

Лабораторная работа № 11

Операционные системы

Голованова Мария Константиновна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	8
4	Выводы	13
	Список литературы	14

Список иллюстраций

3.1	Программа №1	9
3.2	Программа №2	10
3.3	Программа №3	11
3.4	Программа №4	12

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. Используя команды `getopts` `grep`, написать командный файл, который анализирует командную строку с ключами:
 - `-iinputfile` — прочитать данные из указанного файла;
 - `-ooutputfile` — вывести данные в указанный файл;
 - `-р`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 `-р`.
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).

3 Выполнение лабораторной работы

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

- `-iinputfile` — прочитать данные из указанного файла;
- `-ooutputfile` — вывести данные в указанный файл;
- `-р`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 `-p` (рис. 3.1).


```

file-1.sh [----] 24 L:[ 1+14 15/ 15] *
#!/bin/bash

while getopts i:o:p:C:n optletter
do case $optletter in
i) iflag=1; ival=OPTARG;;
o) oflag=1; oval=OPTARG;;
p) pflag=1; pval=OPTARG;;
C) Cflag="i";;
n) nflag=1; nval=OPTAGS;;

*) echo "Illegal option $optletter"
esac
done

grep $nval $Cflag $pval $oval $ival

```

Рис. 3.1: Программа №1

2. Я написала 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 вызывает эту программу и, проанализировав с помощью команды `$?`, выдаёт сообщение о том, какое число было введено (рис. 3.2).

```

file-2.c [----] 17 L:[ 1+ 0 1/
#include<stdio.h>
#include<stdlib.h>

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

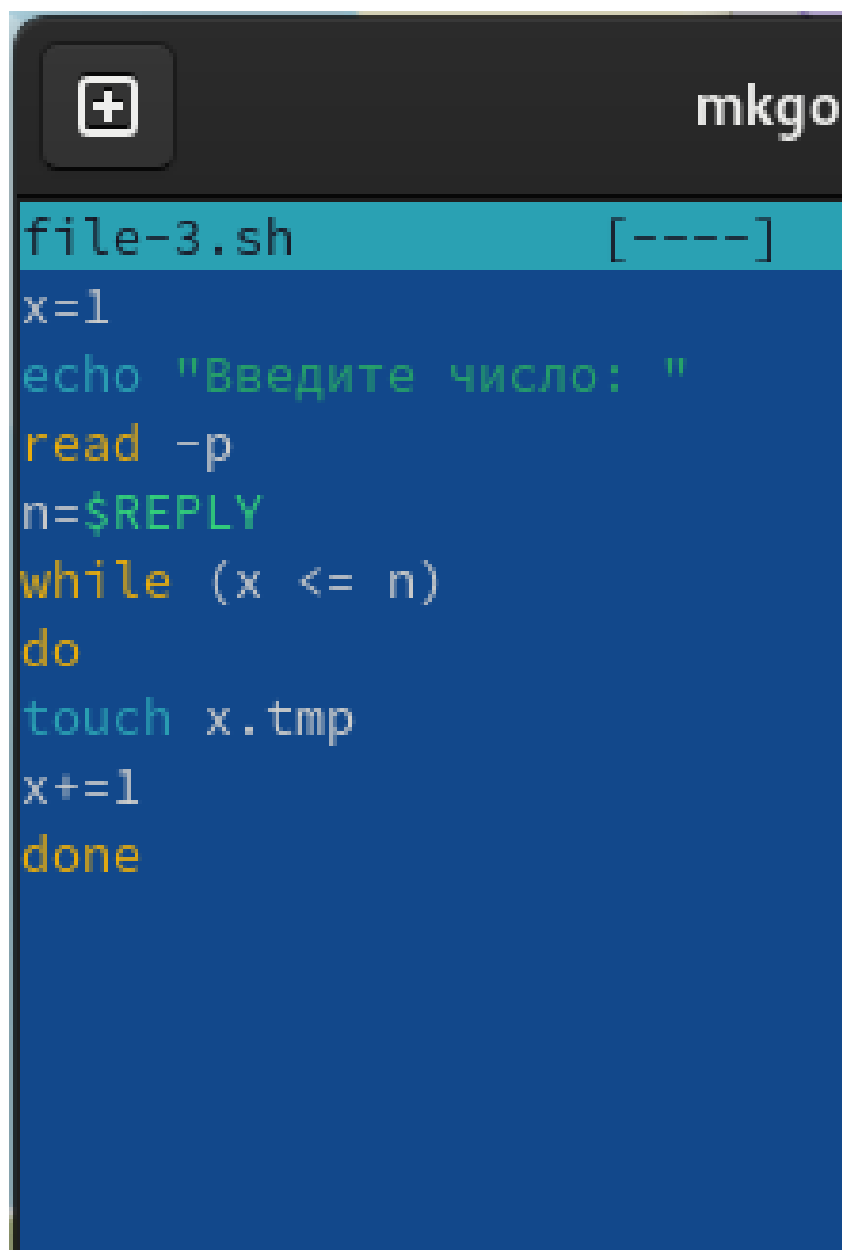
    if (x > 0) {
        n = 1;
    }
    else if (x < 0) {
        n = 2;
    }
    else {
        n = 0;
    }
    exit(n);

    return 0;
}

```

Рис. 3.2: Программа №2

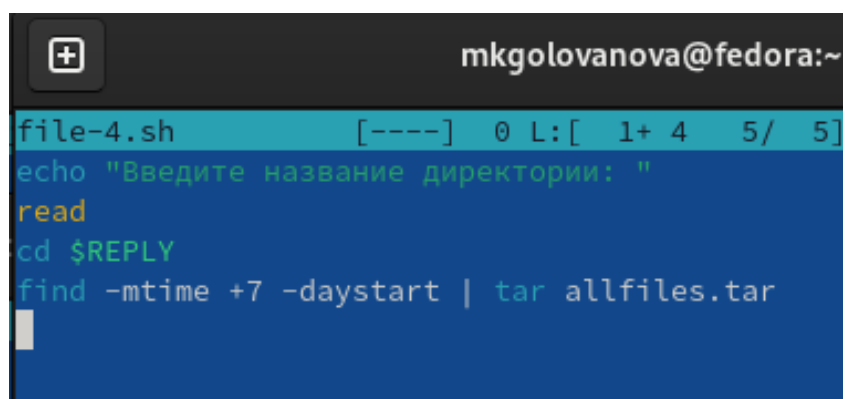
3. Я написала командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N (например 1.tmp, 2.tmp, 3.tmp, 4.tmp и т.д.). Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же командный файл удаляет все созданные им файлы (если они существуют) (рис. 3.3).



```
file-3.sh [-----]
x=1
echo "Введите число: "
read -p
n=$REPLY
while (x <= n)
do
touch x.tmp
x+=1
done
```

Рис. 3.3: Программа №3

4. Я написала командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории, и модифицировала его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовать команду find) (рис. 3.4).



```
file-4.sh [-----] 0 L:[ 1+ 4 5/ 5]
echo "Введите название директории: "
read
cd $REPLY
find -mtime +7 -daystart | tar allfiles.tar
```

Рис. 3.4: Программа №4

4 Выводы

Я изучила основы программирования в оболочке ОС UNIX и научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Список литературы