

Отчёт по лабораторной работе №10

Дугина Виктория Игоревна

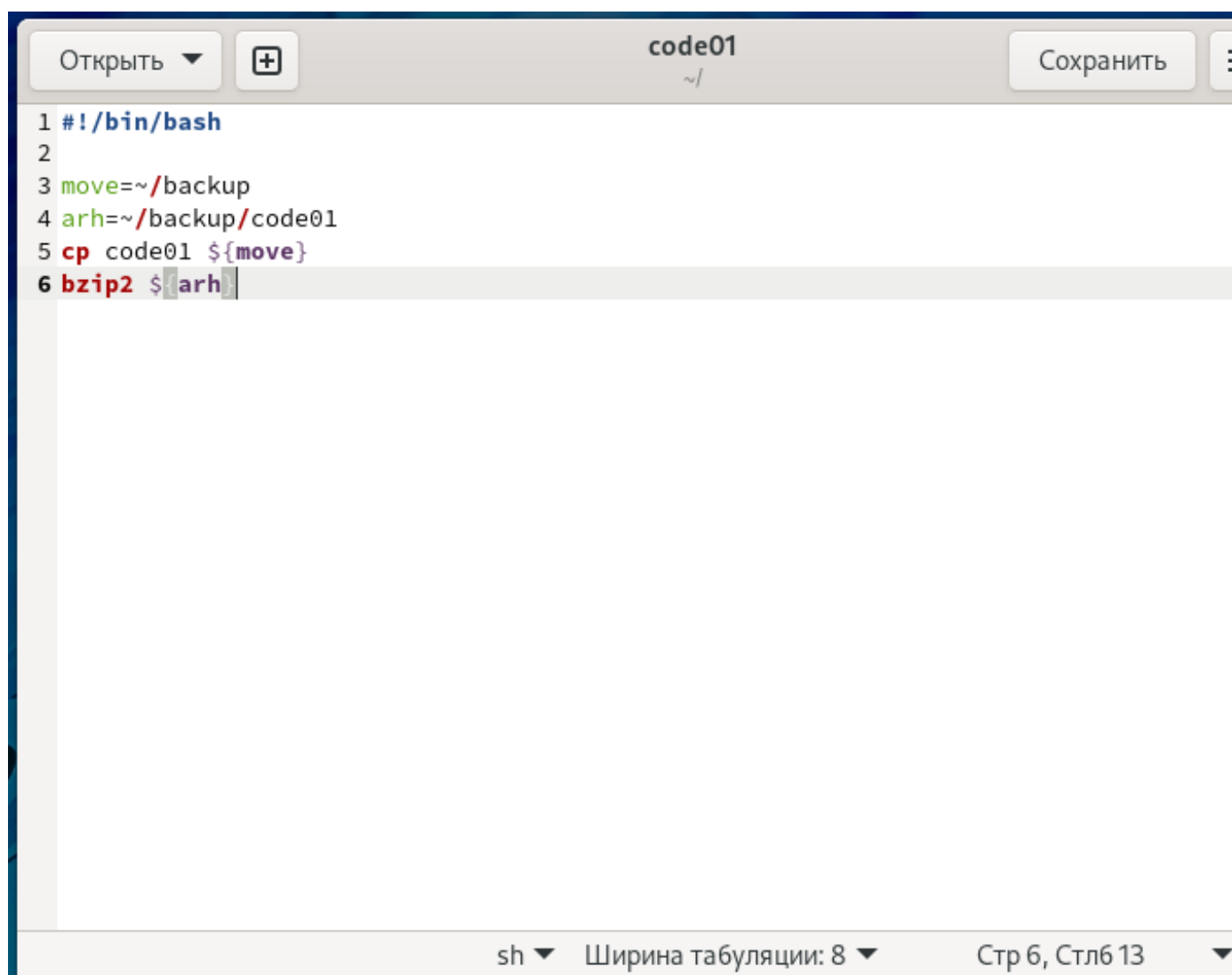
Цель работы: изучить основы программирования в оболочке ОС UNIX/Linux. Научиться писать небольшие командные файлы.

Выполнение работы

1. Написать скрипт, который при запуске будет делать резервную копию самого себя (то есть файла, в котором содержится его исходный код) в другую директорию backup 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.

move - адрес, куда копировать файл. arh - ссылка на копию, которая появится после выполнения следующей строки (копирования).

Файл копируется в нужный каталог и после архивируется там (команда bzip2).

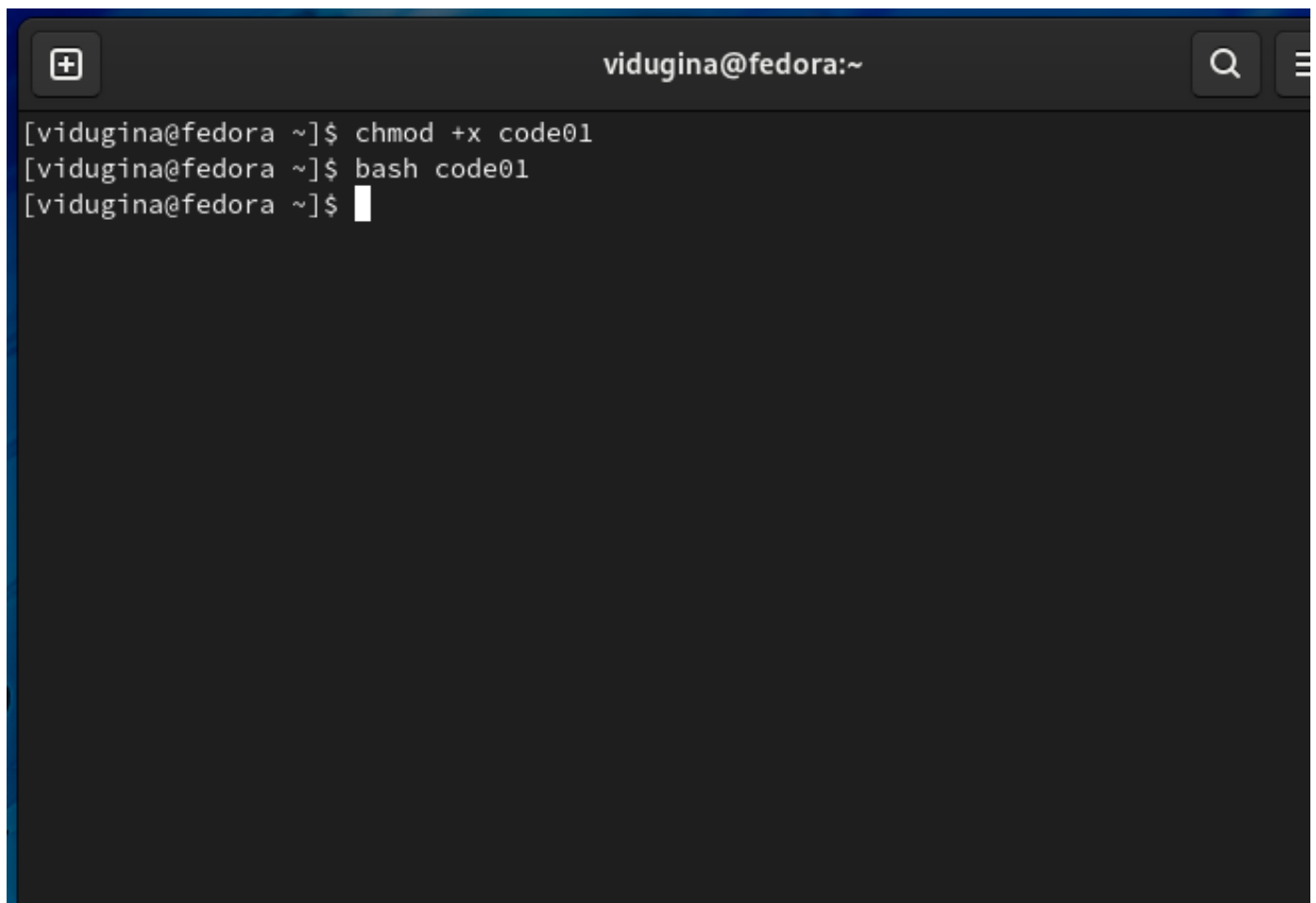


```
1 #!/bin/bash
2
3 move=~/.backup
4 arh=~/.backup/code01
5 cp code01 ${move}
6 bzip2 ${arh}
```

The screenshot shows a code editor window titled 'code01'. The editor contains a shell script with six lines. Line 1 is the shebang '#!/bin/bash'. Line 2 is empty. Line 3 sets 'move' to '~/.backup'. Line 4 sets 'arh' to '~/.backup/code01'. Line 5 uses 'cp' to copy 'code01' to the path stored in 'move'. Line 6 uses 'bzip2' to compress the file at the path stored in 'arh'. The editor has a toolbar at the top with 'Открыть', a plus icon, the title 'code01', a tilde icon, and 'Сохранить'. At the bottom, it shows 'sh', 'Ширина табуляции: 8', and 'Стр 6, Стлб 13'.

Рис 1.1

Запускаем программу.

A terminal window with a dark background and light gray text. The title bar at the top shows a plus icon on the left, the text 'vidugina@fedora:~' in the center, and a magnifying glass icon and a hamburger menu icon on the right. The terminal content shows three lines of text: the first line is '[vidugina@fedora ~]\$ chmod +x code01', the second line is '[vidugina@fedora ~]\$ bash code01', and the third line is '[vidugina@fedora ~]\$' followed by a white cursor block.

```
[vidugina@fedora ~]$ chmod +x code01
[vidugina@fedora ~]$ bash code01
[vidugina@fedora ~]$
```

Рис 1.2

Результат.

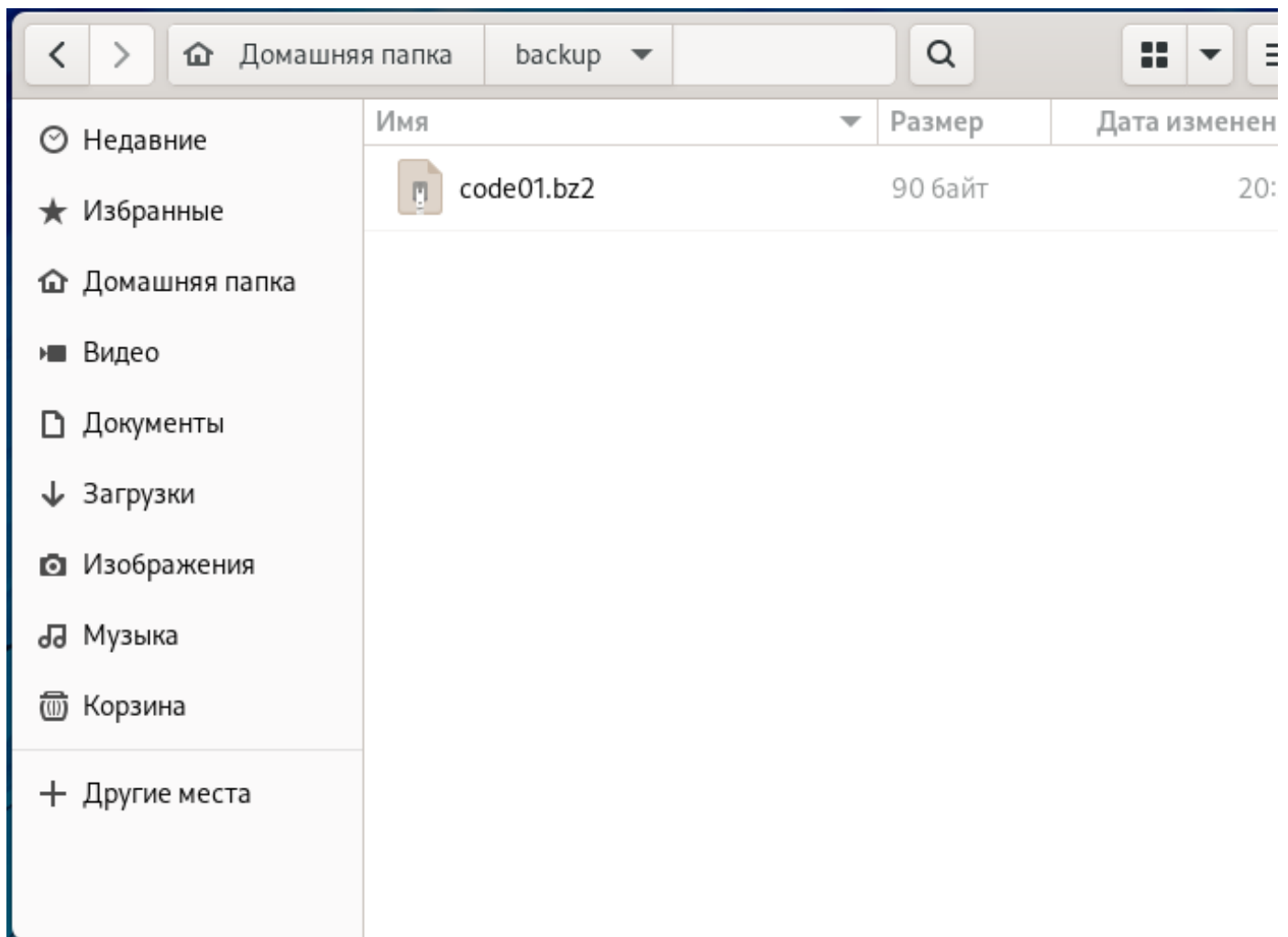


Рис 1.3

2. 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.

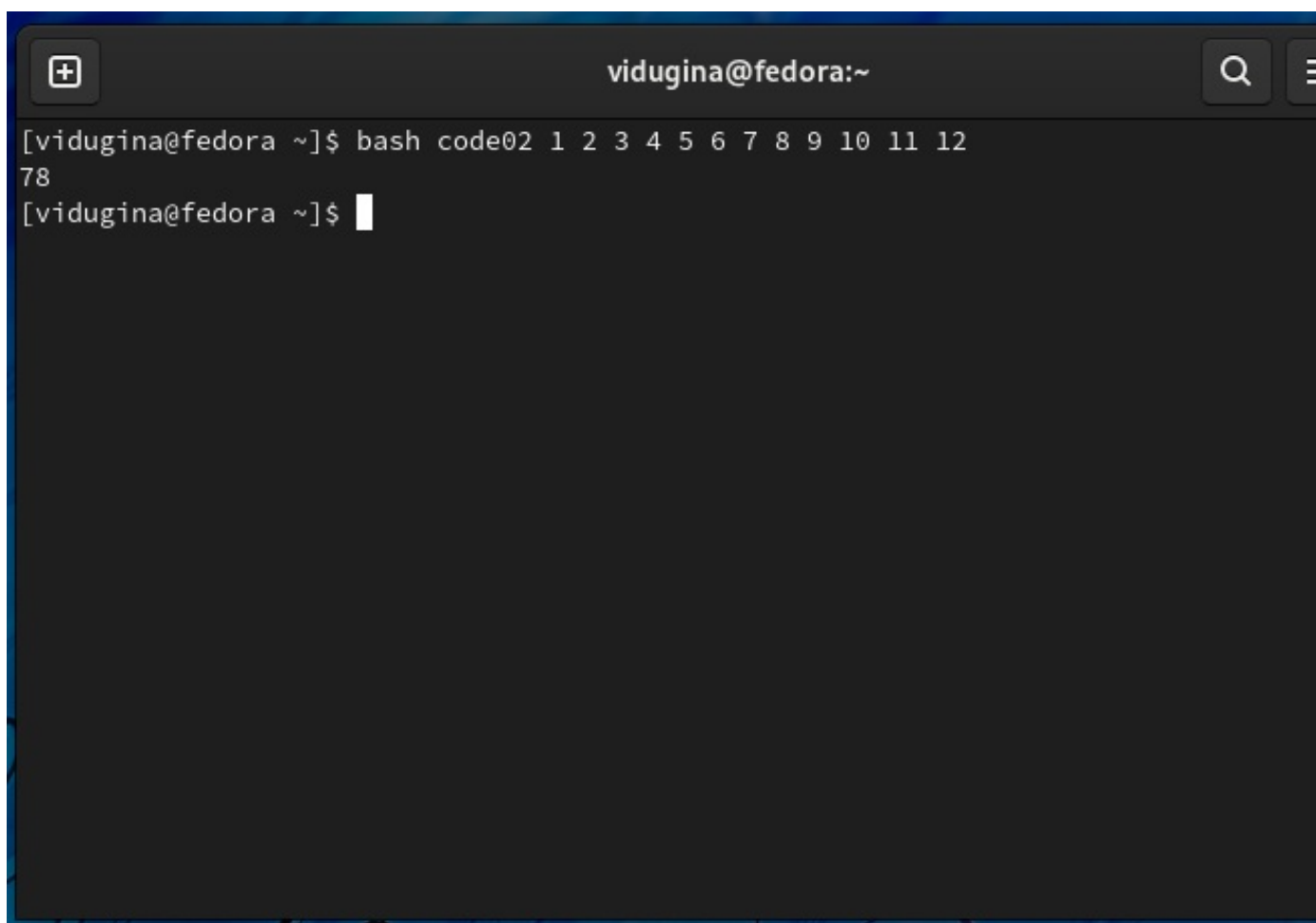
Программа принимает любое количество аргументов (чисел) и выводит их сумму.

Задаём изначальную сумму равную нулю (`let sum=0`). Запускаем цикл `for`, где `x` пробегает по всем значениям аргументов (`$@`) и прибавляет их к общей сумме. В конце выводим результат (`echo $sum`).

```
1 #!/bin/bash
2
3 let sum=0
4 for x in $@
5 do
6 sum=$(( ${x} + sum ))
7 done
8 echo $sum
```

Рис 2.1

Работа программы.

A terminal window with a dark background. The title bar shows 'vidugina@fedora:~'. The prompt is '[vidugina@fedora ~]\$'. The command 'bash code02 1 2 3 4 5 6 7 8 9 10 11 12 78' has been entered and executed. The prompt is now '[vidugina@fedora ~]\$' with a cursor. The window has standard Linux window controls on the left and search/menu icons on the right.

```
vidugina@fedora:~  
[vidugina@fedora ~]$ bash code02 1 2 3 4 5 6 7 8 9 10 11 12 78  
[vidugina@fedora ~]$
```

Рис 2.2

3. Написать командный файл — аналог команды `ls` (без использования самой этой команды и команды `dir`). Требуется, чтобы он выдавал информацию о нужном каталоге и выводил информацию о возможностях доступа к файлам этого каталога.

4. 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.

Вывод: я изучила основы программирования в оболочке ОС UNIX/Linux. Научилась писать небольшие командные файлы.

Контрольные вопросы

1. Объясните понятие командной оболочки. Приведите примеры командных оболочек. Чем они отличаются?
2. Что такое POSIX?
3. Как определяются переменные и массивы в языке программирования `bash`?
4. Каково назначение операторов `let` и `read`?
5. Какие арифметические операции можно применять в языке программирования

bash?

6. Что означает операция (())?

7. Какие стандартные имена переменных Вам известны?

8. Что такое метасимволы?

9. Как экранировать метасимволы?

10. Как создавать и запускать командные файлы?

11. Как определяются функции в языке программирования bash?

12. Каким образом можно выяснить, является файл каталогом или обычным файлом?

13. Каково назначение команд set, typeset и unset?

14. Как передаются параметры в командные файлы?

15. Назовите специальные переменные языка bash и их назначение.