

# Лабораторная работа №10

---

Сиссе Мохамед Ламин; НММбд-01-22

<sup>1</sup>RUDN University, Moscow, Russian Federation

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

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.
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.

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.

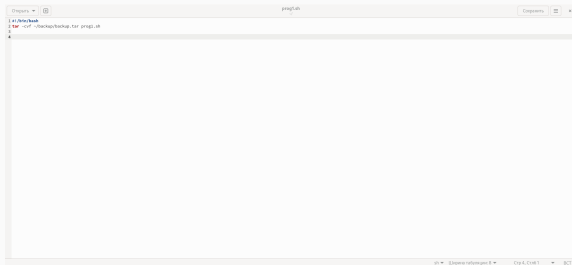
Командный процессор (командная оболочка, интерпретатор команд shell) — это программа, позволяющая пользователю взаимодействовать с операционной системой компьютера. В операционных системах типа UNIX/Linux наиболее часто используются следующие реализации командных оболочек:

- 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 оболочкой 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`. В других оболочках большинство команд будет совпадать с описанными ниже.

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. рис. ((fig:001?; fig:002?; fig:003?))





A screenshot of a terminal window titled "progLab". The window shows a shell prompt "i@i:~/progLab" followed by the command "tar -xvf ~/backup/backup.tar progLab". The command is partially executed, with a cursor at the end of the line. The terminal window has a standard Linux interface with a title bar and window controls.

```
i@i:~/progLab  
tar -xvf ~/backup/backup.tar progLab
```

Рис. 1: Текст первой программы

# Выполнение лабораторной работы

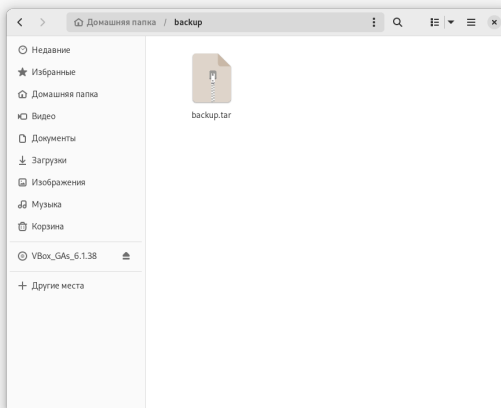
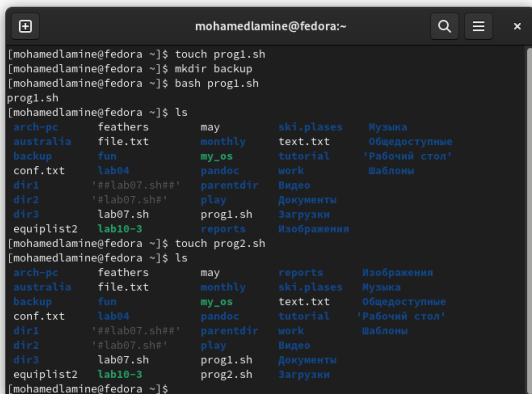


Рис. 2: Проверка работы программы

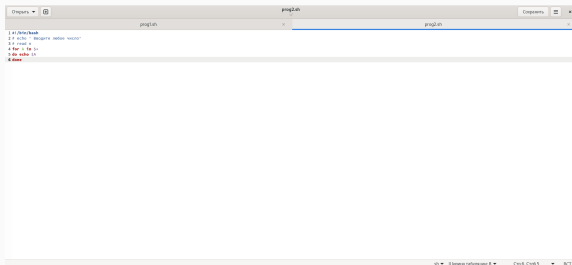
# Выполнение лабораторной работы



```
mohamedlamine@fedora:~  
[mohamedlamine@fedora ~]$ touch prog1.sh  
[mohamedlamine@fedora ~]$ mkdir backup  
[mohamedlamine@fedora ~]$ bash prog1.sh  
prog1.sh  
[mohamedlamine@fedora ~]$ ls  
arch-pc      feathers      may           ski.plases    Музыка  
australia    file.txt      monthly       text.txt      Общедоступные  
backup       fun           my_os         tutorial       'Рабочий стол'  
conf.txt     lab04         pandoc        work          Шаблоны  
dir1         '##lab07.sh##' parentdir     Видео  
dir2         '##lab07.sh##' play         Документы  
dir3         lab07.sh     prog1.sh     Загрузки  
equiplist2   lab10-3      reports      Изображения  
[mohamedlamine@fedora ~]$ touch prog2.sh  
[mohamedlamine@fedora ~]$ ls  
arch-pc      feathers      may           reports        Изображения  
australia    file.txt      monthly       ski.plases     Музыка  
backup       fun           my_os         text.txt      Общедоступные  
conf.txt     lab04         pandoc        tutorial       'Рабочий стол'  
dir1         '##lab07.sh##' parentdir     work          Шаблоны  
dir2         '##lab07.sh##' play         Видео  
dir3         lab07.sh     prog1.sh     Документы  
equiplist2   lab10-3      prog2.sh     Загрузки  
[mohamedlamine@fedora ~]$
```

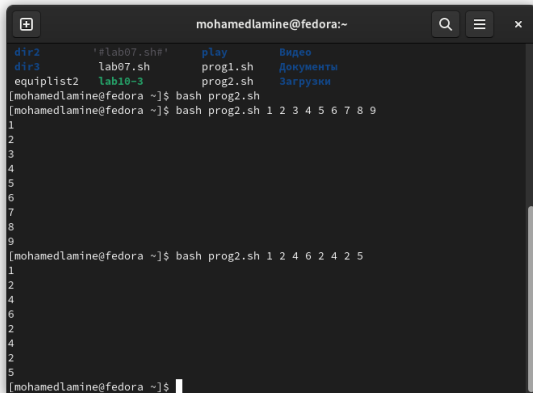
Рис. 3: Создание файла для второй программы, проверка содержимого домашнего каталога

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. ((fig:004?; fig:005?))



```
1 #!/bin/bash
2 set -x
3 set -e
4 vecsum() {
5     local v1=$1
6     local v2=$2
7     local sum=0
8     for i in $(seq 0 $((${#v1} - 1))); do
9         sum=$((sum + ${v1:$i:1} + ${v2:$i:1}))
10     done
11     echo $sum
12 }
```

Рис. 4: Текст второй программы

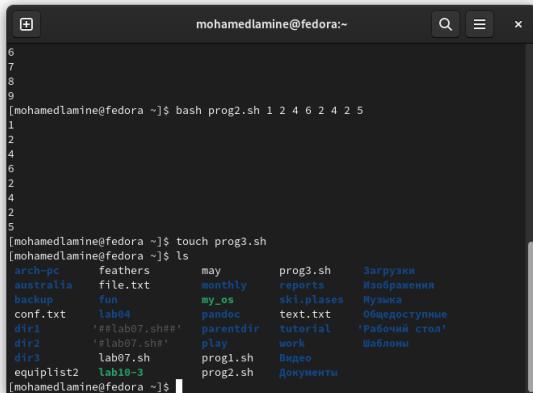


```
mohamedlamine@fedora:~  
dir2      '#lab07.sh#'    play      Видео  
dir3      lab07.sh       prog1.sh  Документы  
equiplist2 lab10-3          prog2.sh  Загрузки  
[mohamedlamine@fedora ~]$ bash prog2.sh  
[mohamedlamine@fedora ~]$ bash prog2.sh 1 2 3 4 5 6 7 8 9  
1  
2  
3  
4  
5  
6  
7  
8  
9  
[mohamedlamine@fedora ~]$ bash prog2.sh 1 2 4 6 2 4 2 5  
1  
2  
4  
6  
2  
4  
2  
5  
[mohamedlamine@fedora ~]$
```

Рис. 5: Проверка работы второй программы

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.  
((fig:006?; fig:007?; fig:008?))

# Выполнение лабораторной работы

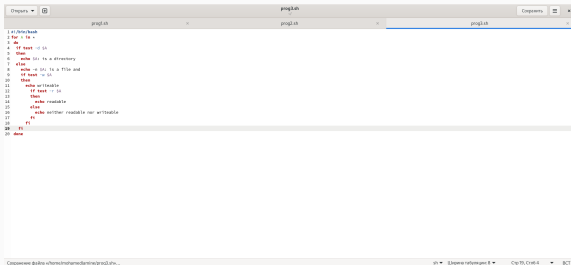


```
mohamedlamine@fedora:~  
6  
7  
8  
9  
[mohamedlamine@fedora ~]$ bash prog2.sh 1 2 4 6 2 4 2 5  
1  
2  
4  
6  
2  
4  
2  
5  
[mohamedlamine@fedora ~]$ touch prog3.sh  
[mohamedlamine@fedora ~]$ ls  
arch-pc      feathers      may           prog3.sh      Загрузки  
australia    file.txt     monthly      reports        Изображения  
backup       fun          my_os        ski.places     Музыка  
conf.txt     lab04        pandoc       text.txt       Общедоступные  
dir1         '##lab07.sh##' parentdir    tutorial       'Рабочий стол'  
dir2         '##lab07.sh##' play         work           Шаблоны  
dir3         lab07.sh    prog1.sh     Видео  
equiplist2   lab10-3     prog2.sh     Документы  
[mohamedlamine@fedora ~]$
```

Рис. 6: Создание файла для третьей программы



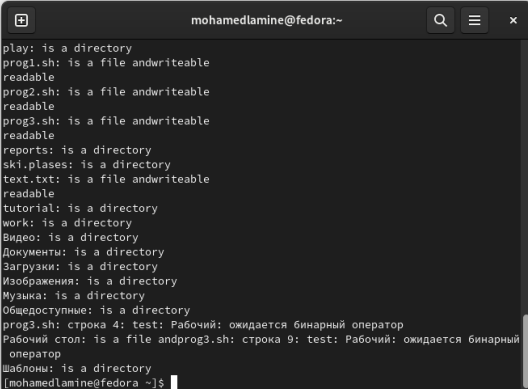
# Выполнение лабораторной работы



```
1 #!/bin/bash
2 for i in *
3 do
4     if test -d $i
5     then
6         echo $i is a directory
7     else
8         echo -e $i is a file and
9         if test -w $i
10        then
11            echo writable
12            if test -x $i
13            then
14                echo readable
15            else
16                echo neither readable nor writable
17            fi
18        fi
19    fi
20 done
```

Рис. 7: Текст третьей программы

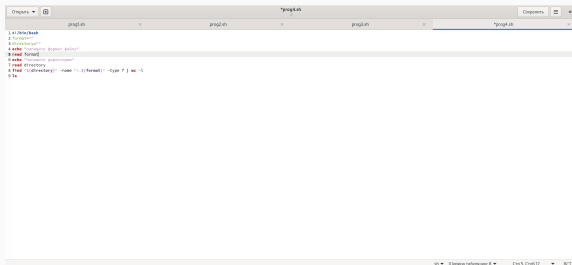
# Выполнение лабораторной работы

A terminal window titled 'mohamedlamine@fedora:~' with search, menu, and close icons. It displays the output of a command, likely 'ls -la', showing permissions for various files and directories. The output includes: 'play: is a directory', 'prog1.sh: is a file and writeable readable', 'prog2.sh: is a file and writeable readable', 'prog3.sh: is a file and writeable readable', 'reports: is a directory', 'ski.places: is a directory', 'text.txt: is a file and writeable readable', 'tutorial: is a directory', 'work: is a directory', 'Видео: is a directory', 'Документы: is a directory', 'Загрузки: is a directory', 'Изображения: is a directory', 'Музыка: is a directory', 'Общедоступные: is a directory', 'prog3.sh: строка 4: test: Рабочий: ожидается бинарный оператор', 'Рабочий стол: is a file and prog3.sh: строка 9: test: Рабочий: ожидается бинарный оператор', and 'Шаблоны: is a directory'. The prompt '[mohamedlamine@fedora ~]\$' is at the bottom.

```
mohamedlamine@fedora:~
play: is a directory
prog1.sh: is a file and writeable
readable
prog2.sh: is a file and writeable
readable
prog3.sh: is a file and writeable
readable
reports: is a directory
ski.places: is a directory
text.txt: is a file and writeable
readable
tutorial: is a directory
work: is a directory
Видео: is a directory
Документы: is a directory
Загрузки: is a directory
Изображения: is a directory
Музыка: is a directory
Общедоступные: is a directory
prog3.sh: строка 4: test: Рабочий: ожидается бинарный оператор
Рабочий стол: is a file and prog3.sh: строка 9: test: Рабочий: ожидается бинарный
оператор
Шаблоны: is a directory
[mohamedlamine@fedora ~]$
```

Рис. 8: Проверка работы третьей программы

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. ((fig:009?; fig:010?))



The screenshot shows a code editor window with a single file named "prog4.cpp". The code is as follows:

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     int n;
6     cout << "Введите количество элементов: ";
7     cin >> n;
8     int *a = new int[n];
9     for (int i = 0; i < n; i++)
10     {
11         a[i] = rand() % 100;
12     }
13     for (int i = 0; i < n; i++)
14     {
15         cout << a[i] << " ";
16     }
17     cout << endl;
18     return 0;
19 }
```

Рис. 9: Текст четвертой программы

# Выполнение лабораторной работы

```
mohamedlamine@fedora:~  
Музыка: is a directory  
Общедоступные: is a directory  
prog3.sh: строка 4: test: Рабочий: ожидается бинарный оператор  
Рабочий стол: is a file and prog3.sh: строка 9: test: Рабочий: ожидается бинарный оператор  
Шаблоны: is a directory  
[mohamedlamine@fedora ~]$ touch prog4.sh  
[mohamedlamine@fedora ~]$ bash prog4.sh  
Напишите формат файла  
txt  
Напишите директорию  
/home/mohamedlamine  
find: '/home/mohamedlamine/Загрузки/pandoc-2.19.2-linux-arm64/pandoc-2.19.2': От  
казано в доступе  
18  
arch-pc      feathers      may           prog3.sh      Документы  
australia    file.txt      monthly       prog4.sh      Загрузки  
backup       fun           my_os         reports       Изображения  
conf.txt     lab04         pandoc        ski.places    Музыка  
dir1         '#lab07.sh##' parentdir     text.txt      Общедоступные  
dir2         '#lab07.sh#' play          tutorial      'Рабочий стол'  
dir3         lab07.sh     prog1.sh      work          Шаблоны  
equiplist2   lab10-3      prog2.sh      Видео  
[mohamedlamine@fedora ~]$
```

Рис. 10: Проверка работы четвертой программы

В процессе выполнения лабораторной работы я изучил основы программирования в оболочке ОС UNIX/Linux. Научилась писать небольшие командные файлы.

1. Лабораторная работа № 10. Программирование в командном процессоре ОС UNIX. Командные файлы [Электронный ресурс]. URL: <https://esystem.rudn.ru/>.

Спасибо за внимание!