

# **Отчет по лабораторной работе №11**

**Программирование в командном процессоре ОС UNIX. Командные  
файлы**

Чигладзе Майя Владиславовна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Порядок выполнения лабораторной работы</b>	<b>6</b>
2.1	Задание 1. . . . .	6
2.2	Задание 2. . . . .	7
2.3	Задание 3. . . . .	9
2.4	Задание 4. . . . .	11
<b>3</b>	<b>Ответы на контрольные вопросы</b>	<b>12</b>
<b>4</b>	<b>Выводы</b>	<b>14</b>
	<b>Список литературы</b>	<b>15</b>

## Список иллюстраций

2.1	Код . . . . .	6
2.2	Результат 1 . . . . .	6
2.3	Результат 2 . . . . .	7
2.4	Код . . . . .	8
2.5	Результат . . . . .	9
2.6	Код . . . . .	10
2.7	Результат . . . . .	11
2.8	Код . . . . .	11
2.9	Результат . . . . .	11

## Список таблиц

# 1 Цель работы

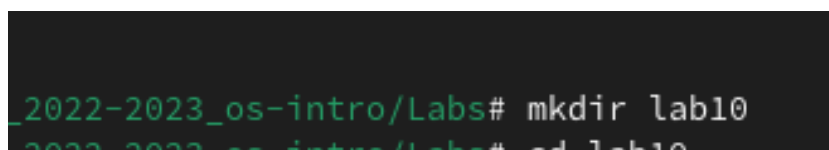
Изучить основы программирования в оболочке ОС UNIX/Linux. Научиться писать небольшие командные файлы

## 2 Порядок выполнения лабораторной работы

### 2.1 Задание 1.

Задача: Написать скрипт, который при запуске будет делать резервную копию самого себя (то есть файла, в котором содержится его исходный код) в другую директорию бэкап в вашем домашнем каталоге. При этом файл должен архивироваться одним из архиваторов на выбор zip, bzip2 или tar. Способ использования команд архивации необходимо узнать, изучив справку.

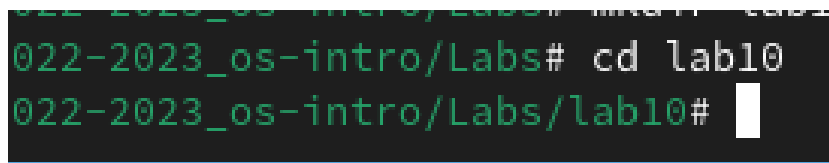
1. Код/файл (рис. 1).



```
_2022-2023_os-intro/Labs# mkdir lab10  
_2022-2023_os-intro/Labs# cd lab10
```

Рис. 2.1: Код

2. Результат (рис. 2).



```
_2022-2023_os-intro/Labs# cd lab10  
_2022-2023_os-intro/Labs/lab10#
```

Рис. 2.2: Результат 1

### 3. Результат (рис. 3).

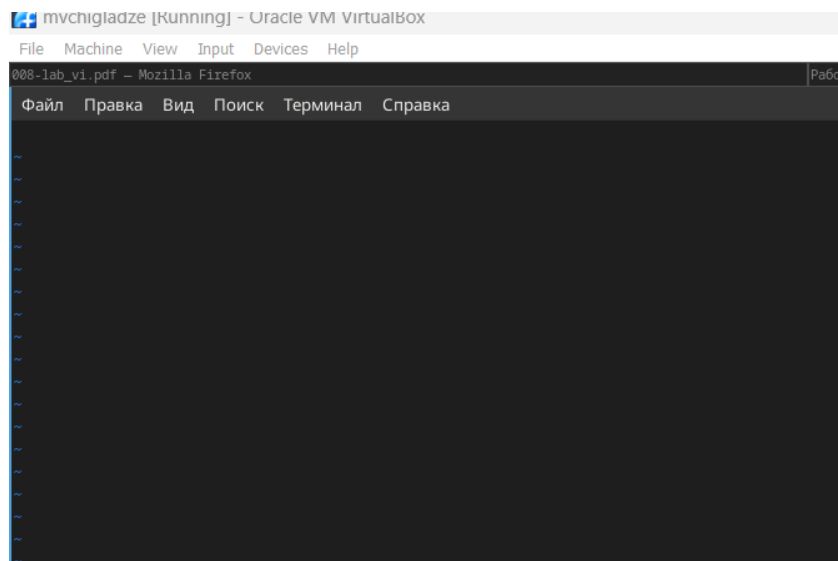
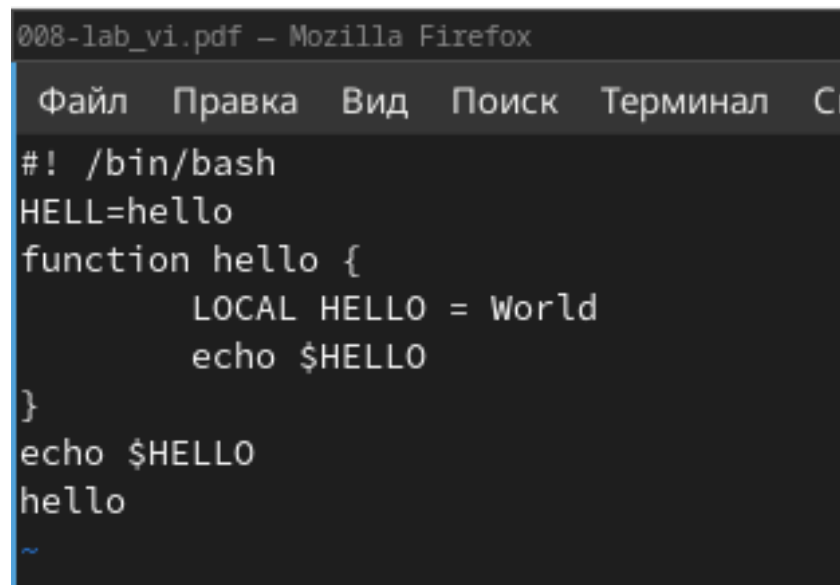


Рис. 2.3: Результат 2

## 2.2 Задание 2.

Задача: Написать пример командного файла, обрабатывающего любое произвольное число аргументов командной строки, в том числе превышающее десять. Например, скрипт может последовательно распечатывать значения всех переданных аргументов

### 1. Код/файл (рис. 4).



```
008-lab_vi.pdf - Mozilla Firefox
Файл  Правка  Вид  Поиск  Терминал  Сл
#!/bin/bash
HELL=hello
function hello {
    LOCAL HELLO = World
    echo $HELLO
}
echo $HELLO
hello
~
```

Рис. 2.4: Код

2. Результат (рис. 5).





о нужном каталоге и выводил информацию о возможностях доступа к файлам этого каталога.

1. Код/файл (рис. 6).

```
#!/bin/bash
HELL=hello
function hello {
    LOCAL HELLO = World
    echo $HELLO
}
echo $HELLO
hello
```

Рис. 2.6: Код

2. Результат (рис. 7).

```
os-intro/Labs# cd lab10  
os-intro/Labs/lab10# vi hallo.sh  
os-intro/Labs/lab10#
```

Рис. 2.7: Результат

## 2.4 Задание 4.

Задача: Написать командный файл, который получает в качестве аргумента командной строки формат файла (.txt, .doc, .jpg, .pdf и т.д.) и вычисляет количество таких файлов в указанной директории. Путь к директории также передаётся в виде аргумента командной строки

1. Код/файл (рис. 8).

```
каталога  
os-intro/Labs/lab10# chmod +x hallo.sh  
os-intro/Labs/lab10#
```

Рис. 2.8: Код

2. Результат (рис. 9).

```
каталога  
os-intro/Labs/lab10# chmod +x hallo.sh  
os-intro/Labs/lab10# vi hallo.sh
```

Рис. 2.9: Результат

### 3 Ответы на контрольные вопросы

1. Командная оболочка - это программа, которая обеспечивает интерфейс между пользователем и операционной системой. Она принимает команды от пользователя и передает их на выполнение операционной системе. Примеры командных оболочек включают в себя Bash, Zsh, Fish и PowerShell. Они отличаются синтаксисом команд, возможностями настройки и поддержкой дополнительных функций.
2. POSIX (Portable Operating System Interface) - это стандарт, определяющий интерфейс между операционной системой и приложениями. Он описывает, как приложения должны взаимодействовать с операционной системой для обеспечения переносимости кода между различными платформами.
3. Переменные и массивы определяются с помощью символов доллара (\$), знака равенства (=) и имени переменной. Массивы создаются с использованием квадратных скобок ([]).
4. Оператор let используется для выполнения арифметических операций и присвоения значений переменным. Оператор read считывает ввод пользователя.
5. Арифметические операции включают сложение (+), вычитание (-), умножение (\*) и деление (/).
6. Операция (( )) используется для вычисления арифметических выражений.
7. Стандартные имена переменных включают \$PWD (текущий рабочий каталог), \$RANDOM (случайное число), \$? (код возврата последней команды) и \$PATH (путь поиска для исполняемых файлов).

8. Метасимволы - это символы, имеющие специальное значение в контексте регулярных выражений или командных строк.
9. Для экранирования метасимволов необходимо использовать обратный слэш (`\`).
10. Командные файлы создаются с помощью текстового редактора и сохраняются с расширением `.sh`. Для запуска командного файла нужно выполнить его в командной строке.
11. Функции определяются с использованием ключевого слова `function`, за которым следует имя функции и список параметров.
12. Команда `"ls -la"` может использоваться для определения типа файла (обычный файл или каталог).
13. Команды `set`, `typeset` и `unset` используются для определения, изменения и удаления переменных соответственно.
14. Параметры передаются в командной строке через разделитель (обычно пробел).
15. Специальные переменные включают `$*`, `$?`, `$!`, `$@` и `$#`.

## **4 Выводы**

В ходе лабораторной работы, я изучила основы программирования в оболочке ОС UNIX/Linux. Научилась писать небольшие командные файлы

## **Список литературы**