

# **Отчёта по лабораторной работе №11.**

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

Гандич Дарья Владимировна. НБИбд-02-22.

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Вывод</b>	<b>15</b>

## Список иллюстраций

2.1	1 создание файлов . . . . .	6
2.2	1 файл txt . . . . .	7
2.3	1 код/файл sh . . . . .	8
2.4	1 проверка . . . . .	9
2.5	2 создание файлов/проверка . . . . .	10
2.6	2 код/файл sh . . . . .	10
2.7	2 код/файл с . . . . .	11
2.8	3 код/файл sh . . . . .	12
2.9	3 проверка . . . . .	12
2.10	3 создание файлов . . . . .	12
2.11	3 проверка наличия файлов . . . . .	13
2.12	3 удаление файлов . . . . .	13
2.13	4 код/файл sh . . . . .	14
2.14	4 проверка . . . . .	14
2.15	4 проверка наличия архива . . . . .	14

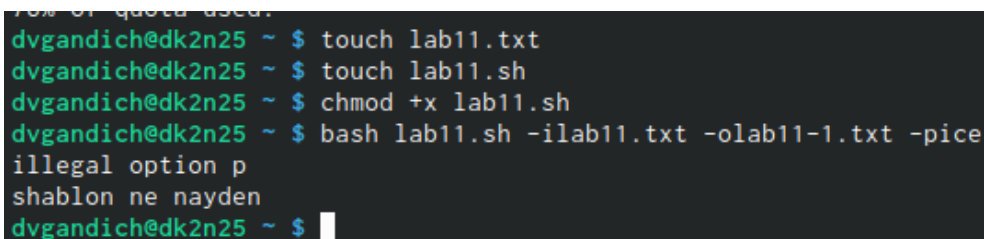
## Список таблиц

# 1 Цель работы

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

## 2 Выполнение лабораторной работы

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`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. Создаем файлы форматов `txt` и `sh`, в файл `txt` заносим любой текст из интернета, а в файл `sh` пишем специальный код, который поможет сделать файл исполняемым



```
70% of quota used.  
dvgandich@dk2n25 ~ $ touch lab11.txt  
dvgandich@dk2n25 ~ $ touch lab11.sh  
dvgandich@dk2n25 ~ $ chmod +x lab11.sh  
dvgandich@dk2n25 ~ $ bash lab11.sh -ilab11.txt -olab11-1.txt -pice  
illegal option p  
shablon ne nayden  
dvgandich@dk2n25 ~ $
```

Рис. 2.1: 1 создание файлов

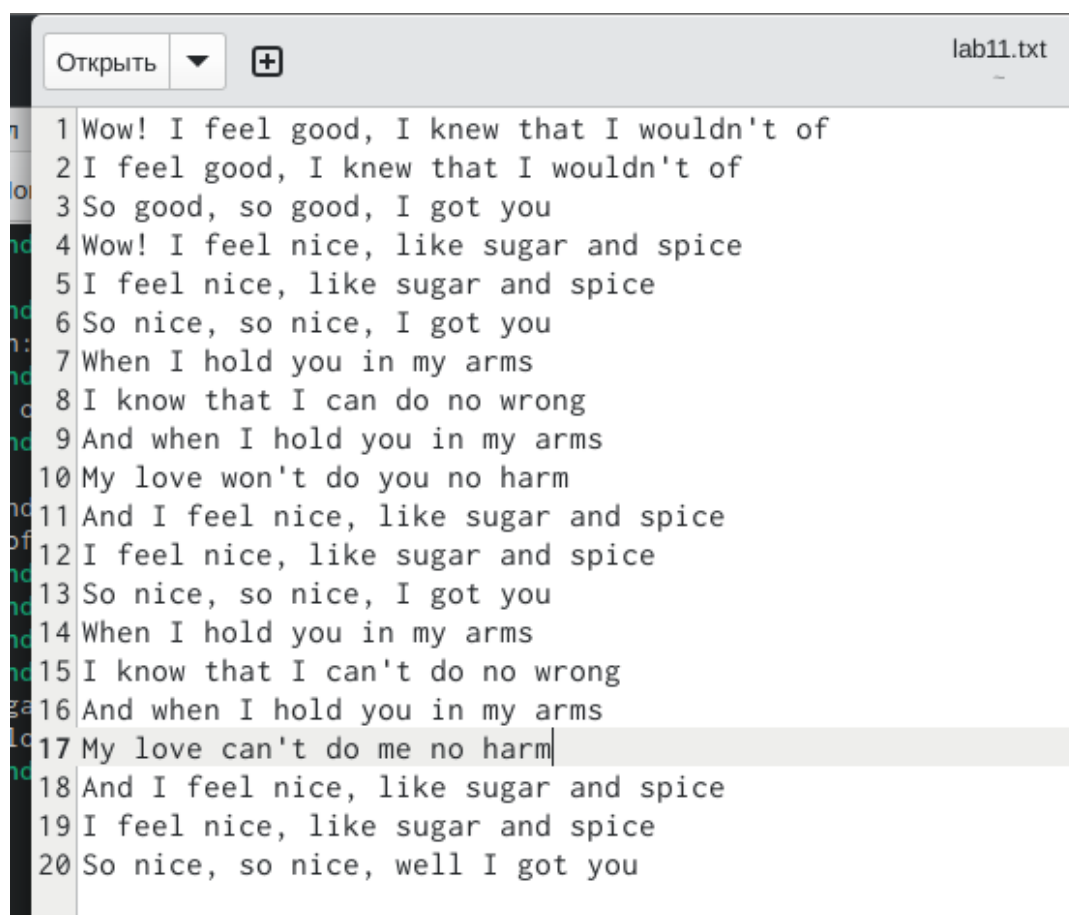


Рис. 2.2: 1 файл txt

Приложения Места GNU Emacs

File Edit Options Buffers Tools Sh-Script Outline Hide/Show Help

```
#!/bin/bash
iflag=0; oflag=0; pflag=0; Cflag=0; nflag=0;
while getopts i:o:p:C:n optletter
do case $optletter in
    i) iflag=1; ival=$OPTARG;;
    o) oflag=1; oval=$OPTARG;;
    i) pflag=1; pval=$OPTARG;;
    C) Cflag=1;;
    n) nflag=1;;
    *) echo illegal option $optletter
esac
done
if (($pflag=0))
then echo "shablon ne nayden"
else
    if (($iflag=0))
    then echo "fail ne nayden"
    else
        if (($oflag=0))
        then if (($Cflag=0))
            then if (($nflag=0))
                then grep $pval $ival
                else grep -n $pval $ival
            fi
            else if (($nflag=0))
                then grep -i $pval $ival
                else grep -i -n $pval $ival
            fi
        fi
        else if (($Cflag=0))
            then if (($nflag=0))
                then grep $pval $ival > $oval
                else grep -n $pval $ival > $oval
            fi
            else if (($nflag=0))
                then grep -i $pval $ival > $oval
                else grep -i -n $pval $ival > $oval
            fi
        fi
    fi
fi
fi
```

Рис. 2.3: 1 код/файл sh



### 3. Проверяем, что все сработало

```
78% of quota used.
dvgandich@dk2n25 ~ $ touch lab11.txt
dvgandich@dk2n25 ~ $ touch lab11.sh
dvgandich@dk2n25 ~ $ chmod +x lab11.sh
dvgandich@dk2n25 ~ $ bash lab11.sh -ilab11.txt -olab11-1.txt -pice
illegal option p
shablon ne nayden
dvgandich@dk2n25 ~ $ cat ~/lab11.txt
Wow! I feel good, I knew that I wouldn't of
I feel good, I knew that I wouldn't of
So good, so good, I got you
Wow! I feel nice, like sugar and spice
I feel nice, like sugar and spice
So nice, so nice, I got you
When I hold you in my arms
I know that I can do no wrong
And when I hold you in my arms
My love won't do you no harm
And I feel nice, like sugar and spice
I feel nice, like sugar and spice
So nice, so nice, I got you
When I hold you in my arms
I know that I can't do no wrong
And when I hold you in my arms
My love can't do me no harm
And I feel nice, like sugar and spice
I feel nice, like sugar and spice
So nice, so nice, well I got you
dvgandich@dk2n25 ~ $ ./lab11.sh -i ~/lab11.txt -o ~/lab11-1.txt -p it -C -n
illegal option p
shablon ne nayden
dvgandich@dk2n25 ~ $ cat ~/lab11-1.txt
cat: /afs/.dk.sci.pfu.edu.ru/home/d/v/dvgandich/lab11-1.txt: Нет такого файла или каталога
dvgandich@dk2n25 ~ $
```

Рис. 2.4: 1 проверка

4. 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
5. Повторяем все действия, что и делали ранее, только теперь создаем файл `sh` и `c`

```
dvgandich@dk2n25 ~ $ touch prog1.c prog1.sh
dvgandich@dk2n25 ~ $ chmod +x *.sh
dvgandich@dk2n25 ~ $ ./prog1.sh
Vvedite chislo: 7
Chislo bol'she 0
dvgandich@dk2n25 ~ $
```

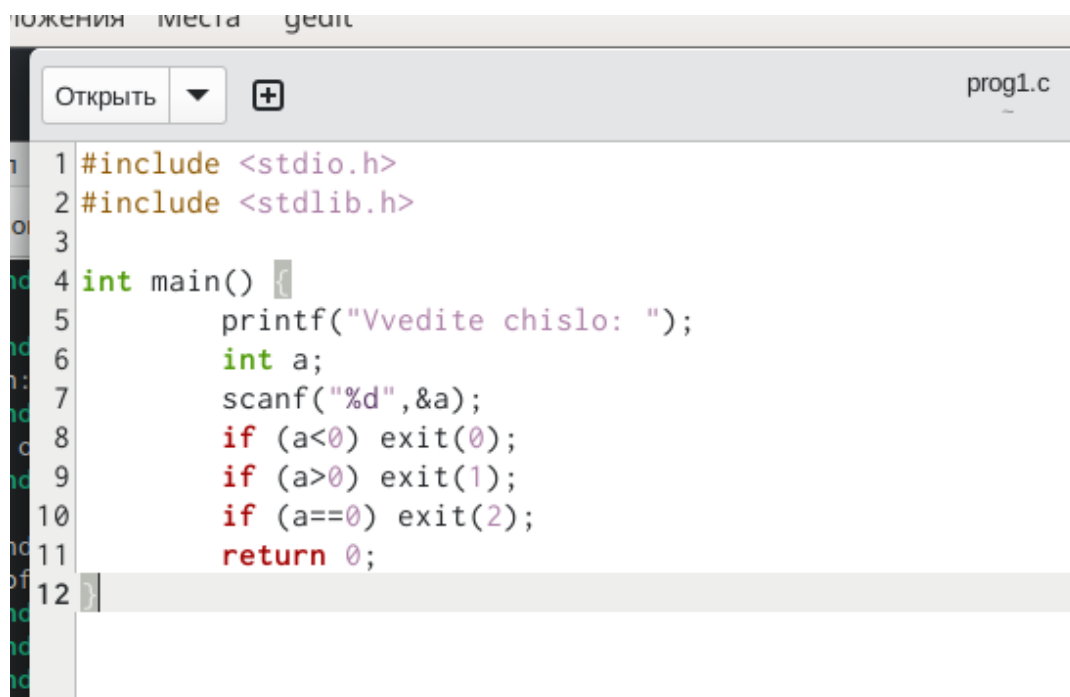
Рис. 2.5: 2 создание файлов/проверка



```
#!/bin/bash

gcc prog1.c -o prog1
./prog1
code=$?
case $code in
    0) echo "Chislo men'she 0";;
    1) echo "Chislo bol'she 0";;
    2) echo "Chislo ravno 0";;
esac
```

Рис. 2.6: 2 код/файл sh



```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main() {
5     printf("Vvedite chislo: ");
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. 2.7: 2 код/файл с

6. Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до ☒ (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).
7. Делаем вновь то же самое, теперь создаем файл только формата sh и в качестве проверки должны создаться три файла, затем мы их удаляем

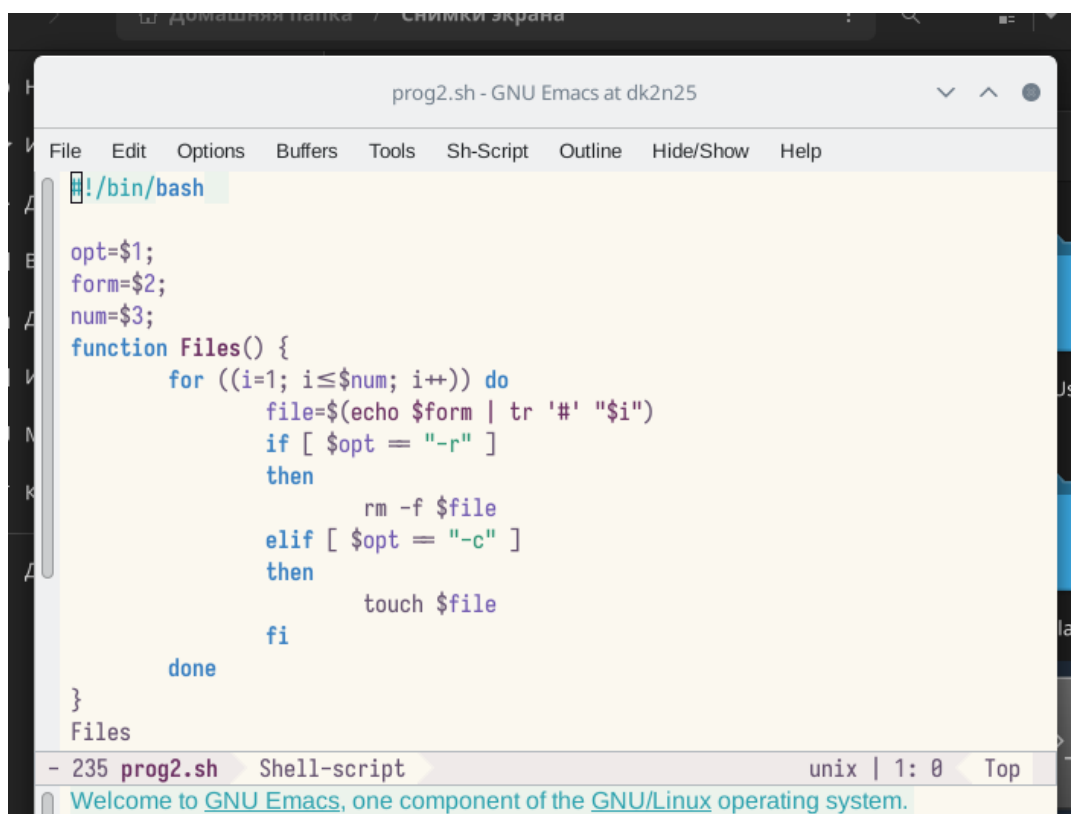


Рис. 2.8: 3 код/файл sh

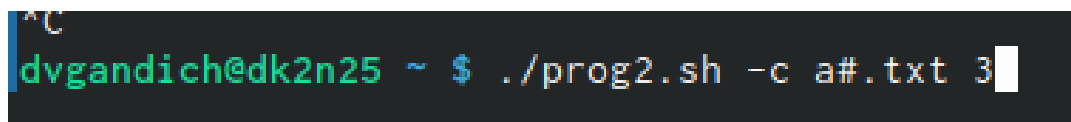


Рис. 2.9: 3 проверка

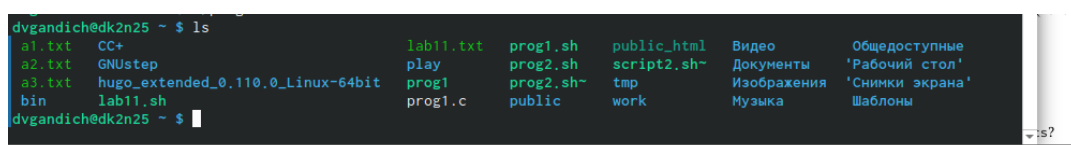


Рис. 2.10: 3 создание файлов

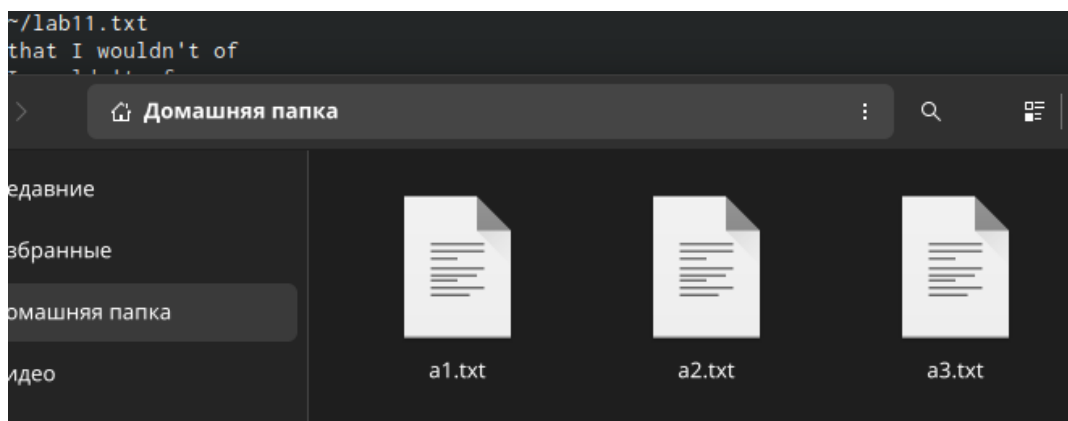


Рис. 2.11: 3 проверка наличия файлов

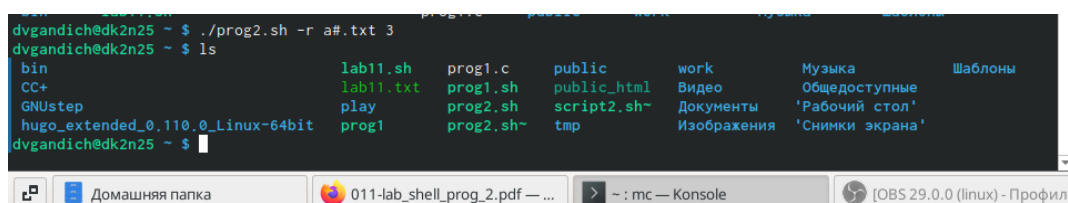


Рис. 2.12: 3 удаление файлов

8. 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`).
9. Делаем все то же самое, проверкой будет созданный архив формата `tar`

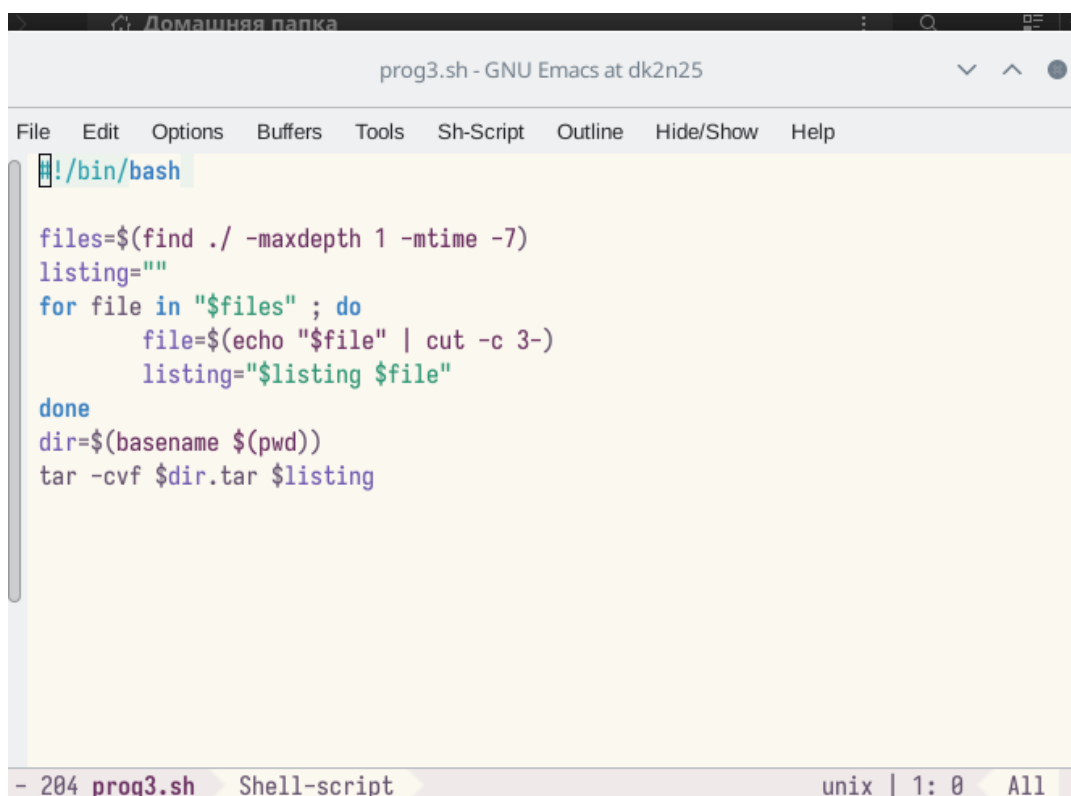


Рис. 2.13: 4 код/файл sh

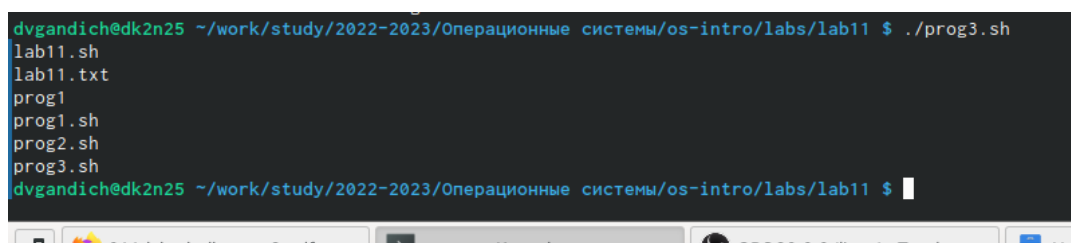


Рис. 2.14: 4 проверка

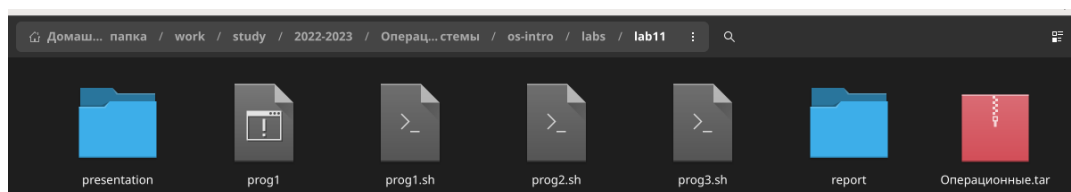


Рис. 2.15: 4 проверка наличия архива

## 3 Вывод

Мы изучили основы программирования в оболочке Linux, научились писать более сложные командные файлы, используя разные управляющие конструкции и циклы