

Отчёт по лабораторной работе №13

Операционные системы

Бережной Иван Александрович

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Написание первого командного файла	7
3.2	Написание второго командного файла и программы на Си	10
3.3	Написание третьего командного файла	11
3.4	Написание четвёртого командного файла	12
4	Выводы	14
	Список литературы	15

Список иллюстраций

3.1	Первый скрипт	8
3.2	Текстовый файл	9
3.3	Запуск первого скрипта	9
3.4	Результат первого скрипта	9
3.5	Программа на Си	10
3.6	Второй скрипт	11
3.7	Проверка работы файлов	11
3.8	Третий скрипт	12
3.9	Проверка работы скрипта	12
3.10	Четвёртый скрипт	12
3.11	Запуск четвёртого скрипта	13

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 Задание

1. Написать командный файл, ищущий нужные строки
2. Написать программу на языке Си, определяющую, какое число было введено, и командный файл, вызывающий эту программу
3. Написать командный файл, создающий указанное количество файлов, пронумерованных последовательно от 1 до N
4. Написать командный файл, который сжимаем указанные файлы в архив

3 Выполнение лабораторной работы

3.1 Написание первого командного файла

Создадим файл в терминале и присвоим ему право на исполнение. Напишем скрипт, который анализирует командную строку с ключами и ищет в указанном файле нужные строки, определённые после ключа -р (рис. 3.1). Создадим текстовый файл, в котором будем искать строки (рис. 3.2). Запустим скрипт (рис. 3.3) и получим вот такой результат (рис. 3.4).

```

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 Illegal option $optletter;;
    esac
done

if ! test $cflag
then
    cf=-i
fi

if test $nflag
then
    nf=-n
fi

grep $cf $nf $pval $ival >> $oval

```

Рис. 3.1: Первый скрипт

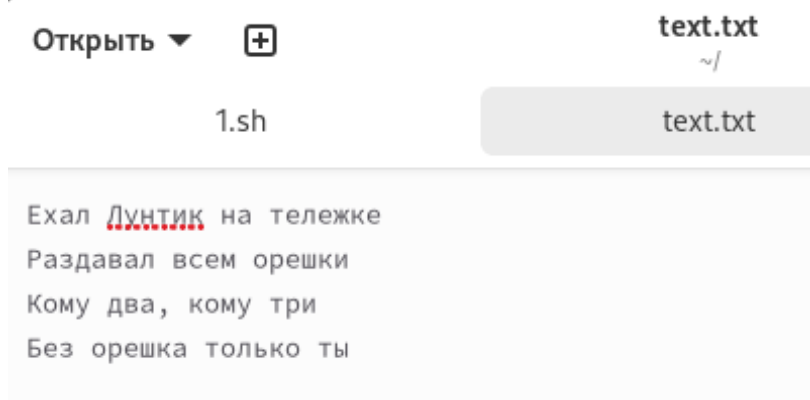


Рис. 3.2: Текстовый файл

```
laberezhnoy@laberezhnoy:~$ bash 1.sh -p ореш -i text.txt -o result.txt -c -n  
laberezhnoy@laberezhnoy:~$
```

Рис. 3.3: Запуск первого скрипта

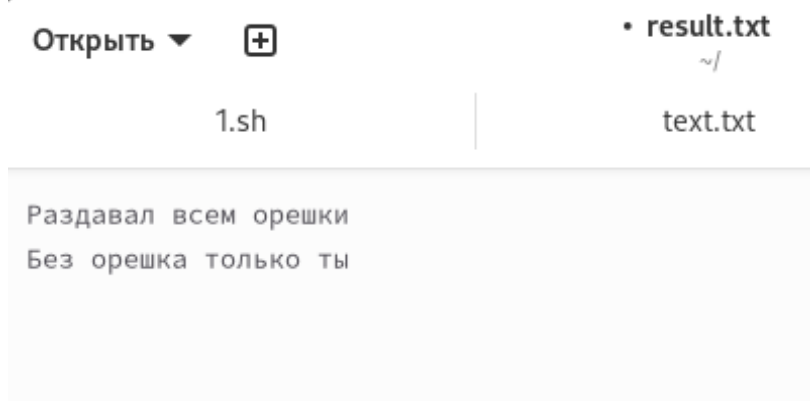


Рис. 3.4: Результат первого скрипта

3.2 Написание второго командного файла и программы на Си

Создадим второй командный файл и файл с расширением .c. В последнем напомним команду, которая будет проверять, является ли число положительным, отрицательным или нулём (рис. 3.5). Напишем скрипт, который будет вызывать этот файл и выводить в терминал результат в виде текста (рис. 3.6). Проверим работу файлов (рис. 3.7).



```
Открыть ▼  + 22.c ~/
22.cpp 2.sh

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

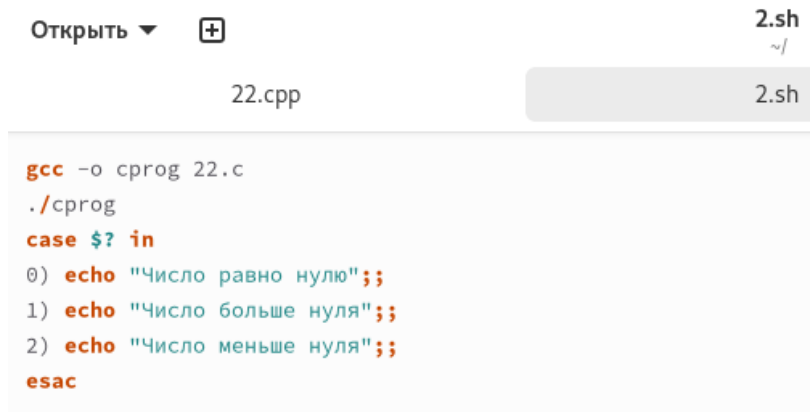
int main()
{
    int n;
    printf ("Введите число: ");
    scanf ("%d", &n);

    if(n>0)
    {
        exit(1);
    }

    else if (n==0)
    {
        exit(0);
    }

    else
    {
        exit(2);
    }
}
```

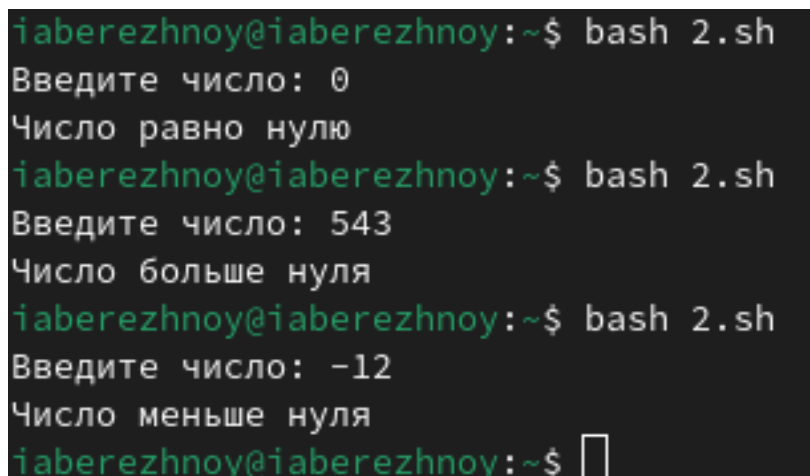
Рис. 3.5: Программа на Си



```
Открыть ▾ + 2.sh ~/
22.cpp 2.sh

gcc -o cprog 22.c
./cprog
case $? in
0) echo "Число равно нулю";;
1) echo "Число больше нуля";;
2) echo "Число меньше нуля";;
esac
```

Рис. 3.6: Второй скрипт



```
iaberezhnoy@iaberezhnoy:~$ bash 2.sh
Введите число: 0
Число равно нулю
iaberezhnoy@iaberezhnoy:~$ bash 2.sh
Введите число: 543
Число больше нуля
iaberezhnoy@iaberezhnoy:~$ bash 2.sh
Введите число: -12
Число меньше нуля
iaberezhnoy@iaberezhnoy:~$
```

Рис. 3.7: Проверка работы файлов

3.3 Написание третьего командного файла

Теперь напишем командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N, при этом количество файлов будет передаваться в аргументы командной строки (рис. 3.8).

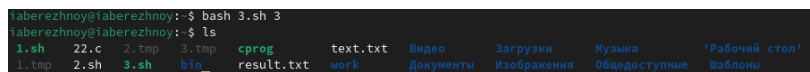
Проверим, работает ли скрипт (рис. 3.9).



Открыть ▾ + 3.sh
~/

```
for((i=1; i<=$*; i++))  
do  
if test -f "$i".tmp  
then rm "$i".tmp  
else touch "$i.tmp"  
fi  
done
```

Рис. 3.8: Третий скрипт



```
laberezhnoy@laberezhnoy:~$ bash 3.sh 3  
laberezhnoy@laberezhnoy:~$ ls  
1.sh 22.c 2.tmp 3.tmp cprog text.txt Видео Загрузки Музыка 'Рабочий стол'  
1.tmp 2.sh 3.sh bin result.txt work Документы Изображения Общедоступные Шаблоны
```

Рис. 3.9: Проверка работы скрипта

3.4 Написание четвёртого командного файла

И, наконец, напомним командный файл, который с помощью команды `tar` запаковывает в архив все файлы в указанной директории (рис. 3.10). Запустим скрипт через командную строку (рис. 3.11).



Открыть ▾ + 4.sh
~/

```
find $* -mtime -7 -mtime +0 -type f > files.txt  
tar -cf archive.tar -T files.txt
```

Рис. 3.10: Четвёртый скрипт

```

iaberezhnoy@iaberezhnoy:~$ bash 4.sh /home/iaberezhnoy
tar: Удаляется начальный '/' из имен объектов
tar: Удаляются начальные '/' из целей жестких ссылок
iaberezhnoy@iaberezhnoy:~$ la
bash: la: команда не найдена...
ls
iaberezhnoy@iaberezhnoy:~$ ls
1.sh      2.sh      3.tmp      bin          result.txt  Видео      Изображения  'Рабочий стол'
1.tmp     2.tmp     4.sh      cprog        text.txt    Документы  Музыка      Шаблоны
22.c     3.sh     archive.tar  files.txt   work        Загрузки   Общедоступные
iaberezhnoy@iaberezhnoy:~$

```

Рис. 3.11: Запуск четвёртого скрипта

4 Выводы

В ходе выполнения лабораторной работы мы изучили основы программирования в оболочке ОС UNIX и научились писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Список литературы