

# Лабораторная работа №11

Операционные системы

---

Александрова Ульяна

22 апреля 2023

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Александрова Ульяна Вадимовна
- студентка 1-го курса
- Российский университет дружбы народов

## Цель работы

---

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

## Задание

---

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; – `-rшаблон` — 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. 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.



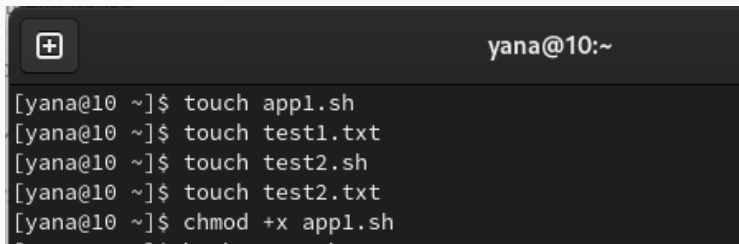
3. Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до `N` (например 1.tmp, 2.tmp, 3.tmp, 4.tmp и т.д.). Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же командный файл должен уметь удалять все созданные им файлы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

## Выполнение лабораторной работы

---

Создала файл app1.sh для первой программы и дополнительные текстовые файлы.

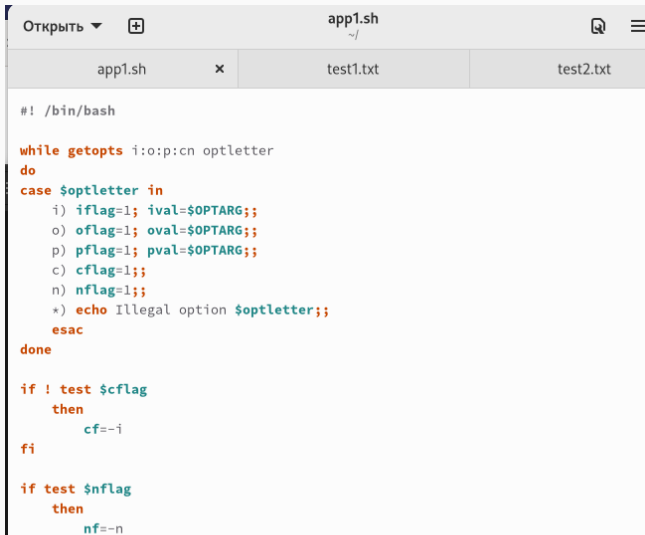
A terminal window with a dark background. The title bar shows a plus icon in a square and the text 'yana@10:~'. The terminal contains five lines of commands and their prompts: '[yana@10 ~]\$ touch app1.sh', '[yana@10 ~]\$ touch test1.txt', '[yana@10 ~]\$ touch test2.sh', '[yana@10 ~]\$ touch test2.txt', and '[yana@10 ~]\$ chmod +x app1.sh'.

```
yana@10:~  
[yana@10 ~]$ touch app1.sh  
[yana@10 ~]$ touch test1.txt  
[yana@10 ~]$ touch test2.sh  
[yana@10 ~]$ touch test2.txt  
[yana@10 ~]$ chmod +x app1.sh
```

Figure 1: Создание файла

# Выполнение лабораторной работы

Заполнила код.



```
Открыть ▾ + app1.sh ~/
app1.sh × test1.txt test2.txt

#!/bin/bash

while getopts i:o:p:cn 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 ! test $cflag
then
    cf=-i
fi

if test $nflag
then
    nf=-n
```

# Выполнение лабораторной работы

Запустила программу. Она работает.

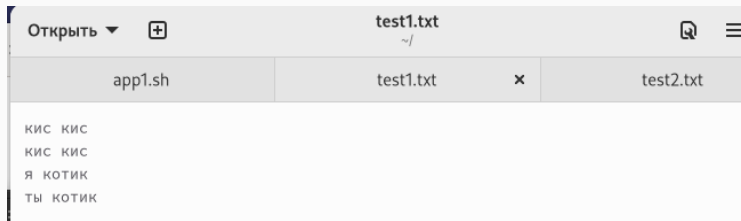


Figure 3: Текстовый файл 1

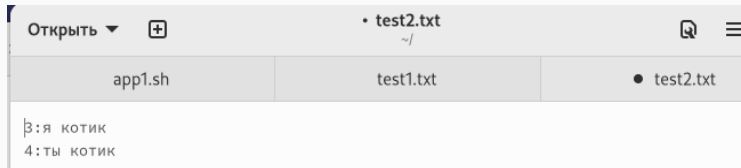



Figure 4: Текстовый файл 2

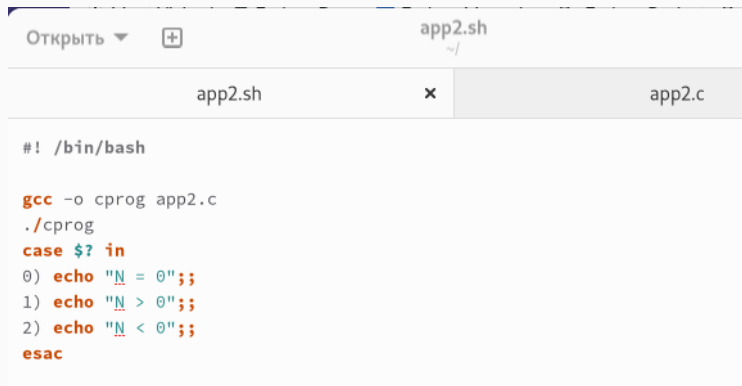
Я создала второй командный файл, а также дополнительный файл.

A terminal window with a dark background and light text. It shows three lines of commands being executed in a shell. The first line creates 'app2.sh', the second creates 'app2.c', and the third sets permissions for 'app2.sh'.

```
[yana@10 ~]$ touch app2.sh  
[yana@10 ~]$ touch app2.c  
[yana@10 ~]$ chmod +x app2.sh  
[yana@10 ~]$
```

Figure 5: Создание файла 2

Далее я написала тело обеих программ.

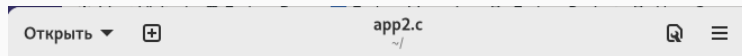


The screenshot shows a code editor window with two tabs: 'app2.sh' (active) and 'app2.c'. The 'app2.sh' tab contains the following shell script code:

```
#!/bin/bash

gcc -o cprog app2.c
./cprog
case $? in
0) echo "N = 0";;
1) echo "N > 0";;
2) echo "N < 0";;
esac
```


Figure 6: Тело 1



The bottom of the image shows the top of another code editor window with a tab labeled 'app2.c'. The 'Открыть' (Open) button and a plus icon are visible on the left.



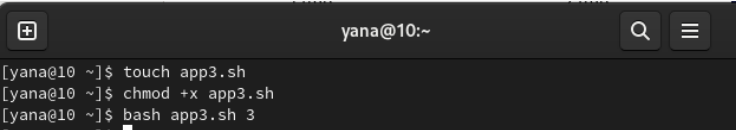
Проверила работу приложения.



```
[yana@10 ~]$ bash app2.sh
Enter the number: 12
N > 0
[yana@10 ~]$ bash app2.sh
Enter the number: 0
N = 0
[yana@10 ~]$ bash app2.sh
Enter the number: -2
N < 0
```

Figure 8: Проверка работы 2

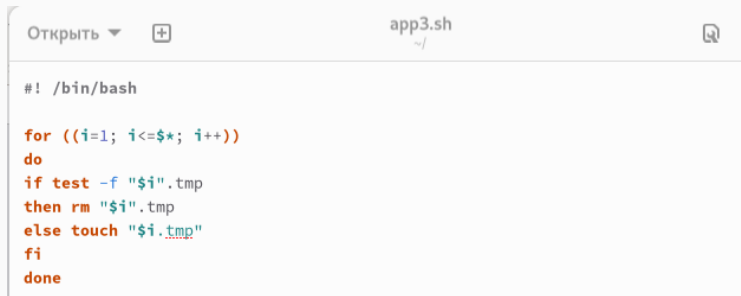
Создала третий файл app3.sh.

A terminal window with a dark background. The title bar shows a plus icon, the text 'yana@10:~', a search icon, and a menu icon. The terminal contains three lines of text: '[yana@10 ~]\$ touch app3.sh', '[yana@10 ~]\$ chmod +x app3.sh', and '[yana@10 ~]\$ bash app3.sh 3'.

```
yana@10:~  
[yana@10 ~]$ touch app3.sh  
[yana@10 ~]$ chmod +x app3.sh  
[yana@10 ~]$ bash app3.sh 3
```

Figure 9: Создание файла 3

Записала код программы.

A screenshot of a code editor window. The title bar shows 'app3.sh' and a file icon. The editor contains a shell script with the following content:

```
#!/bin/bash

for ((i=1; i<=$*; i++))
do
if test -f "$i".tmp
then rm "$i".tmp
else touch "$i.tmp"
fi
done
```

Figure 10: Код программы

## Выполнение лабораторной работы

Проверила работу кода. Новые файлы создались.

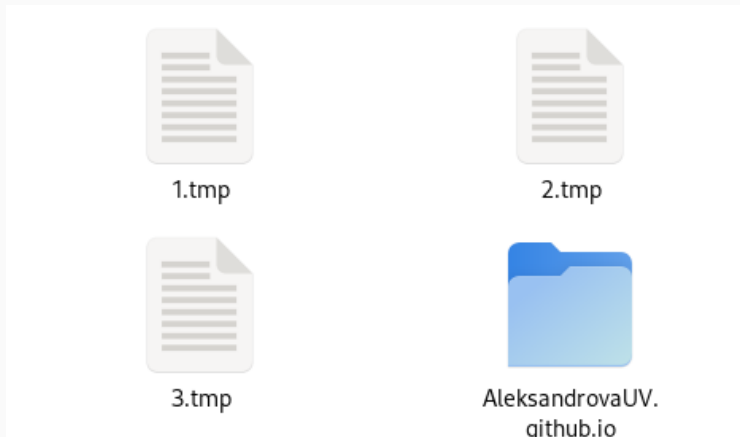


Figure 11: Проверка программы

Я создала последний командный файл app4.sh и заполнила файл.

A screenshot of a code editor window. The title bar at the top shows 'Открыть' with a dropdown arrow, a plus icon, and the filename 'app4.sh' with a tilde and slash indicating the current directory. The editor area contains three lines of text: the first line is a shebang '#! /bin/bash', the second line is a 'find' command 'find \$\* -mtime -7 -mtime +0 -type f > FILES.txt', and the third line is a 'tar' command 'tar -cf archive.tar -T FILES.txt|'.

```
Открыть ▼ + app4.sh  
~/  
  
#!/bin/bash  
  
find $* -mtime -7 -mtime +0 -type f > FILES.txt  
tar -cf archive.tar -T FILES.txt|
```

Figure 12: Создание файла 4

Далее запустила приложение. Оно работает исправно.

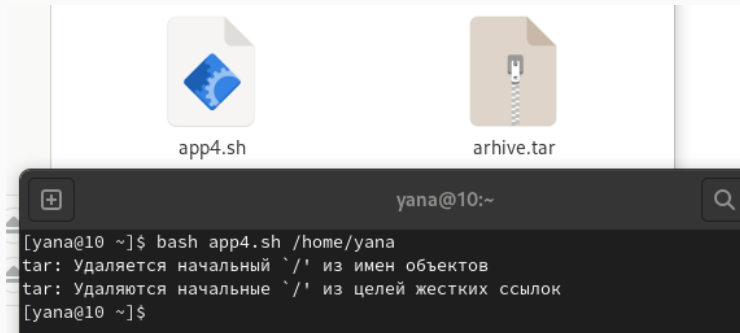


Figure 13: Проверка программы

## Выводы

---

Я изучила основы программирования в оболочке ОС UNIX. Научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

...