

Лабораторная работа №10

Основы программирования в оболочке ОС UNIX/Linux.

Извекова Мария Петровна

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	7
4	Выводы	12
	Список литературы	13

Список иллюстраций

3.1	backup	7
3.2	code1.sh	7
3.3	папка backup	8
3.4	наш файл с кодом	8
3.5	команда bash	8
3.6	code 2	9
3.7	вывод на экран	9
3.8	code 3	10
3.9	вывод третьего кода	10
3.10	code 4	11
3.11	вывод четвертого кода	11

Список таблиц

1 Цель работы

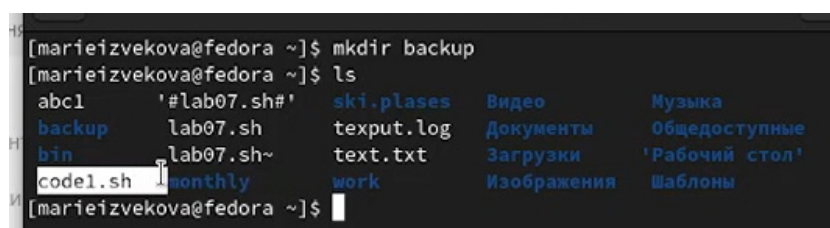
Изучить основы программирования в оболочке ОС UNIX/Linux. Научиться писать небольшие командные файлы.

2 Теоретическое введение

Командный процессор (командная оболочка, интерпретатор команд shell) — это программа, позволяющая пользователю взаимодействовать с операционной системой компьютера. В операционных системах типа UNIX/Linux наиболее часто используются следующие реализации командных оболочек: – оболочка Борна (Bourne shell или sh) — стандартная командная оболочка UNIX/Linux, содержащая базовый, но при этом полный набор функций; – C-оболочка (или csh) — надстройка на оболочке Борна, использующая C-подобный синтаксис команд с возможностью сохранения истории выполнения команд; – оболочка Корна (или ksh) — напоминает оболочку C, но операторы управления программой совместимы с операторами оболочки Борна; – BASH — сокращение от Bourne Again Shell (опять оболочка Борна), в основе своей совмещает свойства оболочек C и Корна (разработка компании Free Software Foundation). POSIX (Portable Operating System Interface for Computer Environments) — набор стандартов описания интерфейсов взаимодействия операционной системы и прикладных программ. Стандарты POSIX разработаны комитетом IEEE (Institute of Electrical and Electronics Engineers) для обеспечения совместимости различных UNIX/Linux-подобных операционных систем и переносимости прикладных программ на уровне исходного кода. POSIX-совместимые оболочки разработаны на базе оболочки Корна. Рассмотрим основные элементы программирования в оболочке bash.

3 Выполнение лабораторной работы

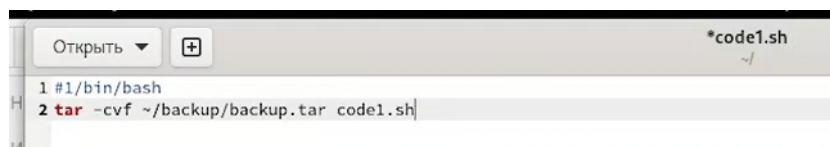
1. создаем папку backup, куда мы будем копировать первый файл code1.sh



```
[marieizvekova@fedora ~]$ mkdir backup
[marieizvekova@fedora ~]$ ls
abc1      '#lab07.sh#'  ski.plases  Видео      Музыка
backup    lab07.sh      texput.log  Документы  Общедоступные
bin       lab07.sh~     text.txt    Загрузки   'Рабочий стол'
code1.sh  monthly      work        Изображения Шаблоны
[marieizvekova@fedora ~]$
```

Рис. 3.1: backup

В файле code1.sh пишем следующий код, который создаст нам директорию сжатой папкой и поместит туда наш первый файл с кодом



```
Открыть ▼ + *code1.sh ~/
1 #!/bin/bash
2 tar -cvf ~/backup/backup.tar code1.sh
```

Рис. 3.2: code1.sh

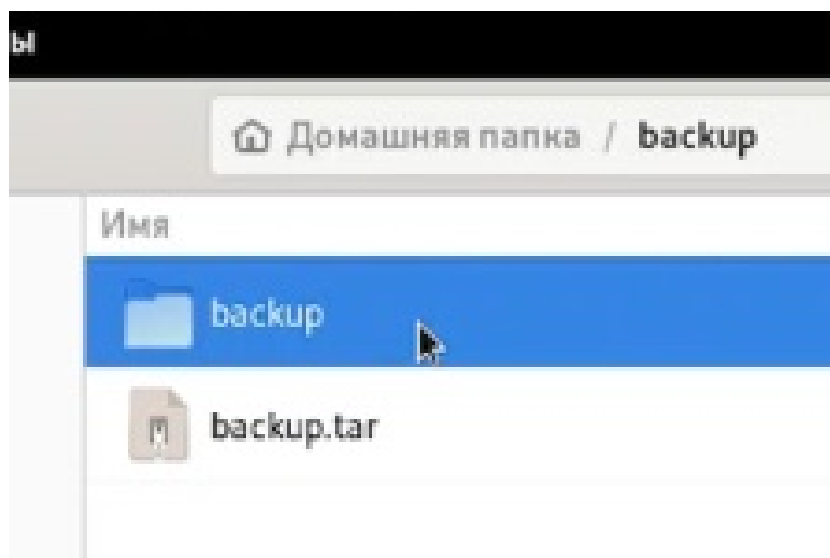


Рис. 3.3: папка backup

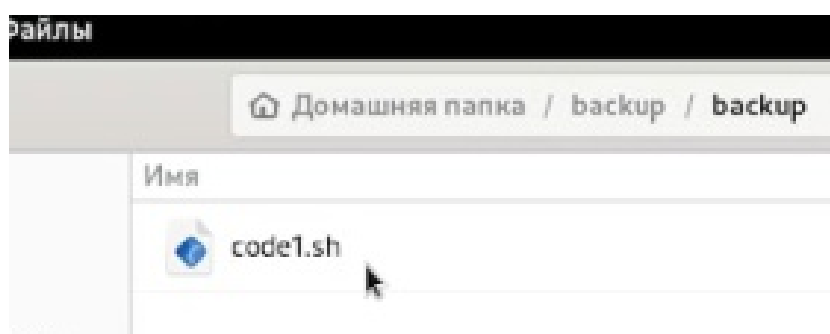


Рис. 3.4: наш файл с кодом

чтобы у нас появился этот файл в нашей директории, мы забиваем следующую команду

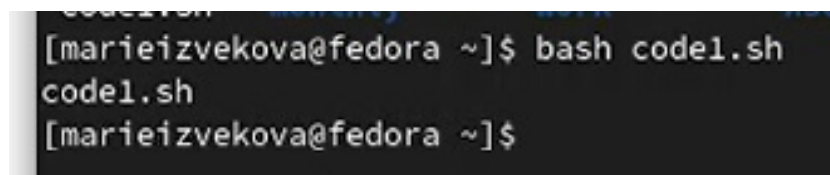
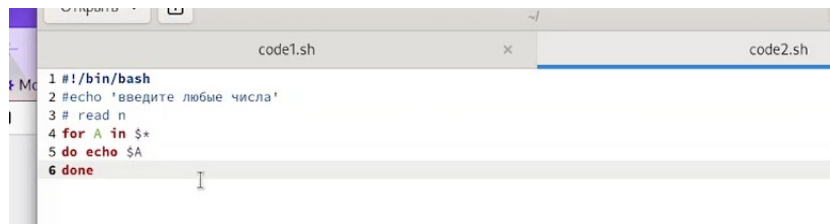


Рис. 3.5: команда bash

2. для вывода следующих чисел мы пишем следующий код с циклом (чтобы он считывал и выводил наши числа) в наш новый файл с тем же расширением



```
1 #!/bin/bash
2 #echo 'введите любые числа'
3 # read n
4 for A in $*
5 do echo $A
6 done
```

Рис. 3.6: code 2

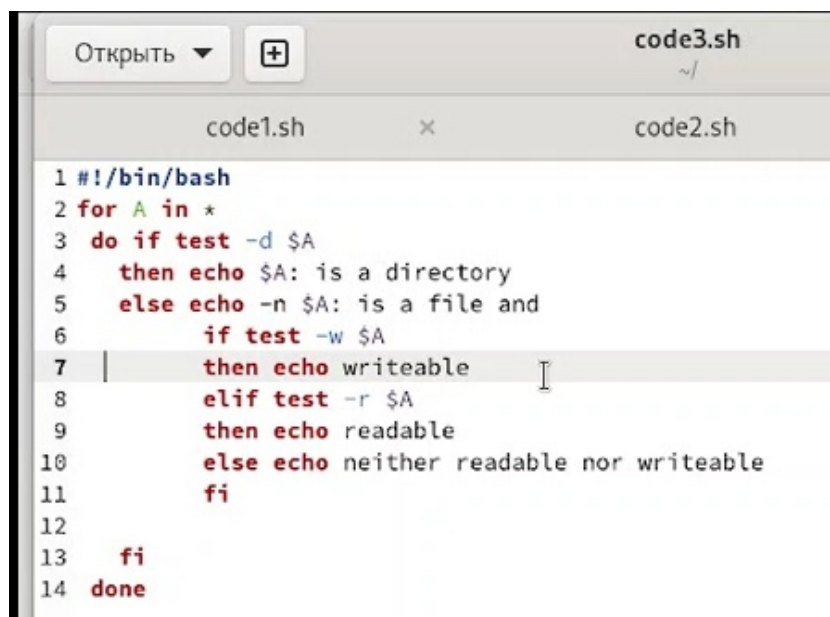
пишем команду `bash` для вывода действия кода на экран



```
[marieizvekova@fedora ~]$ bash code2.sh 1 2 3 4
1
2
3
4
```

Рис. 3.7: вывод на экран

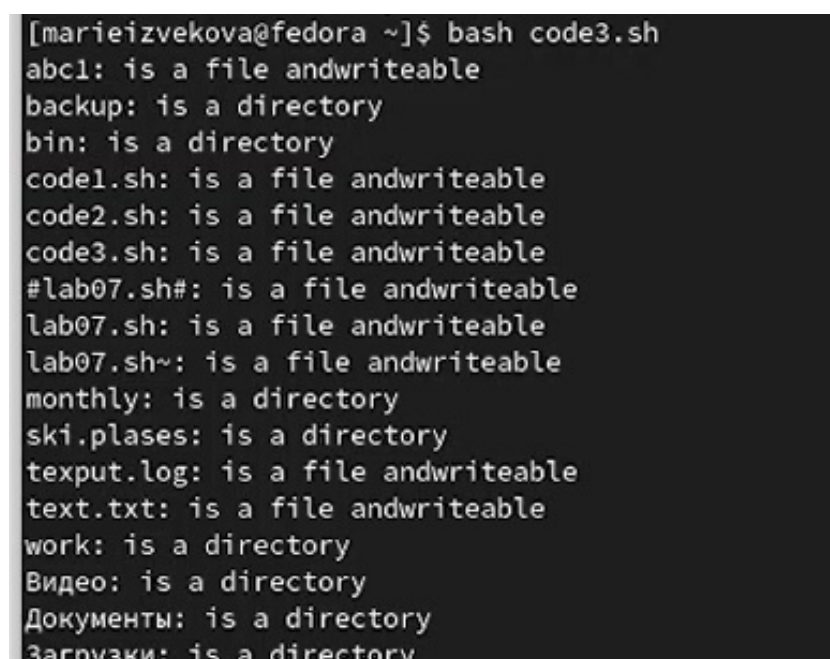
3. для считывания файла и дериктории мы пишем следующую программу, которая будет выводить на экран информацию о каталоге и о ее доступе



```
1 #!/bin/bash
2 for A in *
3 do if test -d $A
4 then echo $A: is a directory
5 else echo -n $A: is a file and
6     if test -w $A
7 then echo writeable
8 elif test -r $A
9 then echo readable
10 else echo neither readable nor writeable
11 fi
12
13 fi
14 done
```

Рис. 3.8: code 3

с помощью команды bash выводим действие команды на экран

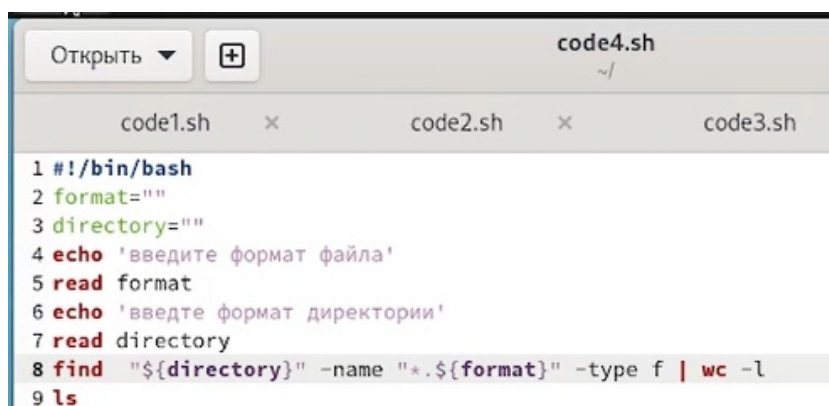


```
[marieizvekova@fedora ~]$ bash code3.sh
abc1: is a file andwriteable
backup: is a directory
bin: is a directory
code1.sh: is a file andwriteable
code2.sh: is a file andwriteable
code3.sh: is a file andwriteable
#lab07.sh#: is a file andwriteable
lab07.sh: is a file andwriteable
lab07.sh~: is a file andwriteable
monthly: is a directory
ski.plases: is a directory
texput.log: is a file andwriteable
text.txt: is a file andwriteable
work: is a directory
Видео: is a directory
Документы: is a directory
Загрузки: is a directory
```

Рис. 3.9: вывод третьего кода

4. для определения сколько файлов определенного формата находятся в на-

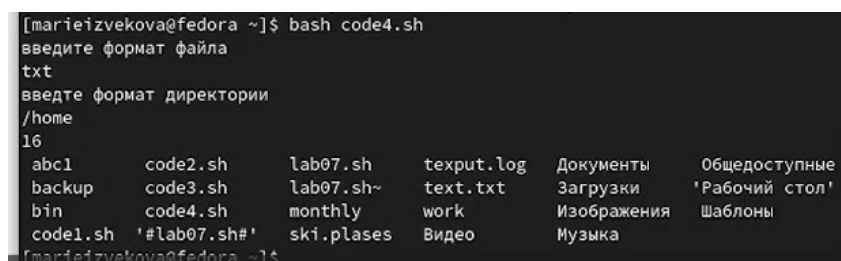
шем каталоге мы пишем в следующую команду



```
1 #!/bin/bash
2 format=""
3 directory=""
4 echo 'введите формат файла'
5 read format
6 echo 'введите формат директории'
7 read directory
8 find "${directory}" -name "*.${format}" -type f | wc -l
9 ls
```

Рис. 3.10: code 4

с помощью команды `bash` выводим действие команды на экран



```
[marieizvekova@fedora ~]$ bash code4.sh
введите формат файла
txt
введите формат директории
/home
16
abc1      code2.sh   lab07.sh   texput.log  Документы  Общедоступные
backup    code3.sh   lab07.sh~  text.txt    Загрузки   'Рабочий стол'
bin        code4.sh   monthly    work        Изображения Шаблоны
code1.sh  '#lab07.sh#' ski.places  Видео      Музыка
[marieizvekova@fedora ~]$
```

Рис. 3.11: вывод четвертого кода

4 Выводы

Изучили основы программирования в оболочке ОС UNIX/Linux. Научились писать небольшие командные файлы.

Список литературы