

# Лабораторная работа №11

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

---

Гандич Дарья Владимировна

20 апреля 2023 г.

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Гандич Дарья Владимировна
- студентка группы НБИбд-02-22
- Российский университет дружбы народов

## Вводная часть

---

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

Содержание исследования

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;
  - `-r`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 `-r`.

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. 1: 1 создание файлов

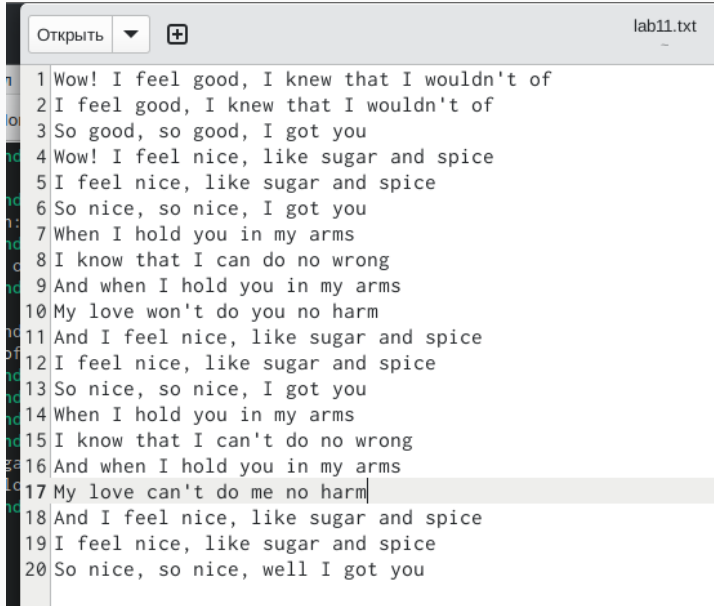


Рис. 2: 1 файл txt

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:op:fi:n optletter
do case $optletter in
    i) iflag=1; ival=$OPTARG;;
    o) oflag=1; oval=$OPTARG;;
    p) pflag=1; pval=$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. 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 ~ $
```

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 с

```
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. 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. 6: 2 код/файл sh

пожения места ген

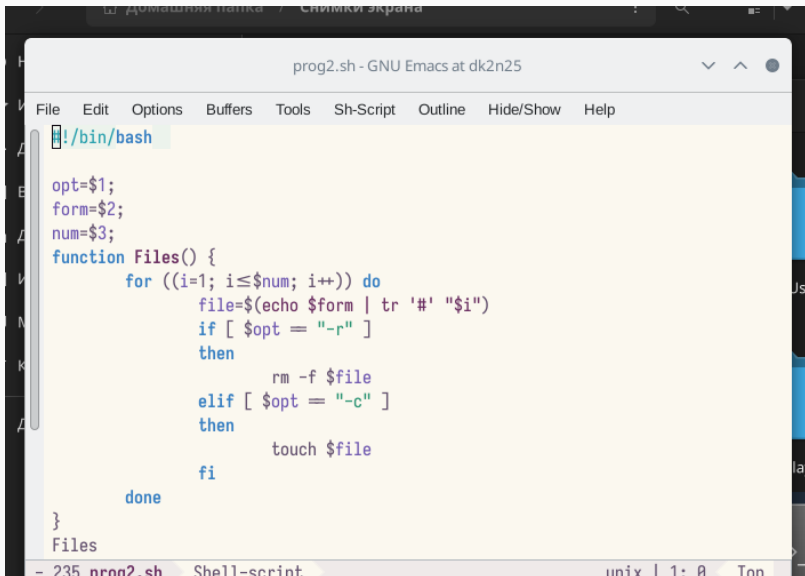
Открыть ▼ + prog1.c

```
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. 7: 2 код/файл с

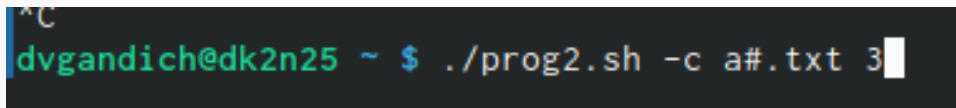
6. Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до `%d` (например 1.tmp, 2.tmp, 3.tmp,4.tmp и т.д.). Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же командный файл должен уметь удалять все созданные им файлы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

7. Делаем вновь то же самое, теперь создаем файл только формата sh и в качестве проверки должны создаваться три файла, затем мы их удаляем



```
prog2.sh - GNU Emacs at dk2n25
File Edit Options Buffers Tools Sh-Script Outline Hide/Show Help
#!/bin/bash

opt=$1;
form=$2;
num=$3;
function Files() {
    for ((i=1; i≤$num; i++)) do
        file=$(echo $form | tr '#' "$i")
        if [ $opt = "-r" ]
        then
            rm -f $file
        elif [ $opt = "-c" ]
        then
            touch $file
        fi
    done
}
Files
Files
```

A terminal window with a dark background. The prompt is 'dvgandich@dk2n25 ~ \$'. The command being entered is './prog2.sh -c a#.txt 3'. A cursor is at the end of the command. Above the prompt, there is a faint, partially visible command '^C'.

```
^C  
dvgandich@dk2n25 ~ $ ./prog2.sh -c a#.txt 3
```

Рис. 9: 3 проверка



```
dvgandich@dk2n25 ~ $ ls
a1.txt      CC+          lab11.txt    prog1.sh     public_html  Видео        Общедоступные
a2.txt      GNUstep      play         prog2.sh     script2.sh~  Документы   'Рабочий стол'
a3.txt      hugo_extended_0.110.0_Linux-64bit  prog1        prog2.sh~    tmp          Изображения  'Снимки экрана'
bin         lab11.sh     prog1.c      public        work         Музыка       Шаблоны
dvgandich@dk2n25 ~ $
```

Рис. 10: 3 создание файлов

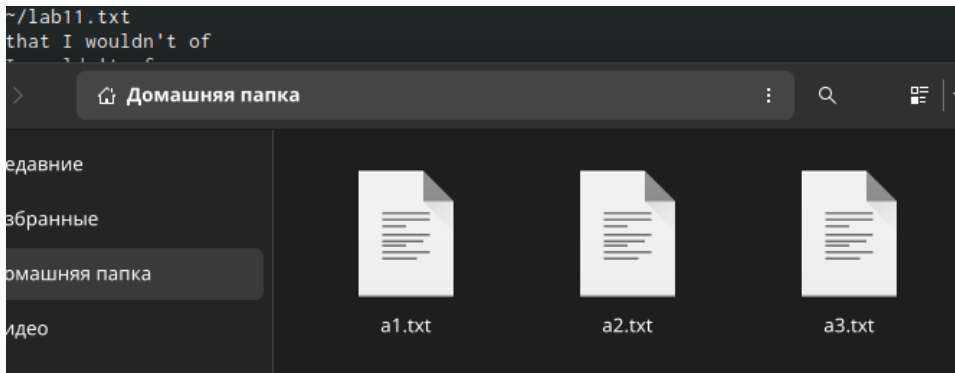


Рис. 11: 3 проверка наличия файлов

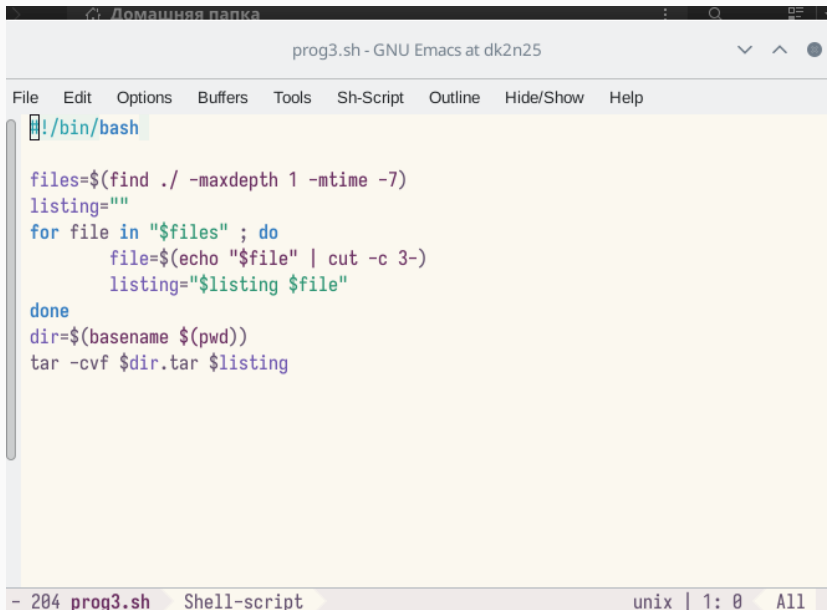
```
dvgandich@dk2n25 ~ $ ./prog2.sh -r a#.txt 3
dvgandich@dk2n25 ~ $ ls
bin                lab11.sh           prog1.c            public              work               Музыка             Шаблоны
CC+                lab11.txt          prog1.sh           public_html         Видео              Общедоступные
GNUstep            play              prog2.sh           script2.sh~         Документы          'Рабочий стол'
hugo_extended_0.110.0_Linux-64bit  prog1              prog2.sh~          tmp                 Изображения        'Снимки экрана'
dvgandich@dk2n25 ~ $
```

Домашняя папка    011-lab\_shell\_prog\_2.pdf — ...    ~ : mc — Konsole    [OBS 29.0.0 (linux) - Профил...

Рис. 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



```
#!/bin/bash

files=$(find ./ -maxdepth 1 -mtime -7)
listing=""
for file in "$files" ; do
    file=$(echo "$file" | cut -c 3-)
    listing="$listing $file"
done
dir=$(basename $(pwd))
tar -cvf $dir.tar $listing
```

The screenshot shows the Emacs editor interface. The title bar reads "prog3.sh - GNU Emacs at dk2n25". The menu bar includes "File", "Edit", "Options", "Buffers", "Tools", "Sh-Script", "Outline", "Hide/Show", and "Help". The script content is as follows:

```
#!/bin/bash

files=$(find ./ -maxdepth 1 -mtime -7)
listing=""
for file in "$files" ; do
    file=$(echo "$file" | cut -c 3-)
    listing="$listing $file"
done
dir=$(basename $(pwd))
tar -cvf $dir.tar $listing
```

The status bar at the bottom displays "- 204 prog3.sh Shell-script" on the left, "unix | 1: 0" in the center, and "All" on the right.

```
dvigandich@dk2n25 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab11 $ ./prog3.sh
lab11.sh
lab11.txt
prog1
prog1.sh
prog2.sh
prog3.sh
dvigandich@dk2n25 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab11 $
```

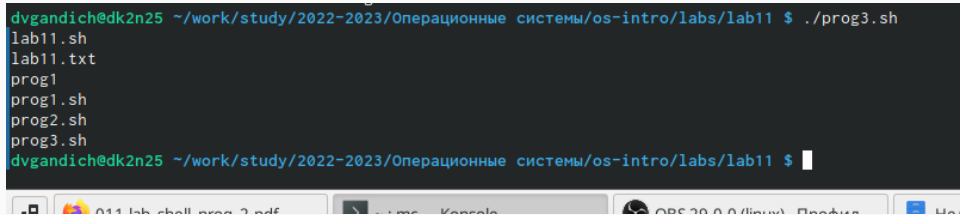
A screenshot of a terminal window with a dark background. The prompt is 'dvigandich@dk2n25' followed by the path '~/work/study/2022-2023/Операционные системы/os-intro/labs/lab11'. The user has entered './prog3.sh'. The output of the script is listed line by line: 'lab11.sh', 'lab11.txt', 'prog1', 'prog1.sh', 'prog2.sh', and 'prog3.sh'. The prompt is now ready for the next command. Below the terminal window, a portion of the desktop taskbar is visible, showing icons for a file manager, a PDF viewer (011 lab chall prog 3.pdf), a terminal (Kongole), and a web browser (CPE 20.0.0 (linux) Firefox).

Рис. 14: 4 проверка

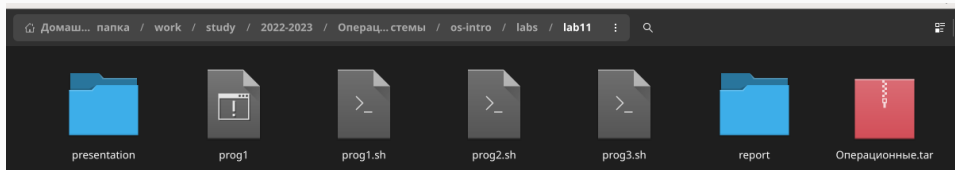


Рис. 15: 4 проверка наличия архива

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



мы молодцы! :::