

Лабораторная работа №11

Операционные системы

Сабралиева Марворид Нуралиевна

Содержание

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

Список иллюстраций

2.1	задание 1	7
2.2	задание 2	8
2.3	задание 3	8
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 управляющих конструкций и циклов.

2 Выполнение лабораторной работы

1. Используя команды `getopts` `grep`, напишите командный файл, который анализирует командную строку с ключами: `-i`inputfile — прочитать данные из указанного файла; `-o`outputfile — вывести данные в указанный файл; `-r`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 `-r`. (рис. 2.1).

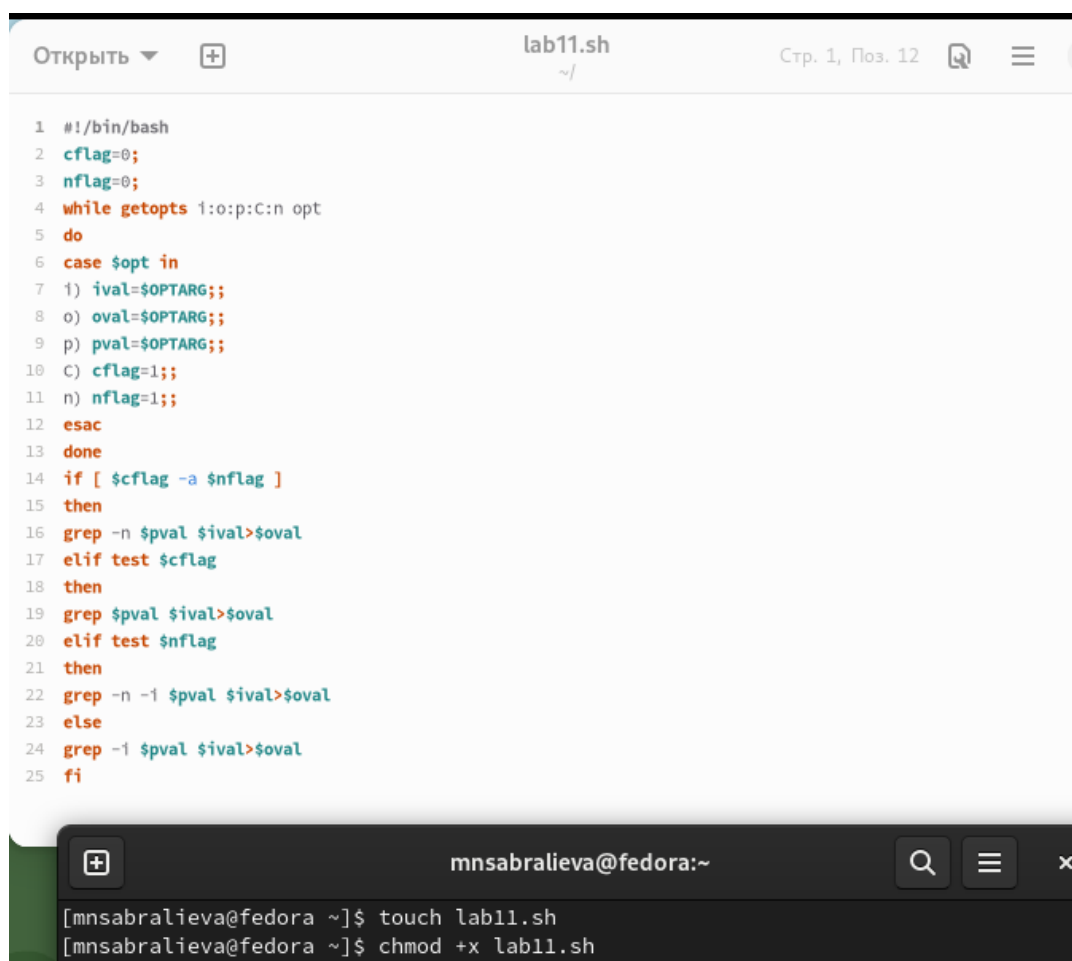


Рис. 2.1: задание 1

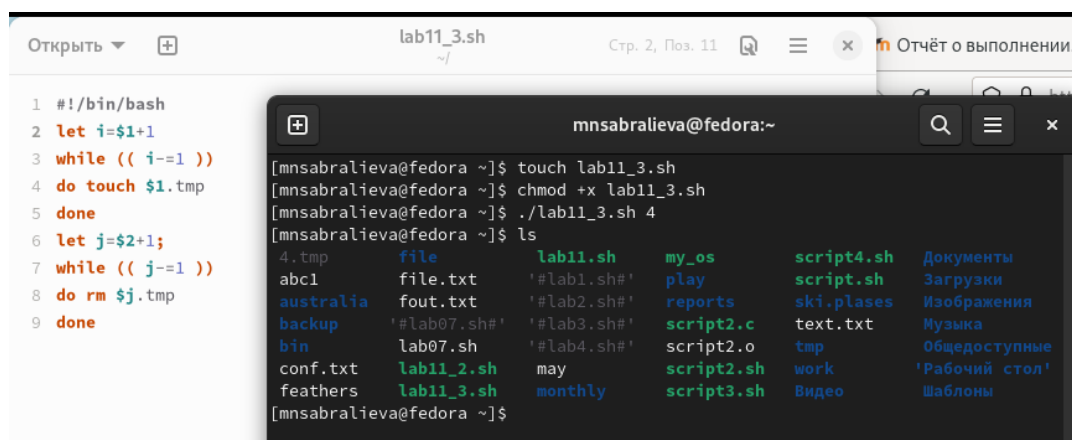
2. Напишем на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем программа завершается с помощью функции `exit(n)`, передавая информацию в о коде завершения в оболочку. Командный файл должен вызывать эту программу и, проанализировав с помощью команды `$?`, выдать сообщение о том, какое число было введено.(рис. 2.2).



```
1 #!/bin/bash
2 gcc -c script2.c
3 gcc -o script2 script2.c
4 ./script2
5 case $? in
6     1) echo положительное;;
7     2) echo равно нулю;;
8     3) echo отрицательное;;
9 esac
```

Рис. 2.2: задание 2

3. Напишем командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до ∞ (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).(рис. 2.3).



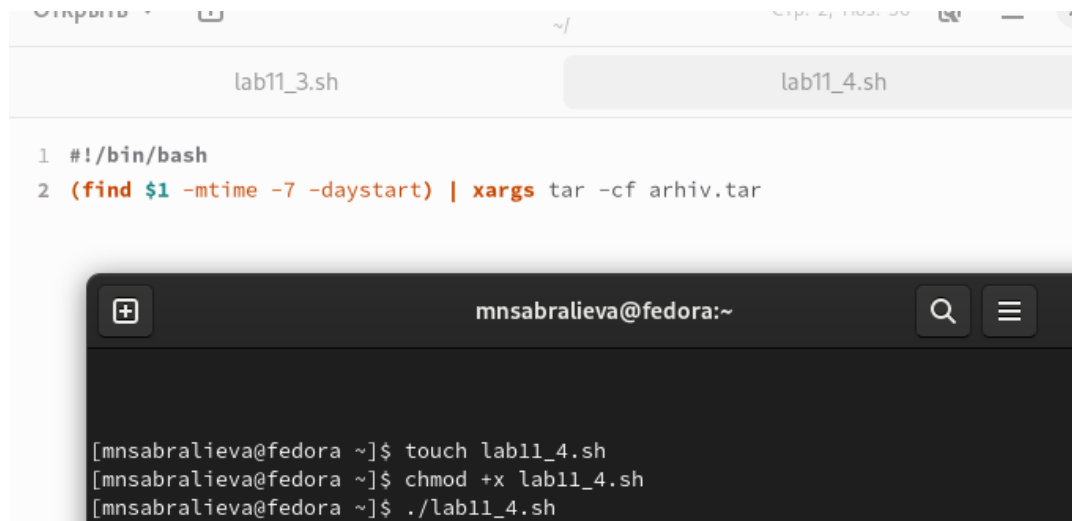
```
1 #!/bin/bash
2 let i=$1+1
3 while (( i--=1 ))
4 do touch $1.tmp
5 done
6 let j=$2+1;
7 while (( j--=1 ))
8 do rm $j.tmp
9 done
```

```
[mnsabralieva@fedora ~]$ touch lab11_3.sh
[mnsabralieva@fedora ~]$ chmod +x lab11_3.sh
[mnsabralieva@fedora ~]$ ./lab11_3.sh 4
[mnsabralieva@fedora ~]$ ls
4.tmp      file      lab11.sh  my_os    script4.sh  Документы
abc1       file.txt  '#lab1.sh#'  play     script.sh   Загрузки
australia  fout.txt  '#lab2.sh#'  reports  ski.plases  Изображения
backup     '#lab07.sh#'  '#lab3.sh#'  script2.c  text.txt    Музыка
bin        lab07.sh  '#lab4.sh#'  script2.o  tmp         Общедоступные
conf.txt   lab11_2.sh  may         script2.sh  work        'Рабочий стол'
feathers    lab11_3.sh  monthly     script3.sh  Видео       Шаблоны
[mnsabralieva@fedora ~]$
```

Рис. 2.3: задание 3

4. Напишем 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). (рис. 2.4).



The image shows a terminal window with two tabs: 'lab11_3.sh' and 'lab11_4.sh'. The active tab 'lab11_4.sh' contains the following script:

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

Below the script, the terminal shows the execution of the script with the following commands and output:

```
[mnsabralieva@fedora ~]$ touch lab11_4.sh
[mnsabralieva@fedora ~]$ chmod +x lab11_4.sh
[mnsabralieva@fedora ~]$ ./lab11_4.sh
```

Рис. 2.4: задание 4

3 Выводы

В данной работе мы изучили основы программирования в оболочке ОС UNIX. Также научились 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:
 - 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 man/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`. # Список литературы{.unnumbered}