

Отчет по лабораторной работе №11

дисциплина Операционные системы

Колобова Елизавета, гр. НММбд-01-22

22 марта 2023

Российский университет дружбы народов, Москва, Россия

Информация

- Колобова Елизавета
- студент
- Российский университет дружбы народов

Вводная часть

- программирование в оболочке ОС UNIX/Linux.

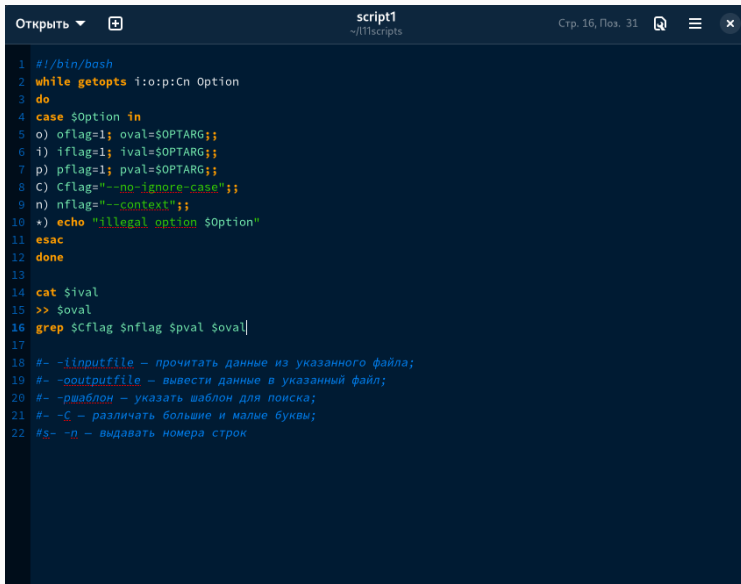
- изучить основы программирования в оболочке ОС UNIX/Linux.
- Научиться писать сложные командные файлы с использованием логических управляющих конструкций и циклов.

- Процессор **pandoc** для входного формата Markdown
- Автоматизация процесса создания: **Makefile**

Создание презентации

1. Используя команды `getopts` `grep`, напомним командный файл, который анализирует командную строку с ключами: – `-iinputfile` — прочитать данные из указанного файла; – `-ooutputfile` — вывести данные в указанный файл; – `-rшаблон` — указать шаблон для поиска; – `-C` — различать большие и малые буквы; – `-n` — выдавать номера строк. а затем ищет в указанном файле нужные строки, определяемые ключом `-r`.





Скрипт поиска по шаблону в задаваемом файле



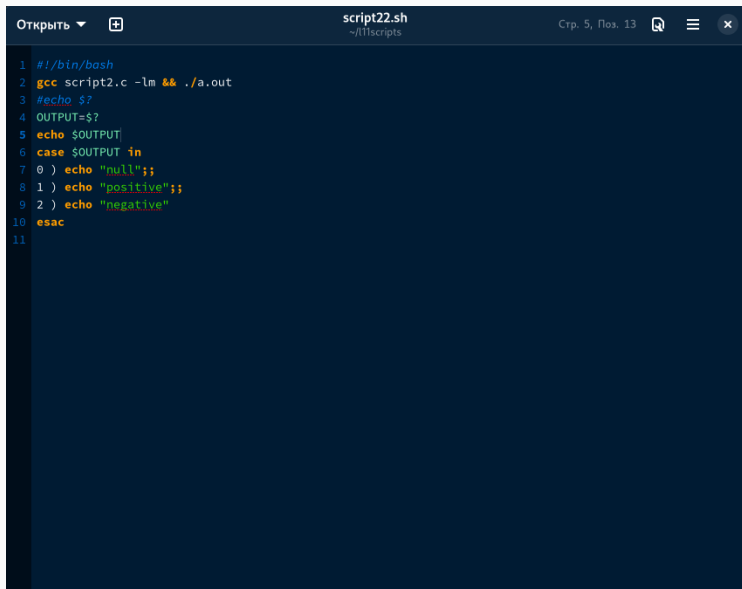
```
Открыть ▾ + script1 ~/l1scripts Стр. 16, Поз. 31 🔍 ☰ ✕  
1 #!/bin/bash  
2 while getopts i:o:p:Cn Option  
3 do  
4 case $Option in  
5 o) oflag=1; oval=$OPTARG;;  
6 i) iflag=1; ival=$OPTARG;;  
7 p) pflag=1; pval=$OPTARG;;  
8 C) Cflag="--no-ignore-case";;  
9 n) nflag="--context";;  
10 *) echo "illegal option $Option"  
11 esac  
12 done  
13  
14 cat $ival  
15 >> $oval  
16 grep $Cflag $nflag $pval $oval  
17  
18 #- iinputfile — прочитать данные из указанного файла;  
19 #- ooutputfile — вывести данные в указанный файл;  
20 #- ршаблон — указать шаблон для поиска;  
21 #- C — различать большие и малые буквы;  
22 #- s -n — выдавать номера строк
```

2. Напишем 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. Команд- ный файл вызывает эту программу и, проанализировав с помощью команды `$?`, выдать сообщение о том, какое число было введено. (положительное, отрицательное или нуль)

Программа для сравнения числа с нулем на Си

```
Открыть ▾  script2.c  
~/l11scripts Стр. 11, Поз. 10     
  
1 #include <stdio.h>  
2 #include <math.h>  
3 #include <stdlib.h>  
4  
5 int main(){  
6  
7     int n;  
8     int res;  
9     scanf("Input the number: %i", &n);  
10  
11     if (n == 0){  
12         res = 0;  
13     }  
14     else if (n > 0){  
15         res = 1;  
16     }  
17     else if (n < 0){  
18         res = 2;  
19     }  
20     exit(res);  
21 }
```

Программа для сравнения числа с нулем на Си

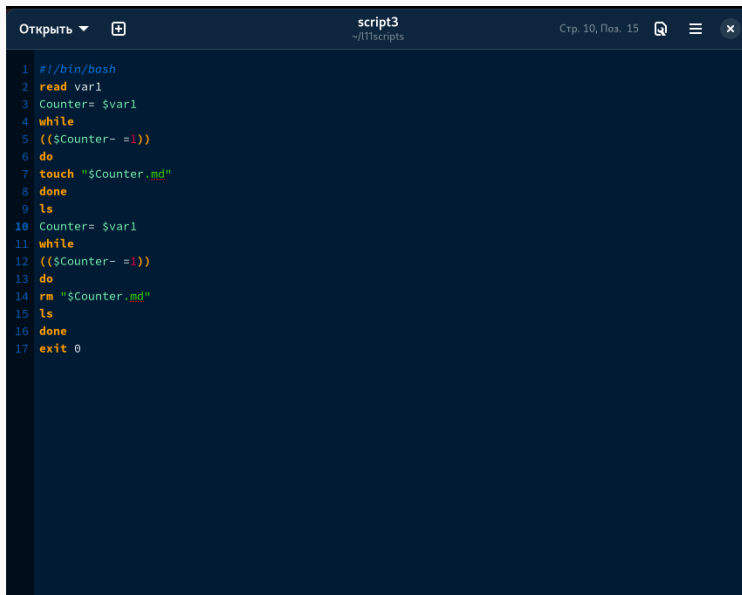


The image shows a terminal window with a dark blue background. The title bar at the top reads "script22.sh" and "~/.11scripts". On the left, there is a button labeled "Открыть" (Open) with a plus icon. On the right, there is a status bar showing "Стр. 5, Поз. 13" (Page 5, Position 13) and icons for search, menu, and close. The main area contains a C script with line numbers 1 through 11. The script uses `gcc` to compile `script2.c` into `a.out`, then uses `echo` to print the result of a comparison. The comparison logic is implemented using a `case` statement that checks for `0`, `1`, and `2`, and prints `null`, `positive`, or `negative` accordingly.

```
1 #!/bin/bash
2 gcc script2.c -lm && ./a.out
3 #echo $?
4 OUTPUT=$?
5 echo $OUTPUT
6 case $OUTPUT in
7 0 ) echo "null";;
8 1 ) echo "positive";;
9 2 ) echo "negative"
10 esac
11
```

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

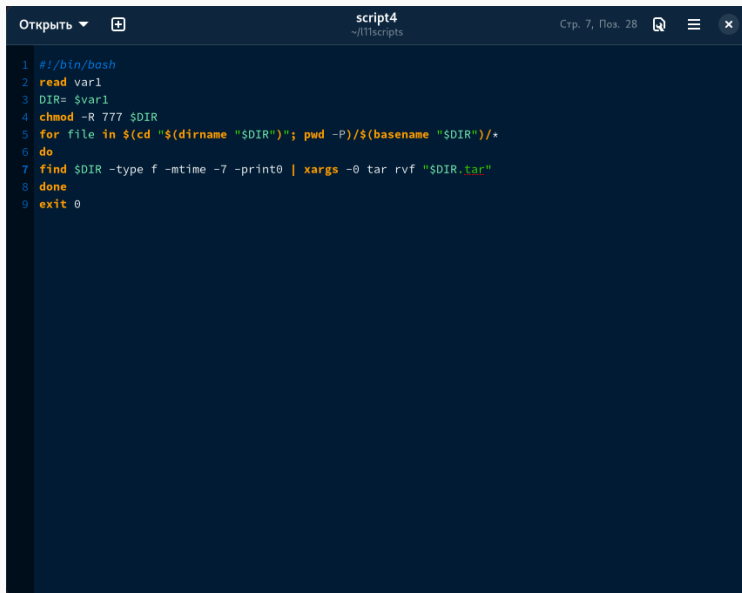
Скрипт создания заданного количества файлов



```
Открыть ▾ + script3 ~/l11scripts Стр. 10, Поз. 15 🔍 ☰ ✕  
1  #!/bin/bash  
2  read var1  
3  Counter= $var1  
4  while  
5  (($Counter- =1))  
6  do  
7  touch "$Counter.md"  
8  done  
9  ls  
10 Counter= $var1  
11 while  
12 (($Counter- =1))  
13 do  
14 rm "$Counter.md"  
15 ls  
16 done  
17 exit 0
```

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

Скрипт архивирования файлов указанной директории



The image shows a terminal window with a dark blue background. The title bar at the top reads "script4" and "~/.l11scripts". On the left, there is a button labeled "Открыть" (Open) with a plus icon. On the right, there is a status bar showing "Стр. 7, Пос. 28" (Page 7, Line 28) and icons for search, menu, and close. The terminal displays a shell script with the following content:

```
1 #!/bin/bash
2 read var1
3 DIR= $var1
4 chmod -R 777 $DIR
5 for file in $(cd "$(dirname "$DIR")"; pwd -P)/$(basename "$DIR")/*
6 do
7 find $DIR -type f -mtime -7 -print0 | xargs -0 tar rvf "$DIR.tar"
8 done
9 exit 0
```

Результаты

- Результатом проделанной работы является изучение основ программирования в оболочке ОС UNIX/Linux.