Отчёт по лабораторной работе №13

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Когенгар Ришард

Содержание

Список иллюстраций

Список таблиц

# 1 Цель работы

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

# 2 Выполнение лабораторной работы

1. Используя команды getopts 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; -p шаблон — указать шаблон для поиска; -C — различать большие и малые буквы; -n — выдавать номера строк;

а затем ищет в указанном файле нужные строки

#!/bin/bash  
cflag=0;  
nflag=0;  
while getopts 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

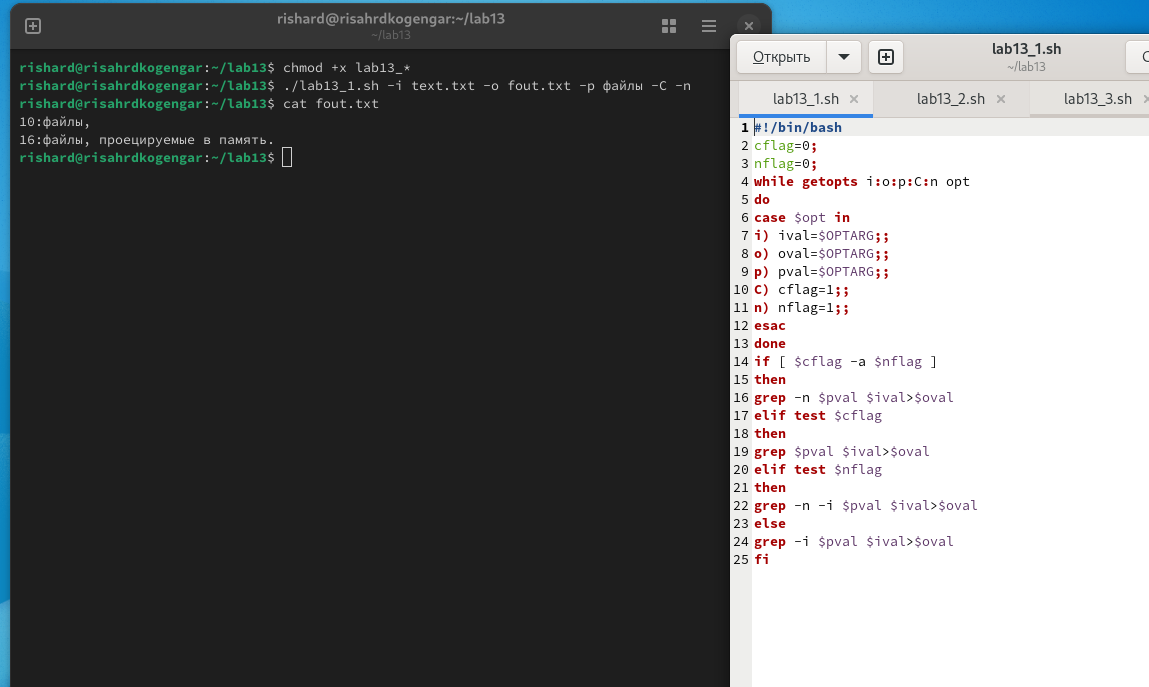


Рис. 1: Задание 1

1. 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

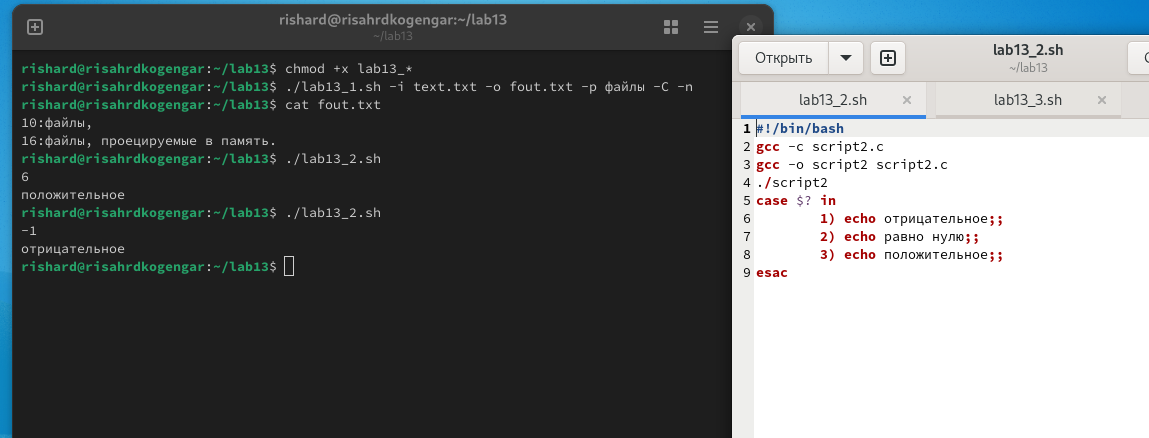


Рис. 2: Задание 2

1. 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

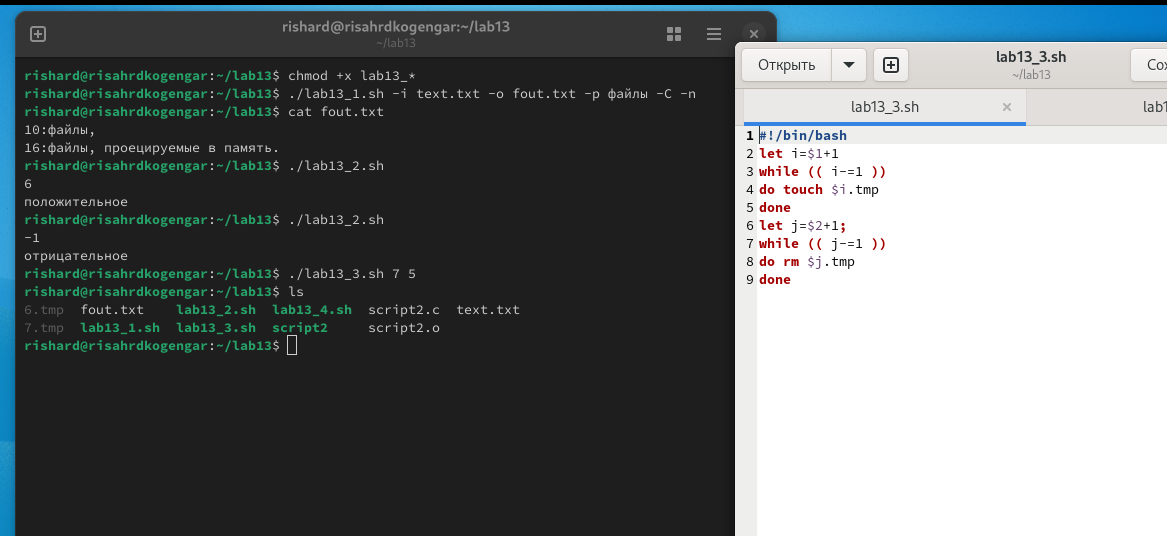


Рис. 3: Задание 3

1. 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

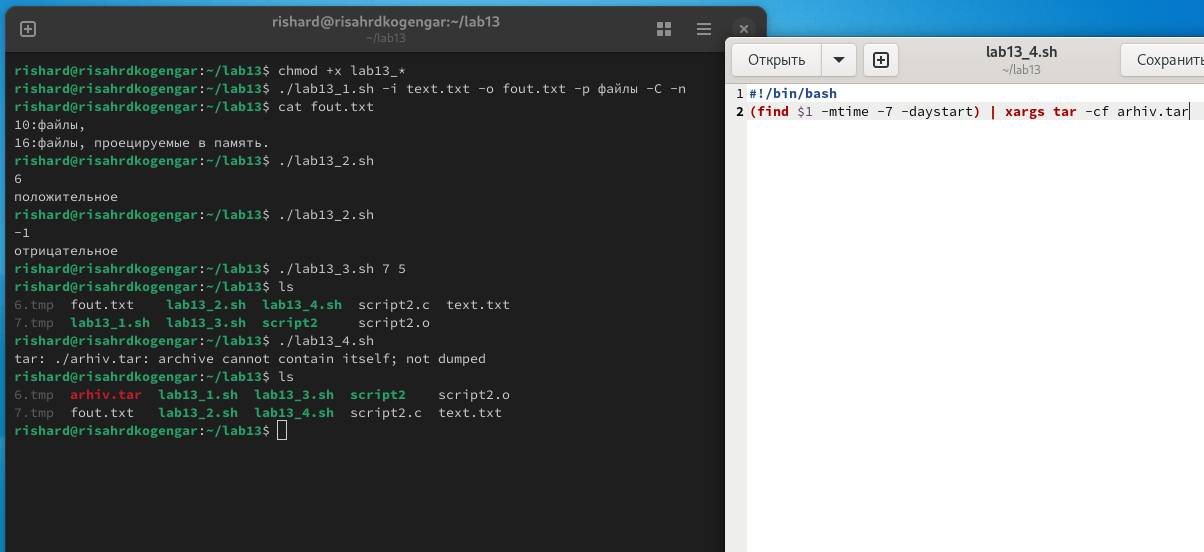


Рис. 4: Задание 4

# 3 Вывод

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

# 4 Контрольные вопросы

1. Каково предназначение команды getopts? 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:
5. for – будет выполнять действие до тех пор, пока есть объекты для выполнения.
6. while – выполняет действие до тех пор, пока условие является истинным.
7. until – будет выполнятся пока условие не станет правдиво.
8. 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.
9. Что означает строка if test -f mani.$s, встреченная в командном файле? Ответ: Проверяет если существует файл его размерность и тип c двумя разными расширениями, заменяя через переменные.
10. 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.