

# **Лабораторная работа №11**

**Программирование в командном процессоре ОС UNIX. Ветвления и  
циклы**

Салькова Кристина Михайловна

# Содержание

|          |                                       |           |
|----------|---------------------------------------|-----------|
| <b>1</b> | <b>Цель работы</b>                    | <b>5</b>  |
| <b>2</b> | <b>Задание</b>                        | <b>6</b>  |
| <b>3</b> | <b>Теоретическое введение</b>         | <b>7</b>  |
| <b>4</b> | <b>Выполнение лабораторной работы</b> | <b>8</b>  |
| <b>5</b> | <b>Выводы</b>                         | <b>12</b> |
|          | <b>Список литературы</b>              | <b>13</b> |

## Список иллюстраций

|     |                          |    |
|-----|--------------------------|----|
| 4.1 | создание файла . . . . . | 8  |
| 4.2 | код1 . . . . .           | 9  |
| 4.3 | проверка . . . . .       | 9  |
| 4.4 | код с . . . . .          | 10 |
| 4.5 | код sh . . . . .         | 10 |
| 4.6 | проверка . . . . .       | 10 |
| 4.7 | код3 . . . . .           | 11 |
| 4.8 | код4 . . . . .           | 11 |

## Список таблиц

# 1 Цель работы

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

## 2 Задание

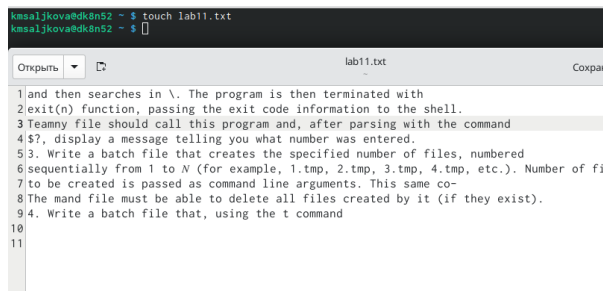
1. Используя команды `getopts` `grep`, написать командный файл, который анализирует командную строку с ключами: `-iinputfile` — прочитать данные из указанного файла; `-ooutputfile` — вывести данные в указанный файл; `-rшаблон` — указать шаблон для поиска; `-C` — различать большие и малые буквы; `-n` — выдавать номера строк. а затем ищет в указанном файле нужные строки, определяемые ключом `-p`.
2. 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 файл должен вызывать эту программу и, проанализировав с помощью команды `$?`, выдать сообщение о том, какое число было введено.
3. Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до  $\infty$  (например `1.tmp`, `2.tmp`, `3.tmp`, `4.tmp` и т.д.). Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же командный файл должен уметь удалять все созданные им файлы (если они существуют).
4. Написать командный файл, который с помощью команды `tag` запаковывает в архив все файлы в указанной директории. Модифицировать его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовать команду `find`).

### **3 Теоретическое введение**

## 4 Выполнение лабораторной работы

1. 1. Используя команды `getopts` `grep`, написать командный файл, который анализирует командную строку с ключами:

- `-iinputfile` — прочитать данные из указанного файла;
- `-ooutputfile` — вывести данные в указанный файл;
- `-р` — указать шаблон для поиска;
- `-C` — различать большие и малые буквы;
- `-n` — выдавать номера строк. а затем ищет в указанном файле нужные строки, определяемые ключом `-р`.



```
kmsaljkova@dkn52 ~ $ touch lab11.txt
kmsaljkova@dkn52 ~ $
```

Открыть | lab11.txt | Сохранить

```
1 and then searches in \. The program is then terminated with
2 exit(n) function, passing the exit code information to the shell.
3 Teammy file should call this program and, after parsing with the command
4 $?, display a message telling you what number was entered.
5 3. Write a batch file that creates the specified number of files, numbered
6 sequentially from 1 to N (for example, 1.tmp, 2.tmp, 3.tmp, 4.tmp, etc.). Number of fi
7 to be created is passed as command line arguments. This same co-
8 The mand file must be able to delete all files created by it (if they exist).
9 4. Write a batch file that, using the t command
10
11
```

Рис. 4.1: создание файла



```
Открыть lab11.sh Сохранить
1 #!/bin/bash
2 iflag=0; oflag=0; pflag=0; cflag=0; nflag=0;
3 while getopts i:op:C:n optletter
4 do case $optletter in
5   i) iflag=1; ival=$OPTARG;;
6   o) oflag=1; oval=$OPTARG;;
7   p) pflag=1; pval=$OPTARG;;
8   C) cflag=1;;
9   n) nflag=1;;
10  *) echo illegal option $optletter
11  esac
12 done
13 if (($flag==0))
14 then echo "Шаблон не найден"
15 else
16   if (($iflag==0))
17   then echo "Файл не найден"
18   else
19     if (($oflag==0))
20     then if (($cflag==0))
21     then if (($nflag==0))
22     then grep $pval $ival
23     else grep -n $pval $ival
24     fi
25     else if (($nflag==0))
26     then grep -i $pval $ival
27     else grep -i -n $pval $ival
28     fi
29     fi
30     else if (($cflag==0))
31     then if (($nflag==0))
32     then grep $pval $ival > $oval
33     else grep -n $pval $ival > $oval
34     fi
35     else if (($nflag==0))
36     then grep -i $pval $ival > $oval
37     else grep -i -n $pval $ival > $oval
38     fi
39     fi
40   fi
41 fi
42 fi
```

Рис. 4.2: код1

```
kmsaljkova@dkn52 ~ $ ./lab11.sh -i ~/lab11.txt -o ~/lab11-1.txt -p it -C -n
kmsaljkova@dkn52 ~ $ cat ~/lab11-1.txt
and then searches in \. The program is then terminated with
exit(n) function, passing the exit code information to the shell.
3:Teammy file should call this program and, after parsing with the command
3. Write a batch file that creates the specified number of files, numbered
The mand file must be able to delete all files created by it (if they exist).
4. Write a batch file that, using the t command
kmsaljkova@dkn52 ~ $ ./lab11.sh -i ~/lab11.txt -o ~/lab11-1.txt -p it -n
kmsaljkova@dkn52 ~ $ cat ~/lab11-1.txt
1:and then searches in \. The program is then terminated with
2:exit(n) function, passing the exit code information to the shell.
3:Teammy file should call this program and, after parsing with the command
3:3. Write a batch file that creates the specified number of files, numbered
8:The mand file must be able to delete all files created by it (if they exist).
9:4. Write a batch file that, using the t command
kmsaljkova@dkn52 ~ $ ./lab11.sh -i ~/lab11.txt -C -n
Шаблон не найден
kmsaljkova@dkn52 ~ $ ./lab11.sh -o ~/lab11-1.txt -p it -n
./lab11.sh: строка 16: ((: ==0: синтаксическая ошибка: ожидается операнд (неверный маркер «==0»)
^C
kmsaljkova@dkn52 ~ $ ./lab11.sh -o ~/lab11-1.txt -p it -n
./lab11.sh: строка 16: ((: ==0: синтаксическая ошибка: ожидается операнд (неверный маркер «==0»)
^C
kmsaljkova@dkn52 ~ $ ./lab11.sh -o ~/lab11-1.txt -p it -n
Файл не найден
kmsaljkova@dkn52 ~ $ ./lab11.sh -o ~/lab11-1.txt -p it -n
```

Рис. 4.3: проверка

2. 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. Команд- ный файл должен вызывать эту программу и, проанализировав с помощью команды  `$?` , выдать сообщение о том, какое число было введено

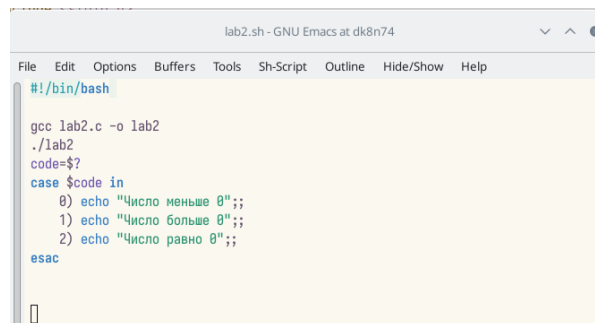


```

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

```

Рис. 4.4: код с



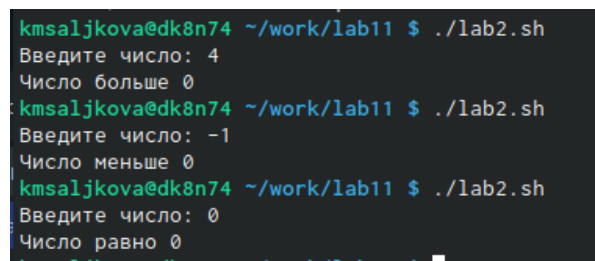
```

#!/bin/bash

gcc lab2.c -o lab2
./lab2
code=$?
case $code in
  0) echo "Число меньше 0";;
  1) echo "Число больше 0";;
  2) echo "Число равно 0";;
esac

```

Рис. 4.5: код sh



```

kmsaljkova@dk8n74 ~/work/lab11 $ ./lab2.sh
Введите число: 4
Число больше 0
kmsaljkova@dk8n74 ~/work/lab11 $ ./lab2.sh
Введите число: -1
Число меньше 0
kmsaljkova@dk8n74 ~/work/lab11 $ ./lab2.sh
Введите число: 0
Число равно 0

```

Рис. 4.6: проверка

3. Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до  $\infty$  (например 1.tmp, 2.tmp, 3.tmp, 4.tmp и т.д.). Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же командный файл должен уметь удалять все созданные им файлы (если они существуют).



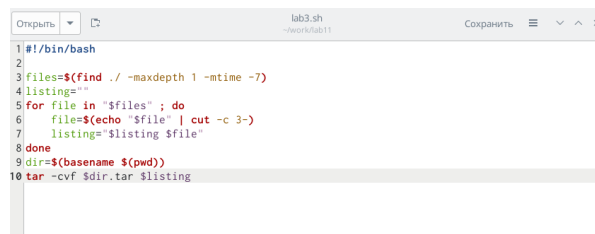
```

1 #!/bin/bash
2
3 opt=$1;
4 form=$2;
5 num=$3;
6 function Files() {
7     for ((i=1; i<=num; i++)) do
8         file=$(echo $form | tr '#' '$i')
9         if [ $opt == "-r" ]
10            then
11                rm -f $file
12            elif [ $opt == "-c" ]
13                then
14                    touch $file
15            fi
16        done
17    }
18 Files

```

Рис. 4.7: код3

4. Написать командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории. Модифицировать его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовать команду find).



```

1 #!/bin/bash
2
3 files=$(find ./ -maxdepth 1 -mtime -7)
4 listing=""
5 for file in "$files" ; do
6     file=$(echo "$file" | cut -c 3-)
7     listing="$listing $file"
8 done
9 dir=$(basename $(pwd))
10 tar -cvf $dir.tar $listing

```

Рис. 4.8: код4

## 5 Выводы

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

## **Список литературы**