

# **Лабораторная работа 11**

Сафин Андрей Алексеевич

# 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>

## Список иллюстраций

4.1	Первая программа . . . . .	8
4.2	Выполнение первой программы . . . . .	9
4.3	Вторая программа . . . . .	9
4.4	Выполнение второй программы . . . . .	10
4.5	Третья программа . . . . .	10
4.6	Выполнение третьей программы . . . . .	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**

Написать программы, описанные в лабораторной работе.

## 3 Теоретическое введение

bash (Bourne Again Shell) - это интерпретатор команд, позволяющий пользователю взаимодействовать с операционной системой через терминал. Bash способен считывать не только команды, вводимые в интерактивном режиме, но и файлы с командами. Эту функцию можно использовать для написания командных файлов.

## 4 Выполнение лабораторной работы

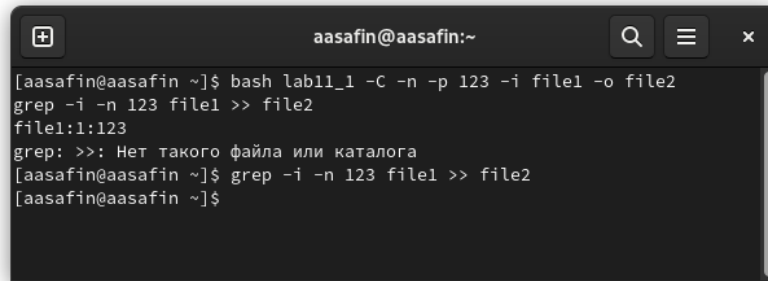
1. Создана программа, которая анализирует командную строку с ключами и в зависимости от них по-разному выполняет grep (рис. 4.1). Как видно, программа написана корректно - составленная команда grep выводится верно (рис. 4.2). Лишь по некой причине оболочка не считывает » как пере-направление вывода (что она делает при ручном введении той же команды).



```
command=grep
while getopts Cnp:iq: optletter
do case $optletter in
    C) command+= ' -i';;
    n) command+= ' -n';;
    p) pval=1 ; command+= ' ' ; command+= $OPTARG;;
    i) ival=1 ; command+= ' ' ; command+= $OPTARG;;
    q) oval=1 ; command+= ' >> ' ; command+= $OPTARG;;
    esac
done
if (($pval==1))
then if (($ival==1))
then echo $command
$command
fi
fi
```

Рис. 4.1: Первая программа

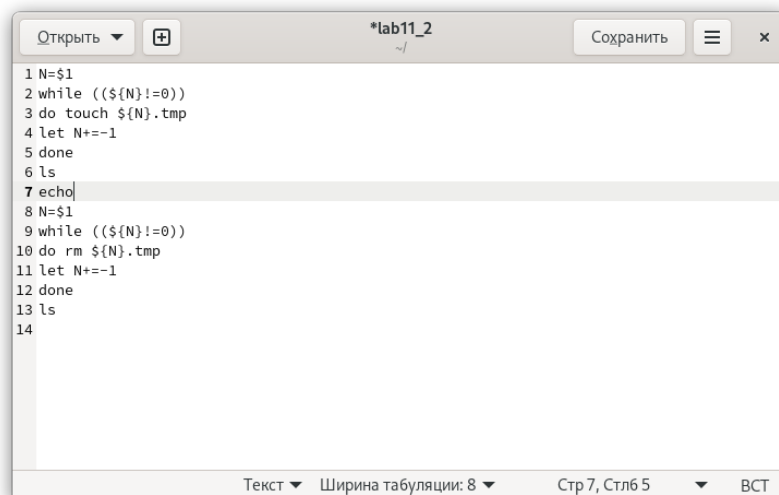




```
aasafin@aasafin:~  
[aasafin@aasafin ~]$ bash lab11_1 -C -n -p 123 -i file1 -o file2  
grep -i -n 123 file1 >> file2  
file1:1:123  
grep: >>: Нет такого файла или каталога  
[aasafin@aasafin ~]$ grep -i -n 123 file1 >> file2  
[aasafin@aasafin ~]$
```

Рис. 4.2: Выполнение первой программы

2. Язык Си автору лабораторной работы не известен.
3. Написана программа, создающая файлы вида  $n.tar$ ,  $1 \leq n \leq N$ , а затем удаляющая их же (рис. 4.3). Программа работает корректно (рис. 4.4).



```
*lab11_2  
~/  
Сохранить  
1 N=1  
2 while (($N!=0))  
3 do touch ${N}.tmp  
4 let N+=-1  
5 done  
6 ls  
7 echo  
8 N=1  
9 while (($N!=0))  
10 do rm ${N}.tmp  
11 let N+=-1  
12 done  
13 ls  
14  
Текст ▾ Ширина табуляции: 8 ▾ Стр 7, Стлб 5 ▾ ВСТ
```

Рис. 4.3: Вторая программа

```

aasafin@aasafin:~
aasafin@aasafin:~ — gedit lab1... x aasafin@aasafin:~ x
[aasafin@aasafin ~]$ bash lab11_2 3
1.tmp file2 lab10_4 lab9 Видео Музыка
2.tmp lab10_1 lab11_1 lab9~ Документы Общедоступные
3.tmp lab10_2 lab11_2 text.txt Загрузки 'Рабочий стол'
file1 lab10_3 lab11_3 work Изображения Шаблоны

file1 lab10_3 lab11_3 work Изображения Шаблоны
file2 lab10_4 lab9 Видео Музыка
lab10_1 lab11_1 lab9~ Документы Общедоступные
lab10_2 lab11_2 text.txt Загрузки 'Рабочий стол'
[aasafin@aasafin ~]$

```

Рис. 4.4: Выполнение второй программы

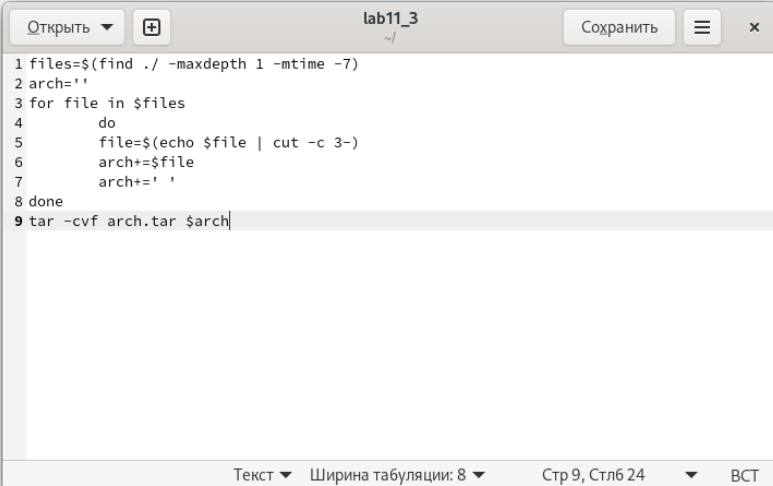
4. Создана программа, запаковывающая в архив все файлы из данной директории, которые редактировались в течение недели (рис. 4.5). Программа выполнена (рис. 4.6).

```

aasafin@aasafin:~
aasafin@aasafin:~ — gedit lab1... x aasafin@aasafin:~ x
[aasafin@aasafin ~]$ bash lab11_3
.vboxclient-clipboard.pid
.vboxclient-seamless.pid
.vboxclient-draganddrop.pid
file1
file2
lab11_1
.lessshst
lab11_3
.bash_history
lab11_2
[aasafin@aasafin ~]$ ls
arch.tar lab10_2 lab11_2 text.txt Загрузки 'Рабочий стол'
file1 lab10_3 lab11_3 work Изображения Шаблоны
file2 lab10_4 lab9 Видео Музыка
lab10_1 lab11_1 lab9~ Документы Общедоступные
[aasafin@aasafin ~]$

```

Рис. 4.5: Третья программа



The image shows a terminal window with a title bar containing 'lab11\_3' and standard window controls. The terminal displays a shell script with the following lines:

```
1 files=$(find ./ -maxdepth 1 -mtime -7)
2 arch=''
3 for file in $files
4 do
5     file=$(echo $file | cut -c 3-)
6     arch+=$file
7     arch+=' '
8 done
9 tar -cvf arch.tar $arch
```

The script finds files in the current directory, iterates through them, and creates a tar archive named 'arch.tar'. The terminal window has a status bar at the bottom with the text 'Текст', 'Ширина табуляции: 8', 'Стр 9, Стлб 24', and 'ВСТ'.

Рис. 4.6: Выполнение третьей программы

## 5 Выводы

Описанные командные файлы созданы, навык работы с bash получен.