

Отчет по лабораторной работе №11

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

Максим Сергеевич Белов

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	8
4	Выполнение лабораторной работы	9
5	Выводы	13
6	Контрольные вопросы	14

Список иллюстраций

4.1	Скрипт 1	9
4.2	Проверка скрипта 1	10
4.3	Программа на си	10
4.4	Командный файл 2.2	10
4.5	Проверка скрипта 2	11
4.6	Скрипт 3	11
4.7	Проверка скрипта 3	12
4.8	Скрипт 4	12
4.9	Проверка скрипта 4	12

List of Tables

3.1 Описание команд для работы с командными файлами 8

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:
 - `-i inputfile` - прочитать данные из указанного файла;
 - `-o outputfile` - вывести данные в указанный файл;
 - `-p шаблон` - 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 N (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 `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).

3 Теоретическое введение

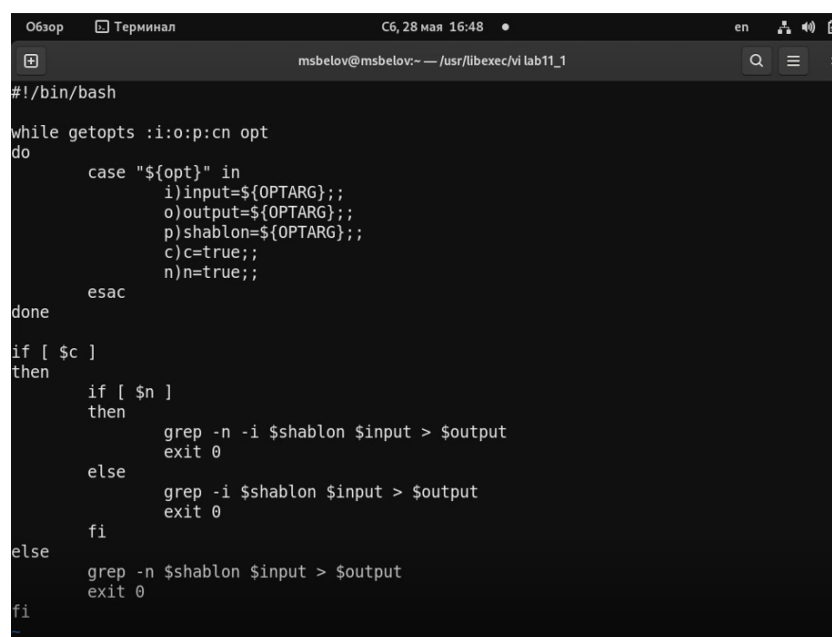
В табл. 3.1 приведено краткое описание команд для работы с командными файлами.

Таблица 3.1: Описание команд для работы с командными файлами

Команда	Описание
getopts	Осуществляет синтаксический анализ командной строки, выделяя флаги, и используется для объявления переменных.

4 Выполнение лабораторной работы

1. Скрипт 1 (Рис. 4.1)



```
#!/bin/bash

while getopts :i:o:p:cn opt
do
    case "${opt}" in
        i) input=${OPTARG};;
        o) output=${OPTARG};;
        p) shablon=${OPTARG};;
        c)c=true;;
        n)n=true;;
    esac
done

if [ $c ]
then
    if [ $n ]
    then
        grep -n -i $shablon $input > $output
        exit 0
    else
        grep -i $shablon $input > $output
        exit 0
    fi
else
    grep -n $shablon $input > $output
    exit 0
fi
```

Рис. 4.1: Скрипт 1

Сделаем файл исполняемым и проверим его работу (Рис. 4.2)

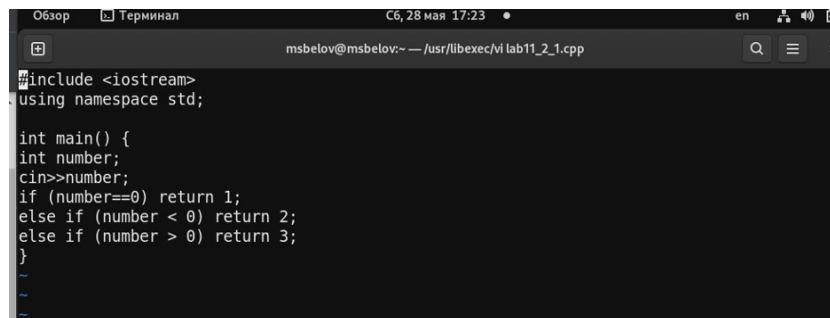


```
msbelov@msbelov:~$ vi lab11_1
msbelov@msbelov:~$ chmod +x lab11_1
msbelov@msbelov:~$ touch test1.txt
msbelov@msbelov:~$ mc

msbelov@msbelov:~$ ./lab11_1 -i test1.txt -o output.txt -p "text" -n -c
msbelov@msbelov:~$ cat output.txt
2:text
3:TEXT
msbelov@msbelov:~$ ./lab11_1 -i test1.txt -o output.txt -p "text" -n
msbelov@msbelov:~$ cat output.txt
2:text
msbelov@msbelov:~$
```

Рис. 4.2: Проверка скрипта 1

2. Программа 2.1 (C++) (Рис. 4.3)




```
msbelov@msbelov:~$ vi lab11_2_1.cpp
#include <iostream>
using namespace std;

int main() {
    int number;
    cin>>number;
    if (number==0) return 1;
    else if (number < 0) return 2;
    else if (number > 0) return 3;
}
```

Рис. 4.3: Программа на си

Командный файл 2.2 (Рис. 4.4)



```
msbelov@msbelov:~$ vi lab11_2_2
#!/bin/bash

echo "Vvedite chislo"
./lab11_2_1

case $? in
    1) echo "Chislo ravno 0";;
    2) echo "Chislo menshe 0";;
    3) echo "Chislo bolshe 0";;
esac
```

Рис. 4.4: Командный файл 2.2

Проверка работы командного файла (Рис. 4.5)



```
msbelov@msbelov:~$ g++ lab11_2_1.cpp -o lab11_2_1
msbelov@msbelov:~$ chmod +x lab11_2_2
msbelov@msbelov:~$ ./lab11_2_2
Vvedite chislo
0
Chislo ravno 0
msbelov@msbelov:~$ ./lab11_2_2
Vvedite chislo
5
Chislo bolshe 0
msbelov@msbelov:~$ ./lab11_2_2
Vvedite chislo
-5
Chislo menshe 0
msbelov@msbelov:~$
```

Рис. 4.5: Проверка скрипта 2

3. Скрипт 3 (Рис. 4.6)



```
#!/bin/bash
while getopts c:d opt
do
case $opt in
c)n="$OPTARG"; for i in $(seq 1 $n); do touch "$i.tmp"; done;;
d)for i in $(find -name "*.tmp"); do rm $i; done;;
esac
done
```

Рис. 4.6: Скрипт 3

Сделаем файл исполняемым и проверим его работу
(Рис. 4.7)



```
msbelov@msbelov:~/lab11_3
[msbelov@msbelov lab11_3]$ ./lab11_3 5
[msbelov@msbelov lab11_3]$ ls
lab11_3
[msbelov@msbelov lab11_3]$ ./lab11_3 -c 5
[msbelov@msbelov lab11_3]$ ls
1.tmp 2.tmp 3.tmp 4.tmp 5.tmp lab11_3
[msbelov@msbelov lab11_3]$ ./lab 3 -d
bash: ./lab 3: Нет такого файла или каталога
[msbelov@msbelov lab11_3]$ ./lab11_3 -d
[msbelov@msbelov lab11_3]$ ls
lab11_3
[msbelov@msbelov lab11_3]$
```

Рис. 4.7: Проверка скрипта 3

4. Скрипт 4 (Рис. 4.8)



```
#!/bin/bash

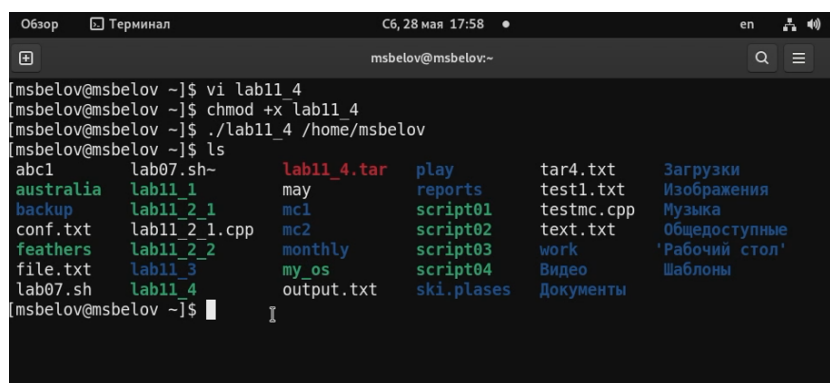
while getopts :d: opt
do
case $opt in
    d)dir="$OPTARG";;
esac
done

find $dir -mtime -7 -mtime +0 -type f > tar4.txt

tar -cvf lab11_4.tar -T tar4.txt
~
~
~
```

Рис. 4.8: Скрипт 4

Сделаем файл исполняемым и проверим его работу (Рис. 4.9)



```
msbelov@msbelov ~]$ vi lab11_4
msbelov@msbelov ~]$ chmod +x lab11_4
msbelov@msbelov ~]$ ./lab11_4 /home/msbelov
msbelov@msbelov ~]$ ls
abcl1      lab07.sh-      lab11_4.tar    play           tar4.txt      Загрузки
australia lab11_1        may            reports        test1.txt     Изображения
backup    lab11_2_1      mc1            script01       testmc.cpp    Музыка
conf.txt  lab11_2_1.cpp mc2            script02       text.txt      Общедоступные
feathers  lab11_2_2      monthly       script03       work          'Рабочий стол'
file.txt  lab11_3        my_os         script04       Видео
lab07.sh lab11_4        output.txt    ski.places     Документы
msbelov@msbelov ~]$
```

Рис. 4.9: Проверка скрипта 4

5 Выводы

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

6 Контрольные вопросы

1. Осуществляет синтаксический анализ командной строки, выделяя флаги, и используется для объявления переменных.
2. При перечислении имен файлов текущего каталога можно использовать некоторые символы.
3. for, case, if, while
4. break
5. true, :- всегда возвращает 0 в качестве кода выхода. false - всегда возвращает 1 в качестве кода выхода.
6. Введенная строка означает условие существования файла `mans/i.$s`
7. Если речь идет о 2-х параллельных действиях, то это while. когда мы показываем, что сначала делается 1-е действие. потом оно заканчивается при наступлении 2-го действия, применяем until.