

Лабораторная работа №11

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

Рытов Алексей Константинович НФИбд-02-21

Список иллюстраций

2.1	Скрипт 1.sh	5
2.2	Текстовый файл 1.txt	6
2.3	Запуск скрипта	6
2.4	Файл 1.txt после отработки скрипта	6
2.5	Программа 2.cpp на языке Си++	7
2.6	Скрипт 2.sh	8
2.7	Вывод скрипта	8
2.8	Скрипт 3.sh	9
2.9	Запуск скрипта	9
2.10	Результат отработки скрипта	10
2.11	Скрипт 4.sh	10
2.12	Запуск скрипта	11
2.13	Результат отработки скрипта	11

1 Цель работы

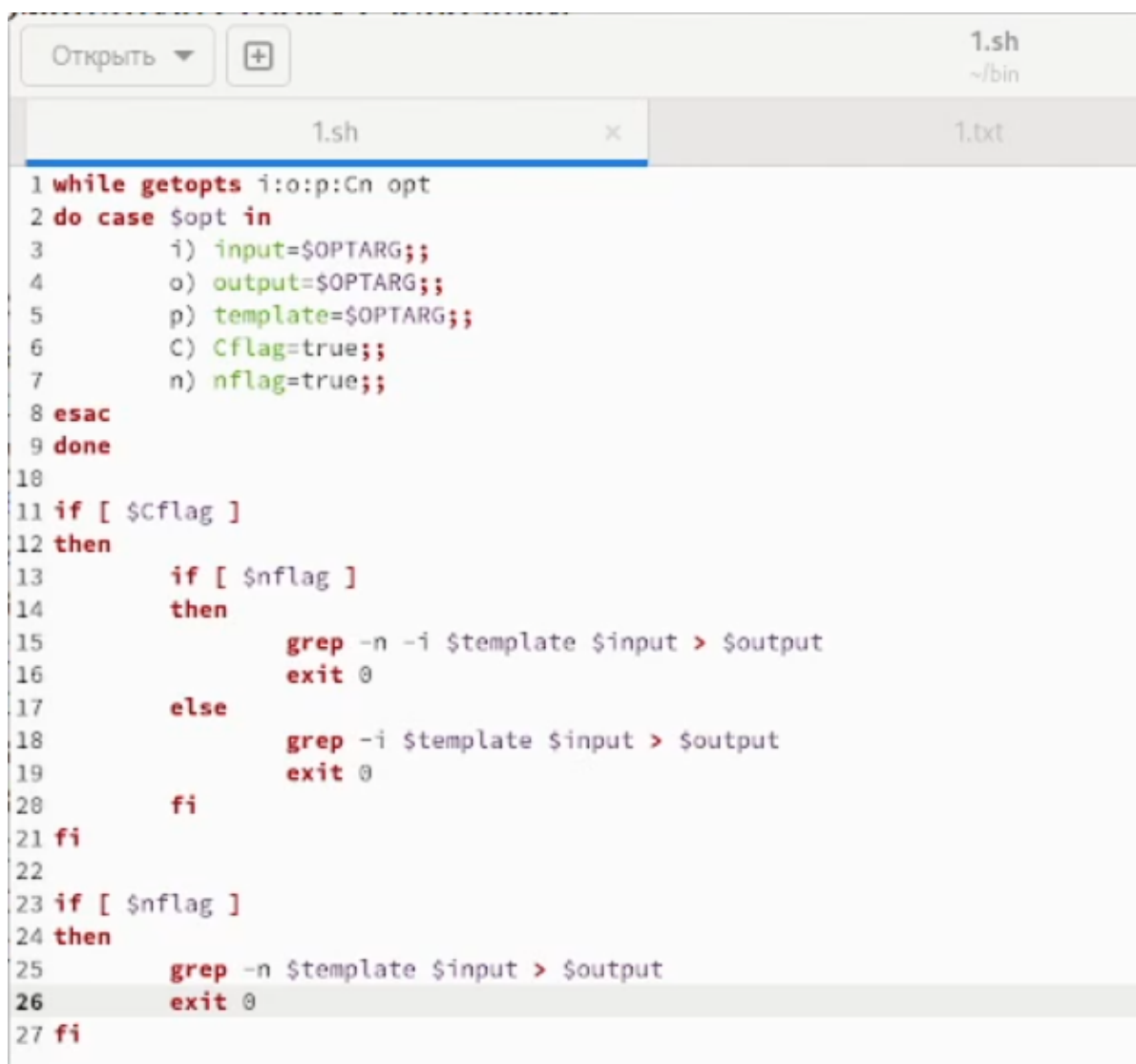
Изучить основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

2 Выполнение лабораторной работы

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

- `-iinputfile` — прочитать данные из указанного файла;
- `-ooutputfile` — вывести данные в указанный файл;
- `-ршаблон` — 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`. Затем написали текстовый файл для проверки скрипта(рис 1-4).



```
1 while getopts i:o:p:Cn opt
2 do case $opt in
3     i) input=$OPTARG;;
4     o) output=$OPTARG;;
5     p) template=$OPTARG;;
6     C) cflag=true;;
7     n) nflag=true;;
8 esac
9 done
10
11 if [ $cflag ]
12 then
13     if [ $nflag ]
14     then
15         grep -n -i $template $input > $output
16         exit 0
17     else
18         grep -i $template $input > $output
19         exit 0
20     fi
21 fi
22
23 if [ $nflag ]
24 then
25     grep -n $template $input > $output
26     exit 0
27 fi
```

Рис. 2.1: Скрипт 1.sh

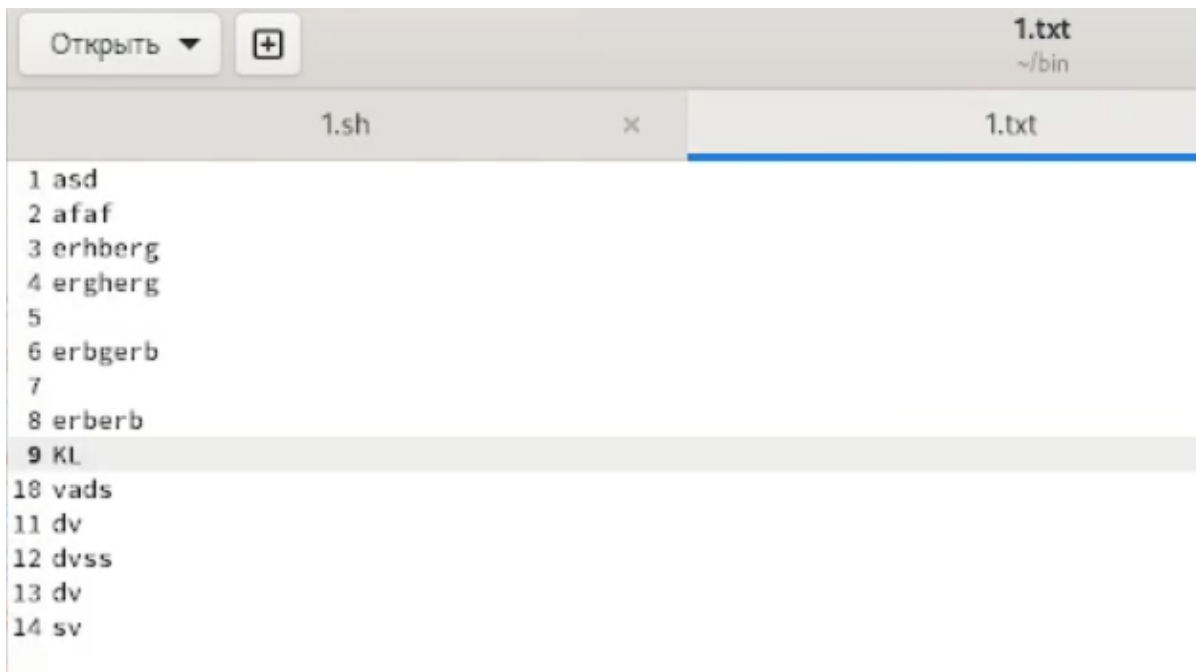


Рис. 2.2: Текстовый файл 1.txt

```
[akrihtov@fedora bin]$ ./1.sh -i 1.txt -o out.txt -p "KL" -C -n
[akrihtov@fedora bin]$
```

Рис. 2.3: Запуск скрипта



Рис. 2.4: Файл 1.txt после обработки скрипта

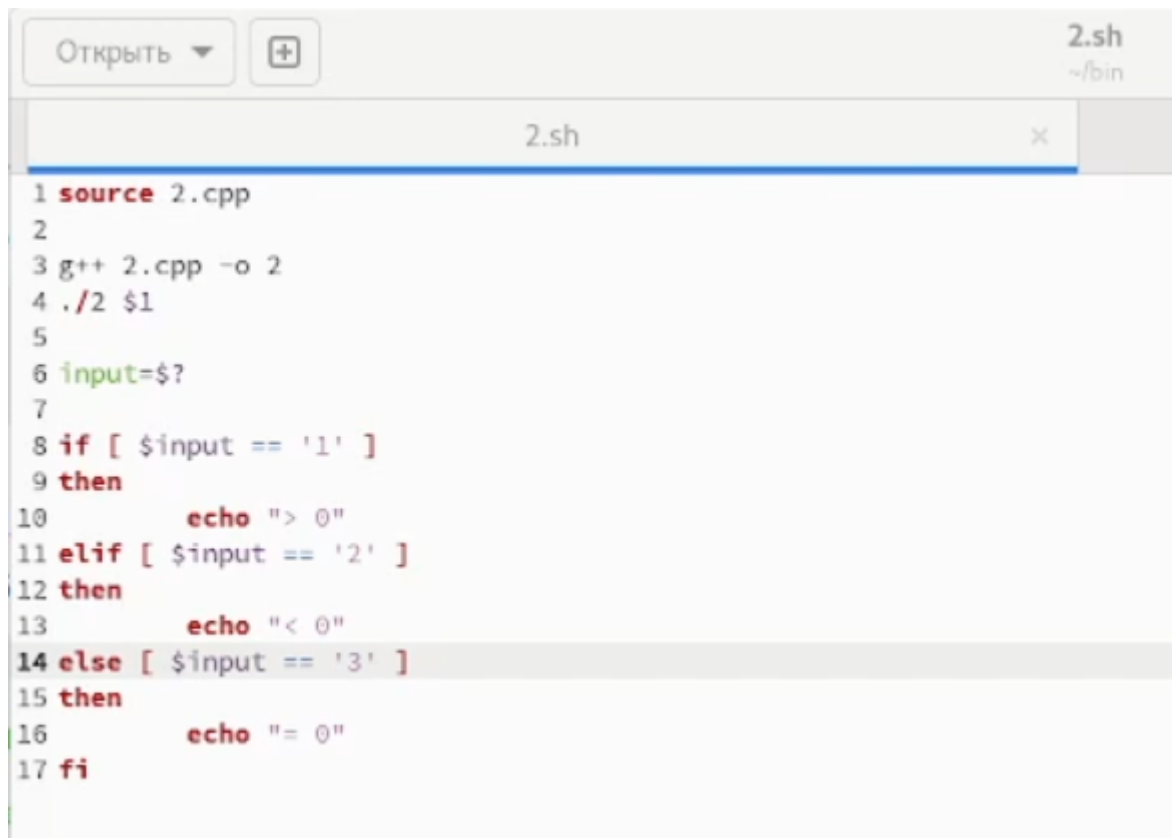
2. Написали на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем программа завершается с помощью функции `exit(n)`, передавая информацию в о коде завершения в оболочку. Командный файл должен вызывать эту программу

и, проанализировав с помощью команды \$?, выдать сообщение о том, какое число было введено(рис. 5-7).

A screenshot of a code editor window. The title bar shows 'Открыть' (Open) with a dropdown arrow, a '+' icon, and the filename '2.cpp' with the path '~/.bin'. Below the title bar, there's a tab labeled '2.sh' with a close button 'x'. The main area contains C++ code with line numbers 1 through 12 on the left. The code is as follows:

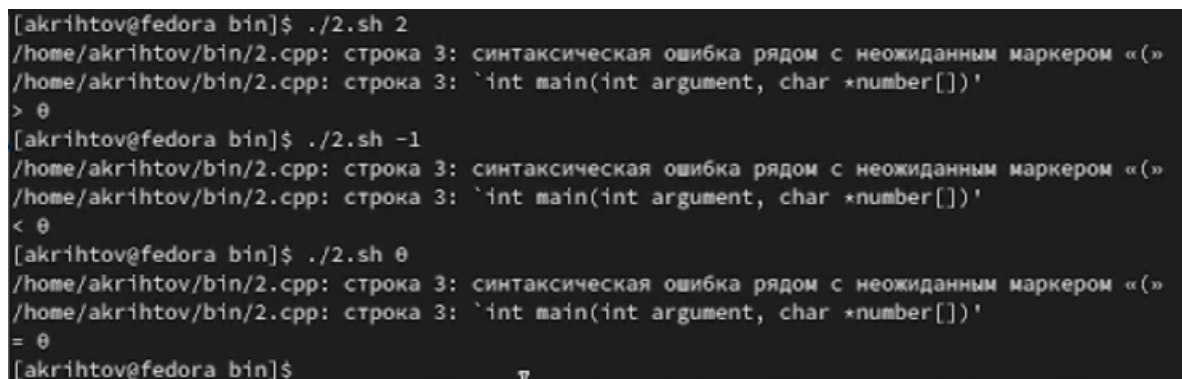
```
1 #include <iostream>
2
3 int main(int argument, char *number[])
4 {
5     if (atoi(number[1]) > 0)
6         exit(1);
7     else if (atoi(number[1]) < 0)
8         exit(2);
9     else exit(3);
10
11     return 0;
12 }
```

Рис. 2.5: Программа 2.cpp на языке Си++



```
1 source 2.cpp
2
3 g++ 2.cpp -o 2
4 ./2 $1
5
6 input=$?
7
8 if [ $input == '1' ]
9 then
10     echo "> 0"
11 elif [ $input == '2' ]
12 then
13     echo "< 0"
14 else [ $input == '3' ]
15 then
16     echo "= 0"
17 fi
```

Рис. 2.6: Скрипт 2.sh



```
[akrihtov@fedora bin]$ ./2.sh 2
/home/akrihtov/bin/2.cpp: строка 3: синтаксическая ошибка рядом с неожиданным маркером «(»
/home/akrihtov/bin/2.cpp: строка 3: `int main(int argument, char *number[])'
> 0
[akrihtov@fedora bin]$ ./2.sh -1
/home/akrihtov/bin/2.cpp: строка 3: синтаксическая ошибка рядом с неожиданным маркером «(»
/home/akrihtov/bin/2.cpp: строка 3: `int main(int argument, char *number[])'
< 0
[akrihtov@fedora bin]$ ./2.sh 0
/home/akrihtov/bin/2.cpp: строка 3: синтаксическая ошибка рядом с неожиданным маркером «(»
/home/akrihtov/bin/2.cpp: строка 3: `int main(int argument, char *number[])'
= 0
[akrihtov@fedora bin]$
```

Рис. 2.7: Вывод скрипта

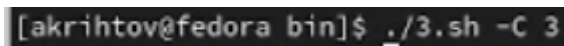
3. Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N (например 1.tmp, 2.tmp, 3.tmp, 4.tmp и т.д.). Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же командный файл должен уметь удалять все

созданные им файлы (если они существуют)(рис. 8-10).



```
1 while getopts C:D opt
2 do case $opt in
3     C) count=$OPTARG
4       for ((i = 1; i<=count; i++))
5       do
6           touch $i.tmp
7       done;;
8     D)
9       for ((i = 1; i<=count; i++))
10      do
11          rm -f $i.tmp
12      done;;
13 esac
14 done
```

Рис. 2.8: Скрипт 3.sh



```
[akrihtov@fedora bin]$ ./3.sh -C 3
```

Рис. 2.9: Запуск скрипта



Рис. 2.10: Результат отработки скрипта

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`)(рис. 11-13).

```
Открыть ▾ (+) 4.sh ~/bin
1.sh
1 while getopts C opt
2 do case $opt in
3   C) dir=$OPTARG
4     find $dir -mtime -7 -mtime +8 -type f > output.txt
5     tar -cvf archive.tar -T output.txt;;
6 esac
7 done
```

Рис. 2.11: Скрипт 4.sh

```
[akrihtov@fedora bin]$ ./4.sh -C ~  
./lab10/1.sh  
./lab10/2.sh  
./lab10/3.sh  
./lab10/4.sh  
[akrihtov@fedora bin]$
```

Рис. 2.12: Запуск скрипта



Рис. 2.13: Результат отработки скрипта

3 Выводы

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

4 Ответы на контрольные вопросы:

1. Каково предназначение команды `getopts`?

даёт возможность работать с флагами, которые вводятся при запуске скрипта
в консоль

2. Какое отношение метасимволы имеют к генерации имён файлов?

они помогают

3. Какие операторы управления действиями вы знаете?

`if`, `case`, `while`, `for`

4. Какие операторы используются для прерывания цикла?

`continue`; `break`

5. Для чего нужны команды `false` и `true`?

`true`,: - всегда возвращает 0 в качестве кода выхода. `false` - всегда возвращает 1
в качестве кода выхода.

6. Что означает строка `if test -f mans/i.$s`, встреченная в командном файле?

проверка существования файла

7. Объясните различия между конструкциями `while` и `until`.