

Лабораторная работа №11

**Программирование в командном процессоре ОС UNIX. Ветвления и
циклы**

Медникова Екатерина Михайловна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	11
4	Контрольные вопросы	12

Список иллюстраций

2.1	Задание 1	6
2.2	Результат 1	7
2.3	Задание 2	7
2.4	Результат 2	8
2.5	Задание 3	9
2.6	Результат 3	9
2.7	Задание 4	9
2.8	Результат 4	10

Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

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

```
#!/bin/bash

while getopts "i:o:p:Cn" opt
do
    case $opt in
        i) inputfile="$OPTARG";;
        o) outputfile="$OPTARG";;
        p) sample="$OPTARG";;
        C) reg="";;
        n) line="";;
        esac
    done

    grep -n "$sample" "$inputfile" > "outputfile".
```

Рис. 2.1: Задание 1

```
1:min.test  
2:man.test  
3:lan.test
```

Рис. 2.2: Результат 1

2. Написала на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю.

```
#include <iostream>  
using namespace std;  
  
int main(int argc, char *argv[]){  
    if (atoi(argv[1])>0) exit (1);  
    else if (atoi(argv[1])==0) exit(2);  
    else exit(3);  
    return 0;  
}
```

Рис. 2.3: Задание 2

```
#!/bin/bash

RES=result
SRC=comparison.cpp

if [ "$SRC" -nt "$RES" ]
then
    echo "Creating $RES ..."
    g++ -o $RES $SRC
fi

./RES $1

ers = $?

if [ "$erc" == "1" ]
then
    echo "input > 0"
fi
if [ "$erc" == "2" ]
then
    echo "input = 0"
fi
if [ "$erc" == "3" ]
then
    echo "input < 0"
fi
```

Рис. 2.4: Результат 2

3. Написала командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N.


```
#!/bin/bash

while getopts c:r opt
do
    case $opt in
        c)n="$OPTARG"; for i in $(seq 1 $n); do touch "$i.tmp"; done;;
        r)for i in $(find -name "*.tmp"); do rm $i; done;;
        esac
    done
```

Рис. 2.5: Задание 3

```
[emmednikova@fedora lab11.3]$ ./3 -c 3
[emmednikova@fedora lab11.3]$ ls
1.tmp 2.tmp 3 3.tmp
[emmednikova@fedora lab11.3]$ ./3 -r 3
[emmednikova@fedora lab11.3]$ ls
3
[emmednikova@fedora lab11.3]$
```

Рис. 2.6: Результат 3

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

```
#!/bin/bash

while getopts :p opt
do
    case $opt in
        p)dir="$OPTARG";;
        esac
    done

    find $dir -mtime -7 -mtime +0 -type f > arch.txt.

    tar -cf res.tar -T arch.txt
```

Рис. 2.7: Задание 4

```
[emmednikova@fedora lab11.4]$ ./4 -p /home/emmednikova/  
[emmednikova@fedora lab11.4]$ ls  
4 arch.txt res.tar  
[emmednikova@fedora lab11.4]$
```

Рис. 2.8: Результат 4

3 Выводы

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

4 Контрольные вопросы

1. Каково предназначение команды *getopts*?

Осуществляет синтаксический анализ командной строки, выделяя флаги (это опции командной строки, обычно помеченные знаком минус), и используется для объявления переменных. Синтаксис команды следующий:

```
getopts option-string variable [arg ... ]
```

2. Какое отношение метасимволы имеют к генерации имён файлов?

При перечислении имён файлов текущего каталога можно использовать следующие символы:

- * — соответствует произвольной, в том числе и пустой строке;
- ? — соответствует любому одинарному символу;
- [c1-c1] — соответствует любому символу, лексикографически находящемуся между символами c1 и c2.

Например, `echo *` — выведет имена всех файлов текущего каталога, что представляет собой простейший аналог команды `ls`;

– `ls *.c` — выведет все файлы с последними двумя символами, совпадающими с `.c`.

– `echo prog.?` — выведет все файлы, состоящие из пяти или шести символов, первыми пятью символами которых являются `prog.`

– `[a-z]*` — соответствует произвольному имени файла в текущем каталоге, начинающемуся с любой строчной буквы латинского алфавита.

3. *Какие операторы управления действиями вы знаете?*

For, case, if и while.

4. *Какие операторы используются для прерывания цикла?*

Break, continue.

5. *Для чего нужны команды false и true?*

Команда true, которая всегда возвращает код завершения, равный нулю (т.е. истина), и команда false, которая всегда возвращает код завершения, не равный нулю (т. е. ложь).

6. *Объясните различия между конструкциями while и until.*

Выполнение оператора цикла while сводится к тому, что сначала выполняется последовательность команд (операторов), которую задаёт список-команд в строке, содержащей служебное слово while, а затем, если последняя выполненная команда из этой последовательности команд возвращает нулевой код завершения (истина), выполняется последовательность команд (операторов), которую задаёт список-команд в строке, содержащей служебное слово do, после чего осуществляется безусловный переход на начало оператора цикла while. Выход из цикла будет осуществлён тогда, когда последняя выполненная команда из последовательности команд (операторов), которую задаёт список-команд в строке, содержащей служебное слово while, возвратит ненулевой код завершения (ложь). При замене в операторе цикла while служебного слова while на until условие, при выполнении которого осуществляется выход из цикла, меняется на противоположное. В остальном оператор цикла while и оператор цикла until идентичны.