

# Лабораторная работа №12. Программирование в командном процессоре ОС UNIX. Программирование в командном процессоре ОС UNIX. Ветвления и циклы.

---

Гекишева Анастасия Дмитриевна, НБИ-01-20,  
30 апреля, 2021

<sup>1</sup>RUDN University, Moscow, Russian Federation

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

1. Используя команды `getopts` `grep`, написать командный файл, который анализирует командную строку с ключами, а затем ищет в указанном файле нужные строки, определяемые ключом `-p`.
2. Написать на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю.
3. Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N. Этот же командный файл должен уметь удалять все созданные им файлы.
4. Написать командный файл, который с помощью команды `tar` запаковывает в архив все файлы в указанной директории.

## Выполнение лабораторной работы

---

Перед выполнением лабораторной работы я хорошо ознакомилась с теоритическим материалом для её выполнения Ссылка 1

## Выполнение 1-го пункта задания

---

# Командный файл, который анализирует командную строку с ключами

```
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
while getopts i:op: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 Illegaloption $optletter
    esac
done
if(((Cflag==1)&&(nflag==1)))
then grep -e${pval} -i -n ${ival}
    if((oflag==1))
    then grep -e${pval} -i -n ${ival} > ${oval}
    fi
fi
if(((Cflag==0)&&(nflag==0)))
then grep -e${pval} ${ival}
    if((oflag==1))
    then grep -e${pval} ${ival} > ${oval}
    fi
fi
if(((Cflag==0)&&(nflag==1)))
then grep -e${pval} -n ${ival}
    if((oflag==1))
    then grep -e${pval} -n ${ival} > ${oval}
    fi
fi
if(((Cflag==1)&&(nflag==0)))
then grep -e${pval} -i ${ival}
    if((oflag==1))
    then grep -e${pval} -i ${ival} > ${oval}
    fi
fi
fi
```

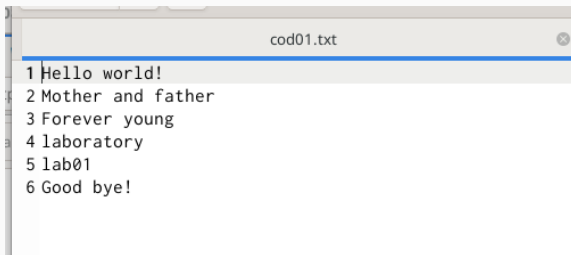


Рис. 2: Текстовый файл



## Результат выполнения командного файла

```
adkekisheva@dk8n76 ~ $ bash cod01.sh -icod01.txt -ocod.txt -plab -Cn
4:laboratory
5:lab01
adkekisheva@dk8n76 ~ $ bash cod01.sh -icod01.txt -ocod.txt -plab
laboratory
lab01
adkekisheva@dk8n76 ~ $ bash cod01.sh -icod01.txt -ocod.txt -pbye -n
6:Good bye!
```

Рис. 3: Результат выполнения командного файла

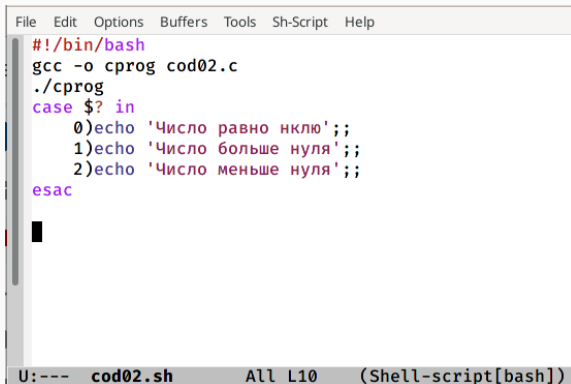
## Выполнение 2-го пункта задания

---

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int number;
6     printf("Введите число: ");
7     scanf("%i",&number);
8     if (number==0) exit(0);
9     else if(number>0) exit(1);
10    else exit(2);
11    return (3);
12 }
```

Рис. 4: Программа на языке Си: cod02.c

## Командный файл, вызывающий программу cod02.c



```
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
gcc -o cprog cod02.c
./cprog
case $? in
    0)echo 'Число равно нклю';;
    1)echo 'Число больше нуля';;
    2)echo 'Число меньше нуля';;
esac

U:--- cod02.sh All L10 (Shell-script[bash])
```

Рис. 5: Командный файл, вызывающий программу cod02.c

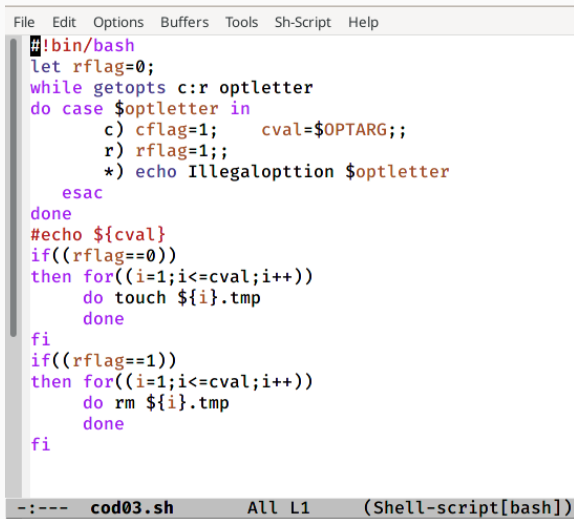
```
adkekisheva@dk8n76 ~ $ emacs cod02.c
adkekisheva@dk8n76 ~ $ bash cod02.sh
Введите число: 0
Число равно нулю
adkekisheva@dk8n76 ~ $ bash cod02.sh
Введите число: 4
Число больше нуля
adkekisheva@dk8n76 ~ $ bash cod02.sh
Введите число: -10
Число меньше нуля
```

Рис. 6: Результат работы программ

## Выполнение 3-го пункта задания

---

Командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N



```
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
let rflag=0;
while getopt s c:r optletter
do case $optletter in
    c) cflag=1;    cval=$OPTARG;;
    r) rflag=1;;
    *) echo Illegal option $optletter
    esac
done
#echo ${cval}
if((rflag==0))
then for((i=1;i<=cval;i++))
do touch ${i}.tmp
done
fi
if((rflag==1))
then for((i=1;i<=cval;i++))
do rm ${i}.tmp
done
fi

-:--- cod03.sh All L1 (Shell-script[bash])
```

Рис. 7: Командный файл, создающий указанное число файлов,

# Результат работы прграммы: создание и удаление

```
adkekisheva@dk8n76 ~ $ emacs cod03.sh
adkekisheva@dk8n76 ~ $ bash cod03.sh -c5
adkekisheva@dk8n76 ~ $ ls
1. tmp          cod03.sh      feathers      public        v
2. tmp          cod03.sh~    file.txt     public_html   Видео
3. tmp          '#cod2.sh#'  getopt.sh    script01.sh~  Документы
4. tmp          cod2.sh      getopt.sh~   script02.sh   Загрузки
5. tmp          cod2.sh~    GNUstep     script02.sh~  Изображения
cod01.sh        cod3.sh      lab11.c      script03.sh   Музыка
cod01.sh~      cod3.sh~    lab11.c~     script03.sh~  Общедоступные
cod01.txt       cod4.sh      laboratory    script04.sh   ОС
cod01.txt~     cod4.sh~    Makefile     script04.sh~  'Рабочий стол'
cod02.c        conf.txt     may          ser.sage      Шаблоны
cod02.c~       cprog       my_os        skr.sh~
cod02.sh       euler2.sage prog.ccp      text.txt
cod02.sh~      euler.sage  Programma    tmp
adkekisheva@dk8n76 ~ $ bash cod03.sh -c5 -r
adkekisheva@dk8n76 ~ $ ls
cod01.sh        cod2.sh~    getopt.sh~   script01.sh~  Видео
cod01.sh~      cod3.sh     GNUstep      script02.sh   Документы
cod01.txt       cod3.sh~    lab11.c      script02.sh~  Загрузки
cod01.txt~     cod4.sh     lab11.c~     script03.sh   Изображения
cod02.c        cod4.sh~    laboratory    script03.sh~  Музыка
cod02.c~       conf.txt    Makefile     script04.sh   Общедоступные
cod02.sh       cprog       may          script04.sh~  ОС
cod02.sh~      euler2.sage my_os        ser.sage      'Рабочий стол'
cod03.sh       euler.sage  prog.ccp     skr.sh~       Шаблоны
cod03.sh~      feathers    Programma    text.txt
'#cod2.sh#'    file.txt   public       tmp
cod2.sh        getopt.sh  public_html  v
```

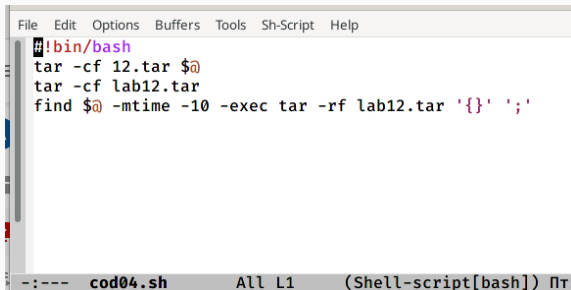
Рис. 8: Результат работы прграммы: создание и удаление



## Выполнение 4-го пункта задания

---

## Командный файл, архивирует файлы в указанной директории



```
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
tar -cf 12.tar $@
tar -cf lab12.tar
find $@ -mtime -10 -exec tar -rf lab12.tar '{}' ';'

-:--- cod04.sh All L1 (Shell-script[bash]) Пт
```

Рис. 9: Командный файл, архивирует файлы в указанной директории

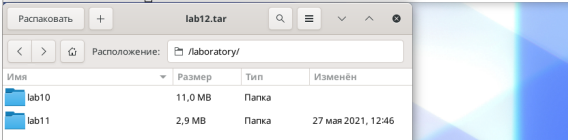
# Вызов командного файла

```
adkekisheva@dk8n76 ~ $ bash cod04.sh laboratory
tar: 12.tar: Функция open завершилась с ошибкой: Превышена дисковая квота
tar: Error is not recoverable: exiting now
tar: Робкий отказ от создания пустого архива
Попробуйте «tar --help» или «tar --usage» для
получения более подробного описания.
tar: lab12.tar: Функция read завершилась с ошибкой: Неправильный дескриптор файла
tar: Начало ленты, завершение работы
tar: Error is not recoverable: exiting now
```

Рис. 10: Вызов командного файла

# Результат работы командного файла

```
adkekisheva@dk8n76 ~ $ ls
cod01.sh      '#cod04.sh#'  conf.txt      Makefile      script03.sh  Изображения
cod01.sh~    cod04.sh      cprog        may           script03.sh~ Музыка
cod01.txt    cod04.sh~    file.txt     my_os        script04.sh  Общедоступные
cod01.txt~   '#cod2.sh#'  getopt.sh    prog.ccp     script04.sh~ ОС
cod02.c      cod2.sh      getopt.sh~   Programma    skr.sh~     'Рабочий стол'
cod02.c~    cod3.sh      GNUstep     public        tmp         Шаблоны
cod02.sh     cod3.sh~    lab11.c     public_html   v
cod02.sh~   cod4.sh     lab11.c~    script01.sh~  Видео
cod03.sh     cod4.sh~    lab12.tar   script02.sh   Документы
cod03.sh~   cod.txt     laboratory  script02.sh~  Загрузки
adkekisheva@dk8n76 ~ $
```



Имя	Размер	Тип	Изменён
lab10	11,0 MB	Папка	
lab11	2,9 MB	Папка	27 мая 2021, 12:46

Рис. 11: Результат работы командного файла

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

1. Ссылка 1