

Отчет по лабораторной работе №12

Тема:

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Российский Университет Дружбы Народов

Факультет Физико-Математических и Естественных Наук

Дисциплина: *Операционные системы*

Студент: Мухамедияр Адиль

Группа: НКНбд-01-20

Москва, 2021г.

Цель работы

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

Введение

Операции с блоками кода — ключ к структурированным и упорядоченным сценариям оболочки. Циклы и ветвления являются теми инструментальными средствами, которые предоставляют возможность достигнуть этой цели.

Циклы

Цикл это блок команд, который повторяется (итерируется) до тех пор, пока не будет выполнено условие выхода из цикла .

Ход работы:

- Для начало я скопировал любой текст из интернета.

File Edit Options Buffers Tools Help



Save



Undo



Why is reading not popular today? Teenagers try to spend their free time with their cell phones. They can chat, play, text, but they don't read.

We can always hear that our ministers of education attribute the crisis of reading to so methods of teaching. In their speeches they call teaching too archaic or too modern.

- Используя команды `getopts` `grep`, написал командный файл.

File Edit Options Buffers Tools Sh-Script 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 не правильный ввод $optletter
esac
done
if (((Cflag=1)&&(nflag=1)))
then grep -e${pval} -n ${ival}
if ((oflag==1))
then grep -e${pval} -n ${ival} > ${oval}
fi
fi
if (((Cflag==0)&&(nflag==1)))
then grep -e${pval} -i -n ${ival}
if ((oflag==1))
then grep -e${pval} -i -n ${ival} > ${oval}
```

- Данный командный файл анализирует командную строку с ключами: -iinputfile — прочитать данные из указанного файла; -ooutputfile — вывести данные в указанный файл; -ршаблон — указать шаблон для поиска; -C — различать большие и малые буквы; -n — выдавать номера строк.



```
then grep -e${pval} -i -n ${ival} > ${oval}
fi
fi
if (((Cflag==1)&&(nflag==0)))
then grep -e${pval} ${ival}
if ((oflag==1))
then grep -e${pval} ${ival} > ${oval}
fi
fi
if (((Cflag==0)&&(nflag==0)))
then grep -e${pval} -i ${ival}
if ((oflag==1))
then grep -e${pval} -i ${ival} > ${oval}
fi
fi
```

- После я написал слово, которую программа ищет в моем тексте, используя ключ -p.

```
[amukhamediyar@amukhamediyar ~]$ bash lab12.sh -ilab12.txt -olab12_2.txt -pthe
1:Why is reading not popular today? Teenagers try to spend their free time with their c
ell phones. They can chat, play, text, but they don't read.
3:We can always hear that our ministers of education attribute the crisis of reading to
methods of teaching. In their speeches they call teaching too archaic or too modern.
[amukhamediyar@amukhamediyar ~]$ bash lab12.sh -ilab12.txt -olab12_2.txt -pthe -n
1:Why is reading not popular today? Teenagers try to spend their free time with their c
ell phones. They can chat, play, text, but they don't read.
3:We can always hear that our ministers of education attribute the crisis of reading to
methods of teaching. In their speeches they call teaching too archaic or too modern.
[amukhamediyar@amukhamediyar ~]$ bash lab12.sh -ilab12.txt -olab12_2.txt -pthe -C
1:Why is reading not popular today? Teenagers try to spend their free time with their c
ell phones. They can chat, play, text, but they don't read.
3:We can always hear that our ministers of education attribute the crisis of reading to
methods of teaching. In their speeches they call teaching too archaic or too modern.
```

2. Написал 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 о коде завершения в оболочку.

File Edit Options Buffers Tools C++ Help

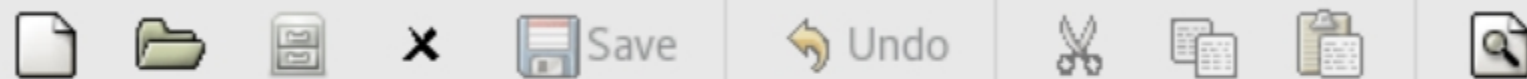


```
#include <stdio.h>
#include <stdlib.h>

int main ()
{
    int x;
    printf("input: ");
    scanf("%a", &x);
    if(x == 0) exit(0);
    else if (x < 0) exit(1);
    else if (x > 0) exit(2);
    return(3);
}
```

- Командный файл вызывает эту программу и, проанализировав с помощью команды \$?, выдает сообщение о том, какое число было введено.

File Edit Options Buffers Tools Sh-Script Help

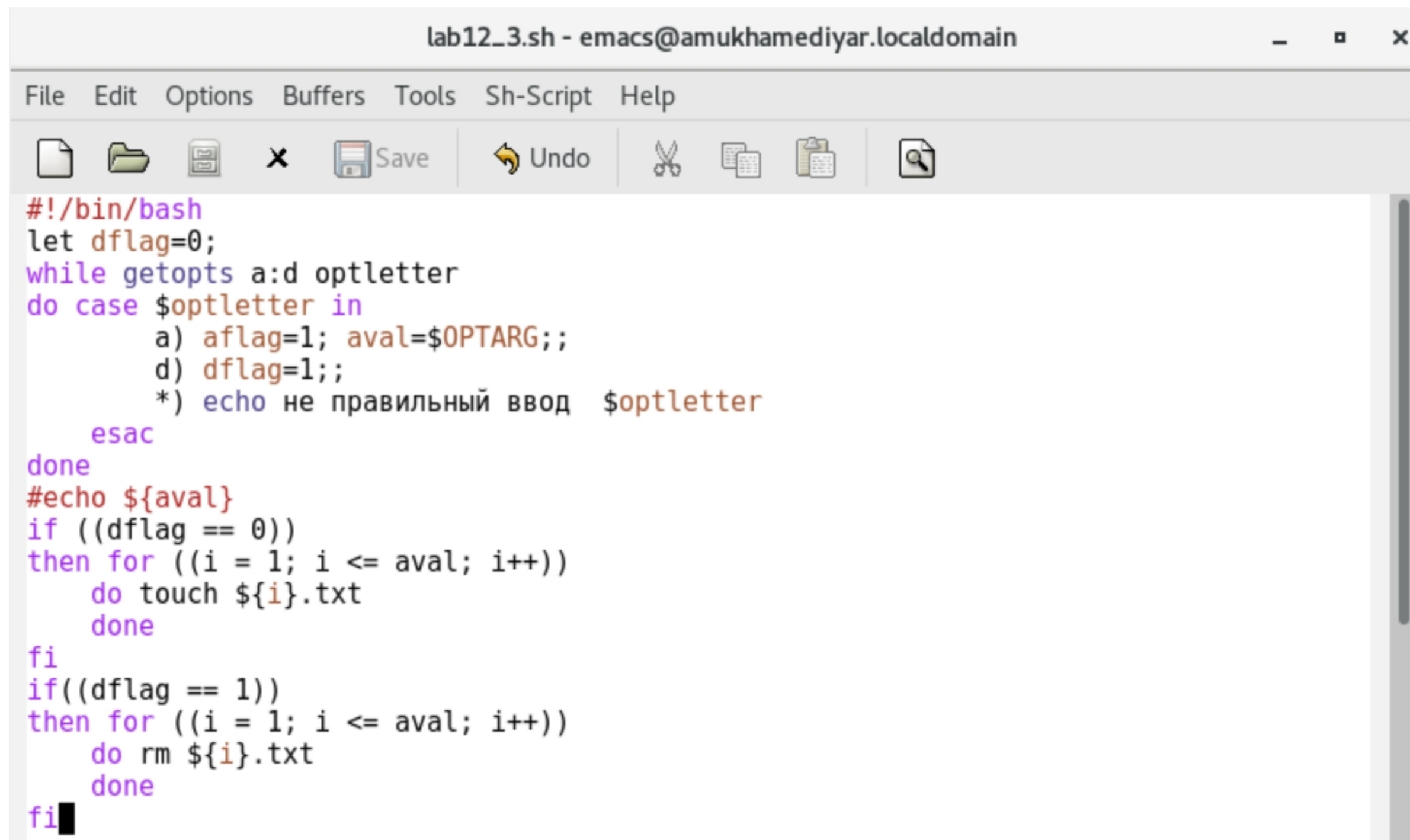


```
#!/bin/bash
gcc -o cprog lab12_2.cpp
./cprog
case $? in
    0) echo "input number is equal to 0 ";;
    1) echo "input number is smaller than 0 ";;
    2) echo "input number is bigger than 0 ";;
esac
```

- Проверил свой код.


```
[amukhamediyar@amukhamediyar ~]$ bash lab12_2.sh
input: 0
input number is equal to 0
[amukhamediyar@amukhamediyar ~]$ bash lab12_2.sh
input: -40
input number is smaller than 0
[amukhamediyar@amukhamediyar ~]$ bash lab12_2.sh
input: 19
input number is bigger than 0
[amukhamediyar@amukhamediyar ~]$
```

3. Написал командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N (например 1.tmp, 2.tmp, 3.tmp, 4.tmp и т.д.).



```
#!/bin/bash
let dflag=0;
while getopts a:d optletter
do case $optletter in
    a) aflag=1; aval=$OPTARG;;
    d) dflag=1;;
    *) echo не правильный ввод $optletter
    esac
done
#echo ${aval}
if ((dflag == 0))
then for ((i = 1; i <= aval; i++))
do touch ${i}.txt
done
fi
if((dflag == 1))
then for ((i = 1; i <= aval; i++))
do rm ${i}.txt
done
fi
```

- Число файлов, которые необходимо создать, передаётся в аргументы командной строки.

```
[amukhamediyar@amukhamediyar ~]$ bash lab12_3.sh -a4
[amukhamediyar@amukhamediyar ~]$ ls -l
итого 104
drwxrwxr-x. 3 amukhamediyar amukhamediyar 47 апр 27 21:35 -
-rw-rw-r--. 1 amukhamediyar amukhamediyar 0 май 28 17:48 1.txt
-rw-rw-r--. 1 amukhamediyar amukhamediyar 0 май 28 17:48 2.txt
-rw-rw-r--. 1 amukhamediyar amukhamediyar 0 май 28 17:48 3.txt
-rw-rw-r--. 1 amukhamediyar amukhamediyar 0 май 28 17:48 4.txt
-rw-rw-r--. 1 amukhamediyar amukhamediyar 0 май 6 15:23 abc1
-rw-rw-r--. 1 amukhamediyar amukhamediyar 0 май 6 15:35 abcd1
-rwxr--r--. 1 amukhamediyar amukhamediyar 0 май 6 15:46 australia
drwxrwxr-x. 2 amukhamediyar amukhamediyar 43 май 28 15:06 backup
-rw-rw-r--. 1 amukhamediyar amukhamediyar 750 май 12 14:48 conf.txt
-rwxrwxr-x. 1 amukhamediyar amukhamediyar 8464 май 28 17:38 cprog
```

- Этот же командный файл удаляет все созданные им файлы (если они существуют).

```
[amukhamediyar@amukhamediyar ~]$ bash lab12_3.sh -a4 -d
[amukhamediyar@amukhamediyar ~]$ ls -l
итого 104
drwxrwxr-x. 3 amukhamediyar amukhamediyar 47 апр 27 21:35 -
-rw-rw-r--. 1 amukhamediyar amukhamediyar 0 май 6 15:23 abc1
-rw-rw-r--. 1 amukhamediyar amukhamediyar 0 май 6 15:35 abcd1
-rwxr--r--. 1 amukhamediyar amukhamediyar 0 май 6 15:46 australia
drwxrwxr-x. 2 amukhamediyar amukhamediyar 43 май 28 15:06 backup
-rw-rw-r--. 1 amukhamediyar amukhamediyar 750 май 12 14:48 conf.txt
-rwxrwxr-x. 1 amukhamediyar amukhamediyar 8464 май 28 17:38 cprog
-rwxr--r--. 1 amukhamediyar amukhamediyar 0 май 6 15:47 feathers
-rw-rw-r--. 1 amukhamediyar amukhamediyar 2857 май 12 14:46 file.txt
-rw-rw-r--. 1 amukhamediyar amukhamediyar 0 май 27 16:52 k.sh~
-rw-rw-r--. 1 amukhamediyar amukhamediyar 204 май 19 17:11 #lab#
```

4. Написал командный файл, который с помощью команды *tar* запаковывает в архив все файлы в указанной директории.

lab12_4.sh - emacs@amukhamediyar.localdomain

File Edit Options Buffers Tools Sh-Script Help

```
#!/bin/bash
tar -cf lab12_4_l.tar $@
find $@ -mtime -7 -exec tar -rf lab12_4_l_modif.tar {} \;
```

- Модифицировал его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовал команду find).

```
[amukhamediyar@amukhamediyar ~]$ ./lab12_4.sh x
[amukhamediyar@amukhamediyar ~]$ ls
-          #lab#      lab11_4.sh~  lab12_3.sh~  my_os      Документы
9.tar      #lab07.sh#  lab11.sh    lab12_4_l_modif.tar  play      Загрузки
abc1       lab07.sh    lab11.sh~   lab12_4_l.tar  reports   Изображения
abcd1      lab_08      lab12_2.C++~ lab12_4.sh    ski.places Музыка
australia  lab11-12    lab12_2.cpp  lab12_4.sh~   test      Общедоступные
backup     lab11_2.c~  lab12_2.cpp~ lab12.sh      testlab   Рабочий стол
conf.txt   lab11_2.sh  lab12_2.sh  lab12.sh~    text.txt  Шаблоны
cprog     lab11_2.sh~ lab12_2.sh~  lab12.txt    tutorial
feathers   lab11_3.sh  lab12_2.txt  lab12.txt~   work
file.txt   lab11_3.sh~ lab12_2.txt~ may
k.sh~      lab11_4.sh  lab12_3.sh  monthly
[amukhamediyar@amukhamediyar ~]$
```

Вывод

Изучил основы программирования в оболочке ОС UNIX, научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Библиография

[Домашний склерозник](#)

Ответы на контрольные вопросы:

1. Команда getoptс является встроенной командой командной оболочки bash, предназначенной для разбора параметров сценариев. Она обрабатывает исключительно однобуквенные параметры как с аргументами, так и без них и этого вполне достаточно для передачи сценариям любых входных данных.
2. При генерации имен используют метасимволы:

- произвольная (возможно пустая) последовательность символов; ? один произвольный символ; [...] любой из символов, указанных в скобках перечислением и/или с указанием диапазона; cat f* выдаст все файлы каталога, начинающиеся с "f"; cat f выдаст все файлы, содержащие "f"; cat program.? выдаст файлы данного каталога с однобуквенными расширениями, скажем "program.c" и "program.o", но не выдаст "program.com"; cat [a-d]* выдаст файлы, которые начинаются с "a", "b", "c", "d". Аналогичный эффект дадут и команды "cat [abcd]" и "cat [bdac]".

3. Операторы && и || являются управляющими операторами. Если в командной строке стоит command1 && command2, то command2 выполняется в том, и только в том случае, если статус выхода из команды command1 равен нулю, что говорит об успешном ее завершении. Аналогично, если командная строка имеет вид command1 || command2, то команда command2 выполняется тогда, и только тогда, когда статус выхода из команды command1 отличен от нуля.
4. Оператор break завершает выполнение ближайшего включающего цикла или условного оператора, в котором он отображается.
5. Команда true всегда возвращает ноль в качестве выходного статуса для индикации успеха. Команда false всегда возвращает не-ноль в качестве выходного статуса для индикации неудачи. Во всех управляющих конструкциях в качестве логического значения используется код возврата из программы, указанной в качестве условия. Код возврата 0 – истина, любое другое значение – ложь. Программа true – всегда завершается с кодом 0, false – всегда завершается с кодом 1.
6. Введенная строка означает условие существования файла man\$s/\$i.\$s
7. Цикл While выполняется до тех пор, пока указанное в нем условие истинно. Когда указанное условие становится ложным - цикл завершается. Цикл Until выполняется до тех пор, пока указанное в нем условие ложно.