

# Отчёт о выполнении лабораторной работы №12

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

**Российский Университет Дружбы Народов**

**Факультет Физико-Математических и Естественных Наук**

**Дисциплина:** *Операционные системы*

**Студент:** *Евселев Дмитрий*

**Группа:** *НКНбд-01-20*

## Цель работы

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

## Задание

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 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.
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`).

## Выполнение работы

1. Используя команды `getopts` `grep`, написала командный файл, который анализирует командную строку с ключами:

Рис.1 Анализ командной строки с ключами

```
#!/bin/bash
while getopts i:o:p:Cn optletter
do case $optletter in
    i) iflag=1; ival=$OPTARG;;
    o) oflag=1; oval=$OPTARG;;
    p) oflag=1; pval=$OPTARG;;
    C) Cflag=1;;
    n) nflag=1;;
    *) echo NoSuchOption $optletter
esac
done
if (((Cflag==1)&&(nflag==1)))
then grep -e${pval} -i -n ${ival}
    if ((oflag==1))
    then grep -e${pval} -i -n ${ival} > ${oval}
    fi
fi
if (((Cflag==0)&&(nflag==1)))
then grep -e${pval} -n ${ival}
    if ((oflag==1))
    then grep -e${pval} -n ${ival} > ${oval}
    fi
fi
if (((Cflag==0)&&(nflag==0)))
then grep -e${pval} ${ival}
    if ((oflag==0))
    then grep -e${pval} ${ival} > ${oval}
    fi
fi
```

- 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 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.

Рис.3 Программа на Си

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(void) {
5      int a;
6
7      printf("Enter the integer:\n");
8      if (scanf("%d", &a) != 1) {
9          printf("Incorrect input\n");
10         exit(-1);
11     }
12
13     if (a > 0)
14         exit(0);
15
16     if (a < 0)
17         exit(1);
18
19     else
20         exit(2);
21 }

```

Рис.4 Командный файл

```

#!/bin/bash

gcc 2.c

./a.out

case $? in
    0) echo pos;;
    1) echo neg;;
    2) echo null
esac

```

Рис.5 Запуск командного файла

```

dimon@vostro:~/work2/2020-2021/os/laboratory/lab12$ ./scr2.sh
4
Enter the integer:
pos
dimon@vostro:~/work2/2020-2021/os/laboratory/lab12$ ./scr2.sh
Enter the integer:
0
null
dimon@vostro:~/work2/2020-2021/os/laboratory/lab12$ ./scr2.sh
Enter the integer:
-3
neg

```

3. Написал командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N. Число файлов, которые необходимо создать, передаётся в

аргументы командной строки. Этот же командный файл умеет удалять все созданные им файлы (если они существуют).

Рис.6 Командный файл

```
#!/bin/bash

name=1
n=$1

while (($name!=($n+1)))
do
    touch $name.tmp
    ((name+=1))
done
name=1
echo "$n files created. delete them?"
read d
if ((d==1))
then while (($name!=($n+1)))
do
    rm $name.tmp
    echo "${name}.tmp has been deleted"
    ((name+=1))
done
fi
```

Рис.7 Запуск командного файла

```

dimon@vostro: ~/work2/2020-2021/os/laboratory/lab12
rm: невозможно удалить 'forback.txt': Нет такого файла или каталога
dimon@vostro:~/work2/2020-2021/os/laboratory/lab12$ ./scr4.sh /home/dimon/arch1
2.tgz /home/dimon/dir
dimon@vostro:~/work2/2020-2021/os/laboratory/lab12$ chmod scr3.sh
chmod: пропущен операнд после «scr3.sh»
По команде «chmod --help» можно получить дополнительную информацию.
dimon@vostro:~/work2/2020-2021/os/laboratory/lab12$ chmod +x scr3.sh
dimon@vostro:~/work2/2020-2021/os/laboratory/lab12$ ./scr3.sh 5
5 files created. delete them?
./scr3.sh: строка 14: ((: d: слишком много вложенных выражений (неверный маркер
«d»)
dimon@vostro:~/work2/2020-2021/os/laboratory/lab12$ ./scr3.sh 5
5 files created. delete them?
1
1.tmp has been deleted
2.tmp has been deleted
3.tmp has been deleted
4.tmp has been deleted
5.tmp has been deleted
dimon@vostro:~/work2/2020-2021/os/laboratory/lab12$ ./scr3.sh 5
5 files created. delete them?

```

4. Написал 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.

```

#!/bin/bash

find $2 -mtime -7 > forbackup.txt
tar czf $1 forbackup.txt
rm forbackup.txt

```

Рис.8 Командный файл

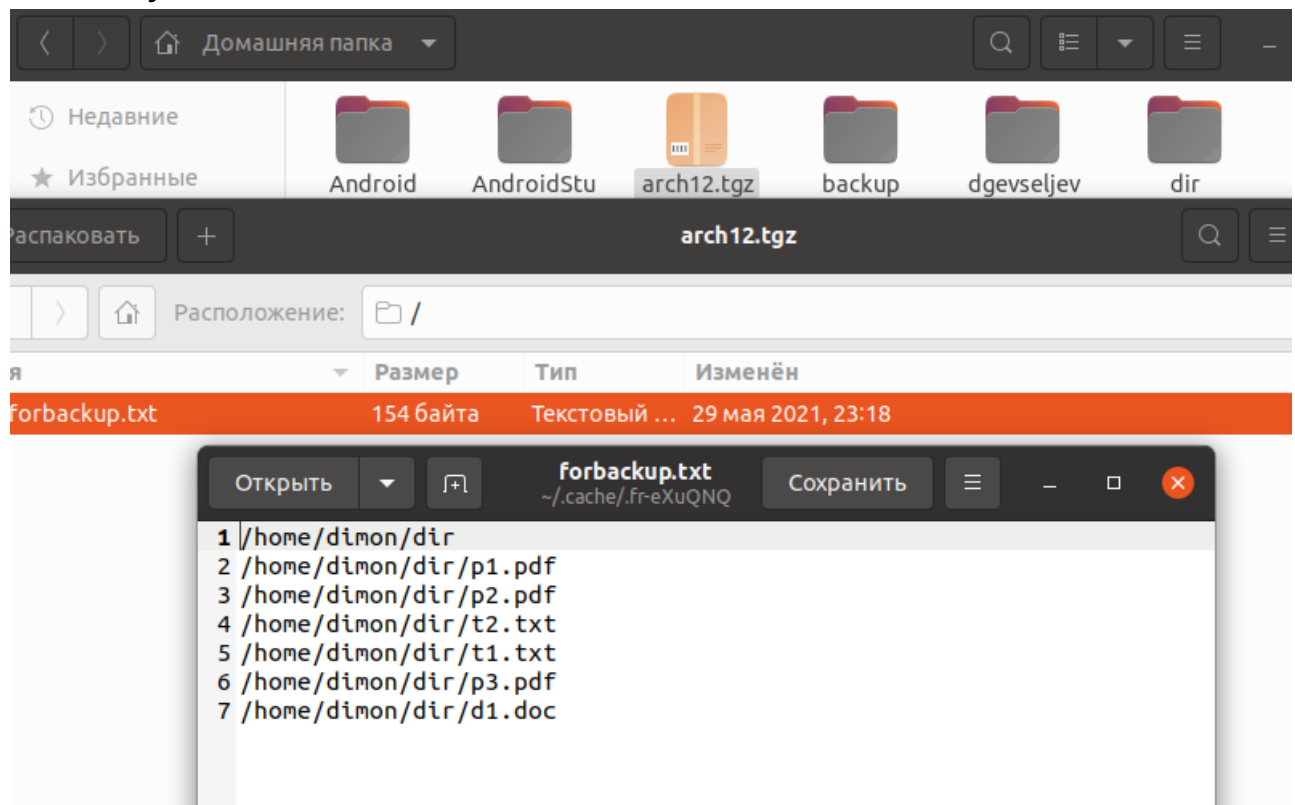
Рис.9 Запуск

```

dimon@vostro:~/work2/2020-2021/os/laboratory/lab12$ ./scr4.sh /home/dimon/arch1
2.tgz /home/dimon/dir

```

Рис.10 Результат



## Вывод

В ходе лабораторной работы я получил и изучил основы программирования в оболочке ОС UNIX/Linux, научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

## Контрольные вопросы

1. Каково предназначение команды `getopts`?

Она осуществляет синтаксический анализ командной строки, выделяя флаги, и используется для объявления переменных.

2. Какое отношение метасимволы имеют к генерации имён файлов?

При генерации имен файлов 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. Какие операторы управления действиями вы знаете?

for, case, if, while

4. Какие операторы используются для прерывания цикла?

Break, continue

5. Для чего нужны команды false и true?

Команды ОС UNIX возвращают код завершения, значение которого может быть использовано для принятия решения о дальнейших действиях. Команда `test`, например, создана специально для использования в командных файлах. Единственная функция этой команды заключается в выработке кода завершения.

6. Что означает строка `if test -f man$s/$i.$s`, встреченная в командном файле?

Означает условие существования файла `man$s/$i.$s`

7. Объясните различия между конструкциями `while` и `until`.

Если речь идет о 2-х параллельных действиях, то это `while`. когда мы показываем, что сначала делается 1-е действие. потом оно заканчивается при наступлении 2-го действия, применяем `until`.

---