

Отчёт по лабораторной работе №13

Программирование в командном процессоре ОС UNIX.
Ветвления и циклы

Фатима Халилова

Содержание

1 Цель работы	5
2 Выполнение лабораторной работы	6
3 Вывод	10
4 Контрольные вопросы	11

Список иллюстраций

2.1	Задание 1	7
2.2	Задание 2	8
2.3	Задание 3	9
2.4	Задание 4	9

Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

1. Используя команды getopt grep напишем командный файл, который анализирует командную строку с ключами и выполним его: -i inputfile — прочитать данные из указанного файла; -o outputfile — вывести данные в указанный файл; -р шаблон — указать шаблон для поиска; -С — различать большие и малые буквы; -n — выдавать номера строк;

а затем ищет в указанном файле нужные строки

```
#!/bin/bash
cflag=0;
nflag=0;
while getopt i:o:p:C:n opt
do
case $opt in
i) ival=$OPTARG;;
o) oval=$OPTARG;;
p) pval=$OPTARG;;
C) cflag=1;;
n) nflag=1;;
esac
done
if [ $cflag -a $nflag ]
then
```

```

grep -n $pval $ival>$oval
elif test $cflag
then
grep $pval $ival>$oval
elif test $nflag
then
grep -n -i $pval $ival>$oval
else
grep -i $pval $ival>$oval
fi

```

```

frhalolova@frhalilova:~/work/study/2025-2026/Операционные системы/os-intro/labs/lab1
3$ chmod +x lab13_1.sh
frhalolova@frhalilova:~/work/study/2025-2026/Операционные системы/os-intro/labs/lab1
3$ ./lab13_1.sh -i text.txt -o fout.txt -p файлы -c -
frhalolova@frhalilova:~/work/study/2025-2026/Операционные системы/os-intro/labs/lab1
3$ cat fout.txt
10:файлы,
16:файла, проецируемые в память.
frhalolova@frhalilova:~/work/study/2025-2026/Операционные системы/os-intro/labs/lab1
3$ 

```

```

#!/bin/bash
# lab13_1.sh
1#!/bin/bash
2 cflag=0;
3 nflag=0;
4 while getopts i:o:p:C:n opt
5 do
6 case $opt in
7   i) ivl=$OPTARG;;
8   o) val=$OPTARG;;
9   p) pval=$OPTARG;;
10  C) cflag=1;;
11  n) nflag=1;;
12 esac
13 done
14 if [ $cflag -a $nflag ]
15 then
16 grep -n $pval $ival>$oval
17 elif test $cflag
18 then
19 grep $pval $ival>$oval
20 elif test $nflag
21 then
22 grep -n -i $pval $ival>$oval
23 else
24 grep -i $pval $ival>$oval
25 fi

```

Рисунок 2.1: Задание 1

2. Напишем сначала на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем завершим программу при помощи функции exit(n), передавая информацию о коде завершения в оболочку. Командный файл вызовет эту программу и, проанализировав с помощью команды \$?, выдаст сообщение о том, какое число было введено

```

#!/bin/bash
gcc -c script2.c

```

```

gcc -o script2 script2.c
./script2
case $? in
    1) echo отрицательное;;
    2) echo равно нулю;;
    3) echo положительное;;
esac

```

Рисунок 2.2: Задание 2

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

```

#!/bin/bash
let i=$1+1
while (( i-=1 ))
do touch $i.tmp
done
let j=$2+1;
while (( j-=1 ))
do rm $j.tmp
done

```

The left terminal window shows the command `./lab13_3.sh 5 2` being run, followed by an `ls` command showing files `fout.txt`, `lab13_3.sh`, `report`, and `script2.o`. The right terminal window shows the script `lab13_3.sh` with its code.

```

#!/bin/bash
2 let i=$1+
3 while (( i>=1 ))
4 do touch $i.tmp
5 done
6 let j=$2+1;
7 while (( j>=1 ))
8 do rm $j.tmp
9 done

```

Рисунок 2.3: Задание 3

4. Напишем командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории. Модифицируем его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад.

```

#!/bin/bash
(find $1 -mtime -7 -daystart) | xargs tar -cf arhiv.tar

```

The left terminal window shows the command `./lab13_4.sh` being run, followed by an `ls` command showing files `arhiv.tar`, `fout.txt`, `lab13_3.sh`, `report`, and `script2.o`. The right terminal window shows the script `lab13_4.sh` with its code.

```

#!/bin/bash
2 (find $1 -mtime -7 -daystart) | xargs tar -cf arhiv.tar

```

Рисунок 2.4: Задание 4

3 Вывод

В данной работе мы изучили основы программирования в оболочке ОС UNIX и писать более сложные командные файлы с использованием логических управляемых конструкций и циклов.

4 Контрольные вопросы

1. Каково предназначение команды getopt? Ответ: Создание по пользовательским аргументам.
2. Какое отношение метасимволы имеют к генерации имён файлов? Ответ: Используют как файлы так и аргументы.
3. Какие операторы управления действиями вы знаете? Ответ: If, else, elif, fi, while, do, done, until, do, done, for, in, do, done, case, in, esac
4. Какие операторы используются для прерывания цикла? Ответ:
 - a) for – будет выполнять действие до тех пор, пока есть объекты для выполнения.
 - b) while – выполняет действие до тех пор, пока условие является истинным.
 - c) until – будет выполняться пока условие не станет правдиво.
5. Для чего нужны команды false и true? Ответ: until – будет выполняться до тех пор, пока условие не станет true, т.е. пока оно не станет false.
6. Что означает строка if test -f mans/i.\$s, встреченная в командном файле? Ответ: Проверяет если существует файл его размерность и тип с двумя разными расширениями, заменяя через переменные.

7. Объясните различия между конструкциями while и until. Ответ:

while – выполняет действие до тех пор, пока условие является истинным.

until – будет выполняться до тех пор, пока условие не станет истинным, т.е. пока оно false.