

# Отчет по лабораторной работе 11

Программирование в командном процессоре ОС UNIX.  
Ветвления и циклы

---

Шалыгин Георгий

РУДН, НФИбд-02-20

1. Цель работы

2. Основные понятия

3. Результаты

4. Вывод

## Цель работы

---

Изучить основы программирования в оболочке ОС UNIX/Linux.  
Научиться писать небольшие командные файлы.

- Объект исследования: система UNIX.
- Предмет исследования: программирование в UNIX.

# Основные понятия

---

- **Командный язык** - это язык, на котором пользователь взаимодействует с системой в интерактивном режиме.
- **Командный интерпретатор**, интерпретатор командной строки - компьютерная программа, часть операционной системы, обеспечивающая базовые возможности управления компьютером посредством интерактивного ввода команд через интерфейс командной строки или последовательного исполнения пакетных командных файлов.

## Результаты

---



# Обработка аргументов

Используя команды `getopts` `grep`, напомним командный файл, который анализирует командную строку с ключами (рис.

`-@fig:002`):

- `-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.

```
gashalygin@geshalygin:~  
Файл Правка Вид Поиск Терминал Справка  
#!/bin/bash  
read=\*;find=\*;of=0;nf=0;cf=0;if=0;pf=0;  
while getopts i:o:p:Cn com  
do case $com in  
  i) if=1; read=$OPTARG;;  
  o) of=1; write=$OPTARG;;  
  p) pf=1; find=$OPTARG;;  
  C) cf=1;;  
  n) nf=1;;  
esac  
done  
if (let $if==1)&&(let $pf==1)  
  then if (let $of==1)  
    then if (let $cf==0)
```

# Запуск приложений на с++

Напишем 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.

```
#include <iostream>
#include <stdlib.h>
using namespace std;
int main()
{
    int num;
    cout << "enter num\n";
    cin >> num;
    if (num == 0)
        return 0;
    if (num > 0) exit(1);
    if (num < 0) exit(2);
}
```

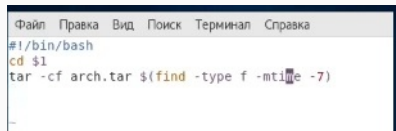
```
Файл  Правка  Вид  Поиск  Терминал  Спра
#!/bin/bash
./12
case $? in
0) echo "its 0";;
1) echo ">0";;
2) echo "<0";;
*) echo "error";;
e
```

# Создание файлов

Напишем командный файл, создающий N 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).

```
#!/bin/bash
typeset -i count=0;
for f in *.tmp
do rm $f
done
for ((count=1;count<=$1;count++))
do
    touch $count.tmp
done
```

Напишем командный файл, который 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`).



```
Файл  Правка  Вид  Поиск  Терминал  Справка
#!/bin/bash
cd $1
tar -cf arch.tar $(find -type f -mtime -7)
```

## Вывод

---

Изучить основы программирования в оболочке ОС UNIX/Linux.  
Научиться писать небольшие командные файлы.

Спасибо за внимание