

# Лабораторная работа №11. Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Дисциплина: Операционные системы

---

Ганина Т. С.

19 апреля 2023

Группа НКАбд-01-22

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Ганина Таисия Сергеевна
- Студентка 1-го курса, группа НКАбд-01-22
- Компьютерные и информационные науки
- Российский университет дружбы народов
- Ссылка на репозиторий гитхаба `tsganina`

## Вводная часть

---

- Работа с программами позволит создать готовые скрипты для различных задач.

- Командный процессор ОС UNIX

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

Выполнение заданий.

---



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; - `-р` 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 `-р`.

```
1  #! /bin/bash
2  while getopts i:o:p:cn optletter
3  do
4  case $optletter in
5      i) iflag=1; ival=$OPTARG;;
6      o) oflag=1; oval=$OPTARG;;
7      p) pflag=1; pval=$OPTARG;;
8      c) cflag=1;;
9      n) nflag=1;;
10     *) echo Illegal option $optletter;;
11     esac
12 done
13 if ! test $cflag
14 then
15     cf=-1
16 fi
17 if test $nflag
18 then
19     nf=-n
20 fi
21 grep $cf $nf $pval $ival >> $oval
```

```
1 В динамиках лишь звенящая тишина,  
2  
3 Шелест волн и, бог знает, какой прибой...  
4  
5 Эта история больше совсем никому не нужна:  
6  
7 Лишь я остаюсь в постоянном контакте с тобой.  
8  
9  
10 Я упрямо слушаю этот извечный шум,  
11  
12 Сквозь помехи лоя одной мне приоткрытый ритм.  
13  
14 Голова ломается от извечных тяжелых дум,  
15  
16 Хочу на дно уйти, добраться до тех глубин.  
17  
18  
19 52 Герца, странно и чуждо всем.
```

Рис. 2: Файл, в котором выполнялась программа

```
1 6:Я упрямо слушаю этот извечный шум,  
2 13:Я тону в океане своих и чужих проблем,  
3 21:Я рисую графики, крепче держу гидрофон.  
4 22:Я готова кричать: я тоже здесь, приходи.  
5 29:Я надеюсь, что ты вернешься ко мне невредим.  
6 33:Я поймала сигнал: нервы тянут не хуже каната.  
7 6:Я упрямо слушаю этот извечный шум,  
8 13:Я тону в океане своих и чужих проблем,  
9 21:Я рисую графики, крепче держу гидрофон.  
10 22:Я готова кричать: я тоже здесь, приходи.  
11 29:Я надеюсь, что ты вернешься ко мне невредим.  
12 33:Я поймала сигнал: нервы тянут не хуже каната.
```

Рис. 3: Результат

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

```
1 #include <stdlib.h>
2 #include <stdio.h>
3
4 int main () {
5     int n;
6     printf ("Введите число: ");
7     scanf ("%d", &n);
8     if(n>0){
9         exit(1);
10    }
11    else if (n==0) {
12        exit(0);
13    }
14    else {
15        exit(2);
16    }
17 }
```

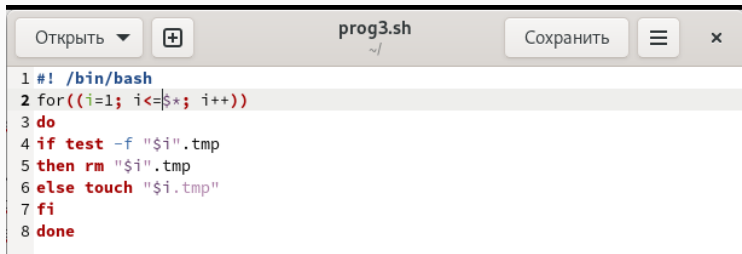
```
1  #!/bin/bash
2
3  gcc -o cprog lab11_2.c
4  ./cprog
5  case $? in
6      0) echo "Число равно нулю";;
7      1) echo "Число больше нуля";;
8      2) echo "Число меньше нуля";;
9  esac
```

Рис. 5: Программный файл

```
[tsganina@fedora ~]$ gedit prog2.sh
[tsganina@fedora ~]$ gedit lab11_2.c
[tsganina@fedora ~]$ bash prog2.sh 12
Введите число: 12
Число больше нуля
[tsganina@fedora ~]$ bash prog2.sh
Введите число: 0
Число равно нулю
[tsganina@fedora ~]$ bash prog2.sh
Введите число: -123
Число меньше нуля
[tsganina@fedora ~]$
```

Рис. 6: Результат

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)



```
1 #! /bin/bash
2 for((i=1; i<=$*; i++))
3 do
4 if test -f "$i".tmp
5 then rm "$i".tmp
6 else touch "$i.tmp"
7 fi
8 done
```

Рис. 7: Программный файл

```

[tsganina@fedora ~]$ bash prog3.sh 5
[tsganina@fedora ~]$ ls
1.tmp      conf.txt    monthly     ski.plases   КомпАл
2.tmp      cprog      my_os       tmp05_lab    лаб10_файлы
3.tmp      feathers    output.txt  work         Музыка
4.tmp      file.txt    play        'Без названия.ipynb'  Общедоступные
5.tmp      lab07.sh   prog1.sh    'Библиотека calibre'  'Рабочий стол'
abc1      lab07.sh~  prog2.sh    Видео        Шаблоны
australia lab11_1.txt prog3.sh     Документы
backup     lab11_2.c -r          Загрузки
bin        may        reports     Изображения

[tsganina@fedora ~]$ bash prog3.sh 5
[tsganina@fedora ~]$ ls
abc1      lab07.sh   play        work         лаб10_файлы
australia lab07.sh~  prog1.sh    'Без названия.ipynb'  Музыка
backup     lab11_1.txt prog2.sh    'Библиотека calibre'  Общедоступные
bin        lab11_2.c  prog3.sh    Видео        'Рабочий стол'
conf.txt   may        -r          Документы    Шаблоны
cprog      monthly    reports     Загрузки
feathers    my_os      ski.plases  Изображения
file.txt    output.txt tmp05_lab    КомпАл

[tsganina@fedora ~]$

```

Рис. 8: Результат



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`)



```
1 #! /bin/bash
2 find $* -mtime -7 -mtime +0 -type f > FILES.txt
3 tar -cf archive.tar -T FILES.txt
```

Рис. 9: Файл и запуск

```
аналогичная команда: touch
[tsganina@fedora ~]$ touch prog4.sh
[tsganina@fedora ~]$ gedit prog4.sh
[tsganina@fedora ~]$ pwd
/home/tsganina
[tsganina@fedora ~]$ ^C
[tsganina@fedora ~]$ bash prog4.sh /home/tsganina
find: '/home/tsganina/monthly': Отказано в доступе
tar: Удаляется начальный '/' из имен объектов
tar: Удаляются начальные '/' из целей жестких ссылок
[tsganina@fedora ~]$ gedit prog4.sh
```



Рис. 10: Текст программы 4

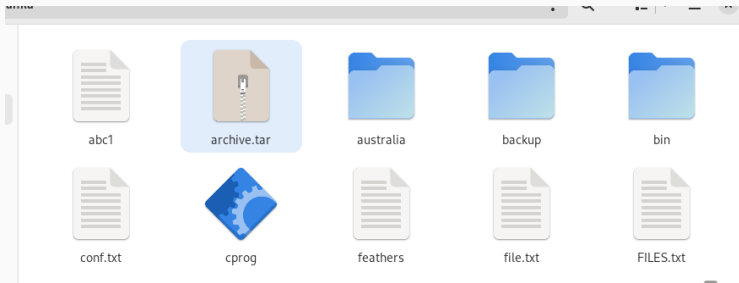
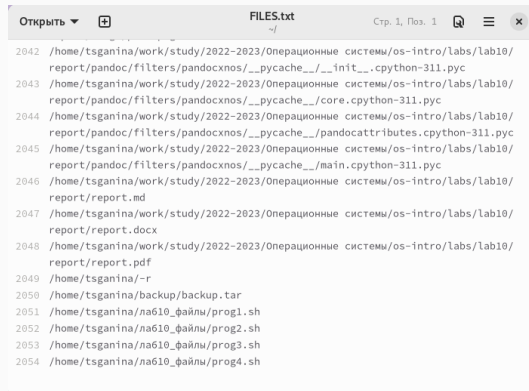


Рис. 11: Созданный архив и файл



```
2042 /home/tsganina/work/study/2022-2023/Операционные системы/os-intro/labs/lab10/  
report/pandoc/filters/pandocxnos/___pycache___/___init___cpython-311.pyс  
2043 /home/tsganina/work/study/2022-2023/Операционные системы/os-intro/labs/lab10/  
report/pandoc/filters/pandocxnos/___pycache___/core.cpython-311.pyс  
2044 /home/tsganina/work/study/2022-2023/Операционные системы/os-intro/labs/lab10/  
report/pandoc/filters/pandocxnos/___pycache___/pandocattributes.cpython-311.pyс  
2045 /home/tsganina/work/study/2022-2023/Операционные системы/os-intro/labs/lab10/  
report/pandoc/filters/pandocxnos/___pycache___/main.cpython-311.pyс  
2046 /home/tsganina/work/study/2022-2023/Операционные системы/os-intro/labs/lab10/  
report/report.md  
2047 /home/tsganina/work/study/2022-2023/Операционные системы/os-intro/labs/lab10/  
report/report.docx  
2048 /home/tsganina/work/study/2022-2023/Операционные системы/os-intro/labs/lab10/  
report/report.pdf  
2049 /home/tsganina/-r  
2050 /home/tsganina/backup/backup.tar  
2051 /home/tsganina/ла610_файлы/prog1.sh  
2052 /home/tsganina/ла610_файлы/prog2.sh  
2053 /home/tsganina/ла610_файлы/prog3.sh  
2054 /home/tsganina/ла610_файлы/prog4.sh
```

Рис. 12: FILES.txt

## Результаты

---

В процессе выполнения данной лабораторной работы я изучила основы программирования в оболочке ОС UNIX. Научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.