

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

---

Кеан Путхеаро НПИбд-01-20<sup>1</sup>

28 мая, 2021, Москва, Россия

<sup>1</sup>Российский Университет Дружбы Народов

# Цели и задачи работы

---

## Цель лабораторной работы

Изучить основы программирования в оболочке ОС UNIX.  
Научится писать бо-лее сложные командные файлы с использованием логических управляющих кон-струкций и циклов.

## Задачи лабораторной работы

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. Написать на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем программа завершается помощью функции `exit(n)`, передавая информацию в о коде завершения оболочку. Командный файл должен вызывать эту программу и, проанализировав с помощью 3/11 команды \$? выдать сообщение о том, какое число было

## Задачи лабораторной работы

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`).

# **Процесс выполнения лабораторной работы**

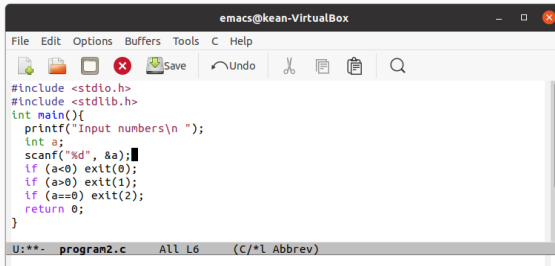
---

## Первый скрипт

```
Terminal - kean@kean-VirtualBox: ~  
File Edit View Terminal Tabs Help  
#!/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 $optletter  
esac  
done  
if (((cflag==1) && (nflag==1)))  
then grep -e${pval} -i -n ${ival}  
    if ((oflag==1)  
        then grep -e${pval} -i -n ${ival} > ${oval}  
        fi  
fi  
if (((cflag==1) && (nflag==0)))  
then grep -e${pval} -i ${ival}  
    if ((oflag==1)  
        then grep -e${pval} -i ${ival} > ${oval}  
        fi  
fi  
if (((cflag==0) && (nflag==1)))  
then grep -e${pval} -n ${ival}  
    if ((oflag==1)  
        then grep -e${pval} -n ${ival} > ${oval}  
        fi  
fi  
if (((cflag==0) && (nflag==0)))  
then grep -e${pval} ${ival}  
    if ((oflag==1)  
        then grep -e${pval} ${ival} > ${oval}  
        fi  
fi
```

:wq

# создание файла program2.c



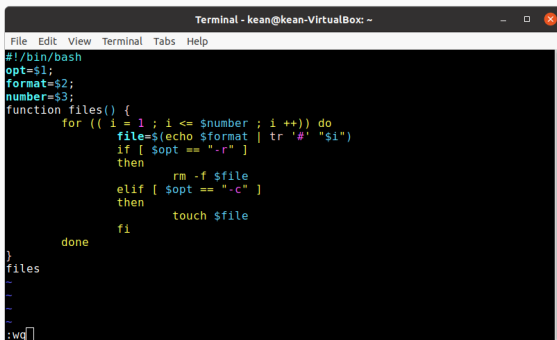
```
#include <stdio.h>
#include <stdlib.h>
int main(){
    printf("Input numbers\n ");
    int a;
    scanf("%d", &a);
    if (a<0) exit(0);
    if (a>0) exit(1);
    if (a==0) exit(2);
    return 0;
}
```

U:\*\*\*- program2.c All L6 (C/\*l Abbrev)

Рис. 2: создание файла program2.c



# создание файла Program3.sh

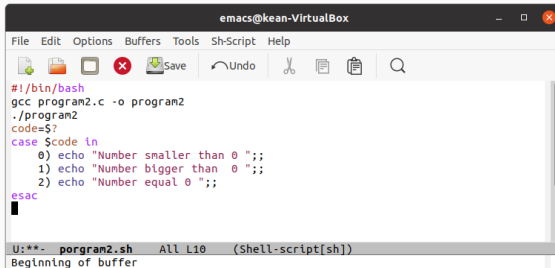


The image shows a terminal window titled "Terminal - kean@kean-VirtualBox: ~". The window contains a shell script being edited. The script starts with a shebang line and sets variables for options, format, and number. It then defines a function named 'files' that iterates from 1 to the number of files, creating or removing files based on the option provided. The script ends with a call to the 'files' function. The terminal shows the script being edited with a cursor at the end of the last line.

```
#!/bin/bash
opt=$1;
format=$2;
number=$3;
function files() {
    for (( i = 1 ; i <= $number ; i ++)) do
        file=$(echo $format | tr '#' "$i")
        if [ $opt == "-r" ]
        then
            rm -f $file
        elif [ $opt == "-c" ]
        then
            touch $file
        fi
    done
}
files
~
~
~
:wq
```

Рис. 3: создание файла Program3.sh

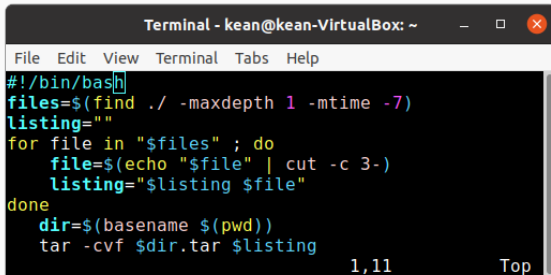
# результат работы командного файла



```
emacs@kean-VirtualBox
File Edit Options Buffers Tools Sh-Script Help
Save Undo
#!/bin/bash
gcc program2.c -o program2
./program2
code=$?
case $code in
  0) echo "Number smaller than 0 ";;
  1) echo "Number bigger than 0 ";;
  2) echo "Number equal 0 ";;
esac
█
U:***- program2.sh All L10 (Shell-script[sh])
Beginning of buffer
```

Рис. 4: результат работы командного файла

# создание файла Program4.sh

A terminal window titled "Terminal - kean@kean-VirtualBox: ~" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The terminal shows the following commands and output:

```
#!/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 cursor is at the end of the first line. The bottom right of the terminal shows "1,11" and "Top".

```
Terminal - kean@kean-VirtualBox: ~
File Edit View Terminal Tabs Help
#!/bin/bas[
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
1,11 Top
```

Рис. 5: создание файла Program4.sh

## **Выводы по проделанной работе**

---

научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

## Библиография

1. (Лабораторная работа №12)  
[https://esystem.rudn.ru/pluginfile.php/1142380/mod\\_resource/content/1/lab\\_shell\\_prog\\_2.pdf](https://esystem.rudn.ru/pluginfile.php/1142380/mod_resource/content/1/lab_shell_prog_2.pdf)
2. (stackexchange)  
<https://vi.stackexchange.com/questions/10209/execute-current-buffer-as-bash-script-from-vim>
3. (BASH: функция getopt — используем опции в скриптах)  
[https://esystem.rudn.ru/pluginfile.php/1142380/mod\\_resource/content/1/lab\\_shell\\_prog\\_2.pdf](https://esystem.rudn.ru/pluginfile.php/1142380/mod_resource/content/1/lab_shell_prog_2.pdf)
4. (stackoverflow)  
<https://stackoverflow.com/questions/16483119/an-example-of-how-to-use-getopts-in-bash>