_СИ.bin
```

```
var=$?
```

```

case $var in
    0) echo "N > 0";;
    1) echo "N < 0";;
    2) echo "N = 0";;
    ?) echo "IT'S OTHER ERROR!!!!"; exit 1;;
esac

```

## 2.3 Программа нумерующая файлы



```
→ code git:(master) X ./numbers.sh -n 5 -f "kek{N}.txt"
Bash version ...
CREATING FILES
→ code git:(master) X ls kek*
kek kek0.txt kek1.txt kek2.txt kek3.txt kek4.txt kek5.txt
→ code git:(master) X ./numbers.sh -n 5 -f "kek{N}.txt" -r
Bash version ...
CREATING FILES
REMOVING FILES
→ code git:(master) X ls
a.out check.sh kek numbers.sh task1.sh МАМА-Я_ОБ_ЭТОМ_НЕ_ЗНАЛ-НО_НА_ПАРАХ_МЫ_УЧИЛИ_СИ.bin
cat check.sh~ meow task1.sh~ МАМА-Я_ОБ_ЭТОМ_НЕ_ЗНАЛ-НО_НА_ПАРАХ_МЫ_УЧИЛИ_СИ.c
→ code git:(master) X
```

Рис. 2.3: Бараш3.sh

Код:

```
#!/bin/zsh
```

```
echo "Bash version ${BASH_VERSION}..."
```

```
N=1
```

```
file=''
```

```
setN () {
```

```
    N=$1
```

```
}
```

```
remove-magic (){
```

```
    echo "REMOVING FILES"
```

```
    for i in {0..$N..1}; do
```

```
        new_file=$(echo $file | sed -e $(echo "s/{N}/"$i"/g"))
```

```
        rm $new_file;
```

```
    done
```

```
}
```

```
call-magic (){
```

```

    echo "CREATING FILES"
    for i in {0..$N..1}; do
        new_file=$(echo $file | sed -e $(echo "s/{N}/"$i"/g"))
        touch $new_file;
    done
}

helpme () {
    echo "-f using to specify file, -n to set files count. File name should be wr
}

while getopts ':n:f:r' opt; do
    case $opt in
        n) setN $OPTARG ;;
        f) file=$OPTARG; call-magic $file ;;
        r) remove-magic;;
        ?) echo "YOU'RE WORNG ☒"; helpme; exit 1;
    esac
done

```

## 2.4 Программа сжимающая файлы

```
code git:(master) X ./tar-current-dir.sh
./
./cat
./meow
./task1.sh~
./task1.sh
./kek
./МАМА-Я_ОБ_ЭТОМ_НЕ_ЗНАЛ-НО_НА_ПАРАХ_МЫ_УЧИЛИ_СИ.с~
./МАМА-Я_ОБ_ЭТОМ_НЕ_ЗНАЛ-НО_НА_ПАРАХ_МЫ_УЧИЛИ_СИ.с
./a.out
./МАМА-Я_ОБ_ЭТОМ_НЕ_ЗНАЛ-НО_НА_ПАРАХ_МЫ_УЧИЛИ_СИ.bin
./check.sh~
./check.sh
./numbers.sh~
./numbers.sh
./.#week-long-tar.sh
./#week-long-tar.sh#
./tar-current-dir.sh~
./tar-current-dir.sh
tar: ./last_week_modified.tar: file is the archive; not dumped
./cat
./meow
./task1.sh~
./task1.sh
./kek
./МАМА-Я_ОБ_ЭТОМ_НЕ_ЗНАЛ-НО_НА_ПАРАХ_МЫ_УЧИЛИ_СИ.с~
./МАМА-Я_ОБ_ЭТОМ_НЕ_ЗНАЛ-НО_НА_ПАРАХ_МЫ_УЧИЛИ_СИ.с
./a.out
./МАМА-Я_ОБ_ЭТОМ_НЕ_ЗНАЛ-НО_НА_ПАРАХ_МЫ_УЧИЛИ_СИ.bin
./check.sh~
./check.sh
./numbers.sh~
./numbers.sh
./.#week-long-tar.sh
./#week-long-tar.sh#
./tar-current-dir.sh~
./tar-current-dir.sh
code git:(master) X ls
a.out          tar-current-dir.sh
cat            task1.sh
check.sh      task1.sh~
check.sh~
kek
last_week_modified.tar  МАМА-Я_ОБ_ЭТОМ_НЕ_ЗНАЛ-НО_НА_ПАРАХ_МЫ_УЧИЛИ_СИ.bin
meow          МАМА-Я_ОБ_ЭТОМ_НЕ_ЗНАЛ-НО_НА_ПАРАХ_МЫ_УЧИЛИ_СИ.с
```

Рис. 2.4: Барашбараш4.sh

Код:

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

```
tar -cvf last_week_modified.tar -o $(find -mtime -7)
```

## **3 Заключение**

### **3.1 Выводы**

МЫ ВЫУЧИЛИ ЯЗЫК С! ПАПА ТОЖЕ МОЖЕТ В СИ, ПАПА МОЖЕТ В СИ! ПОКА  
ВСЕ ТАК, ВСЕ В ПОРЯДКЕ НА РУСИ!