# Российский университет дружбы народов

### Факультет физико-математических и естественных наук

## Отчёт по лабораторной работе №11

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

**Дисциплина:** Операционные системы

**Студент:** Оразгелдиева Огулнур

**Группа:** НПИбд-02-20

**Студ. номер:** 1032205431

2021, Москва

**Лабораторная работа №11**

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

**Цель:**

* изучить основы программирования в оболочке ОС UNIX/Linux.
* научиться писать небольшие командные файлы.

**Задачи:**

1. Ознакомиться с теоретическим материалом
2. Написать скрипт, который при запуске будет делать резервную копию самого себя в другую директорию backup в вашем домашнем каталоге. При этом файл должен архивироваться одним из архиваторов на выбор zip, bzip2 или tar.
3. Написать пример командного файла, обрабатывающего любое произвольное число аргументов командной строки, в том числе превышающее десять.
4. 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.
5. 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.

**Теоретические сведения** [[1]](https://esystem.rudn.ru/pluginfile.php/1142377/mod_resource/content/2/008-lab_shell_prog_1.pdf) [[2]](https://eternalhost.net/blog/sistemnoe-administrirovanie/linux-bash-chto-eto)

Командный процессор (командная оболочка, интерпретатор команд shell) — это программа, позволяющая пользователю взаимодействовать с операционной системой компьютера.

Bash (Bourne again shell или «возрождённый» shell) – это модифицированная версия программной оболочки Bourne-shell (sh или «Оболочка Борна»). Она является командным процессором, работающим интерактивно в текстовом окне. Bash нужен для приема команд пользователя и их отправки операционной системе для последующей обработки.

Взаимодействие оболочки и операционной системы обеспечивается с помощью специальной программы – терминала.

Командный процессор bash обеспечивает возможность использования переменных типа строка символов. Имена переменных могут быть выбраны пользователем. Пользователь имеет возможность присвоить переменной значение некоторой строки символов.

Использование значения, присвоенного некоторой переменной, называется подстановкой. Для того чтобы имя переменной не сливалось с символами, которые могут следовать за ним в командной строке, при подстановке в общем случае используется следующая форма записи: ${имя переменной}

Оболочка bash позволяет работать с массивами. Для создания массива используется команда set с флагом -A.

Оболочка bash поддерживает встроенные арифметические функции. Команда let является показателем того, что последующие аргументы представляют собой выражение, подлежащее вычислению.

Последовательность команд может быть помещена в текстовый файл. Такой файл называется командным. Далее этот файл можно выполнить по команде: *bash командный\_файл [аргументы]*

**Ход работы**

1. Создаём каталог *backup* с помощью команды *mkdir*. (см. рис. 1) Создаём файл lab11-1.sh с помощью текстового редактора. (см. рис. 1)

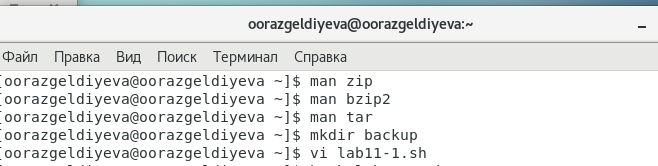
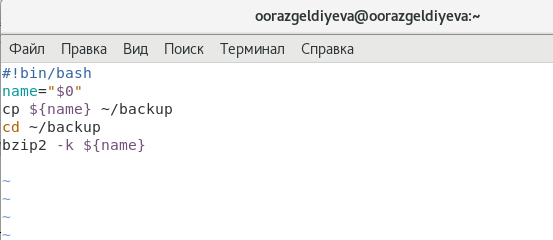


Рисунок 1. Создание каталога backup и файла lab11-1.sh

После открытия файла, пишем программу, по которому копируется командный файл (lab11-2.sh) в каталог backup и архивируется с помощью архиватора bzip2. (см. рис. 2)



*Рисунок 2. Программа к заданию 1*

Выполняем программу с помощью *bash*. (см. рис. 3)

**Пояснения:** задаем переменную *name* с названием данного командного файла (с указателем $0); далее копируем командный файл в каталог *~/backup* с помощью команды *cp*. Переходим в этот каталог и архивируем файл в *bz2* и при этом сохраняем сам файл. В итоге внутри каталога получаем два файла: командный файл *lab11-1.sh* и архив с этим командным файлом *lab11-1.sh.bz2*. (см. рис 3-4)

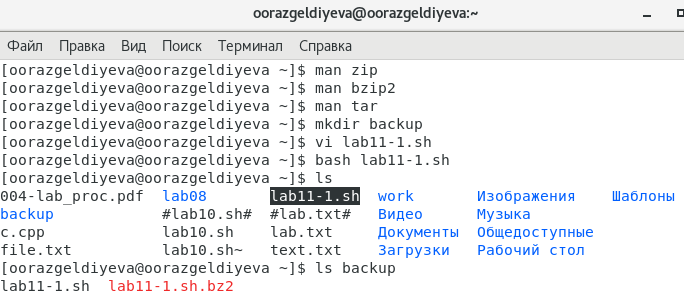


Рисунок 3. Содержимого домашнего каталога и каталога backup

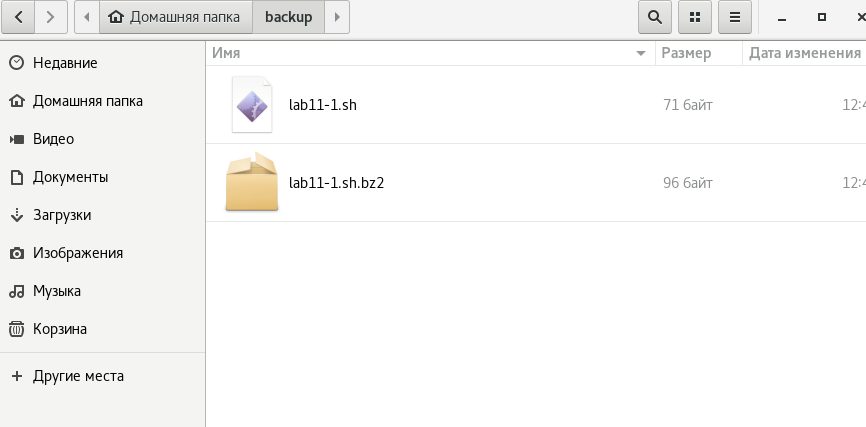


Рисунок 4. Содержимого каталога backup

Откроем полученный архив и посмотрим его содержимое. (см. рис. 5)

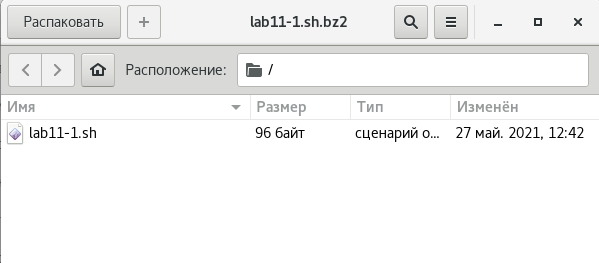


Рисунок 5. Архив

1. Создадим и откроем файл *lab11-2.sh* с программой, обрабатывающей любое произвольное число аргументов командной строки, в том числе превышающее десять. Например, скрипт может последовательно распечатывать значения всех переданных аргументов. (см. рис. 6)

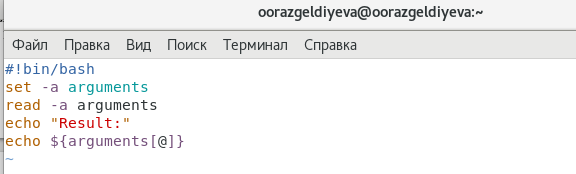


Рисунок 6. Программа к заданию 2

**Пояснения:** сначала объявляем массив для аргументов с помощью команды *set* и флага *-а*; читаем вводимые аргументы с помощью *read* и выводим их на экран с помощью *echo*.

Выполняем командный файл при помощи *bash* и вводим произвольные аргументы. (см. рис. 7)

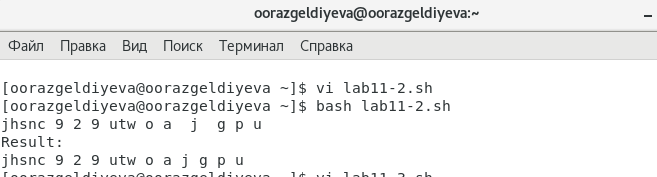


Рисунок 7. Выполнение программы

1. Создадим и откроем командный файл *lab11-3.sh* при помощи текстового редактора vi и напишем программу. (см. рис. 8)

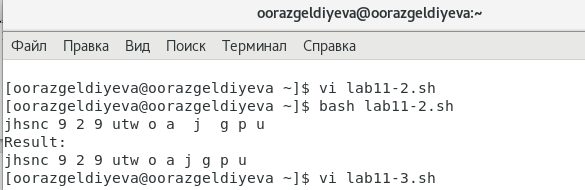


Рисунок 8. Создание файла при помощи vi

Программа - 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. (см. рис. 9)

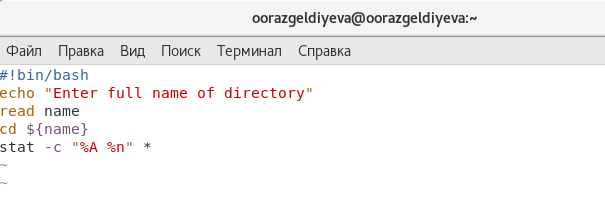


Рисунок 9. Программа к заданию 3

**Пояснения:** выводим запрос на ввод полного имени каталога; чтение названия каталога и переход в него; выводим на экран права доступа к файлам (при помощи %А) и название файлов (%n), использу команду *stat* с опцией *-с*. (в конце указываем "\*", что означает "текущий каталог") [[3]](https://losst.ru/komanda-stat-v-linux)

Выполняем программу с помощью *bash*. Вводим название директории, и в результате выводятся файлы с правами доступа к ним. (см. рис. 10) (в первом случае была ошибка, так как не указала \* для определения текущего каталога)

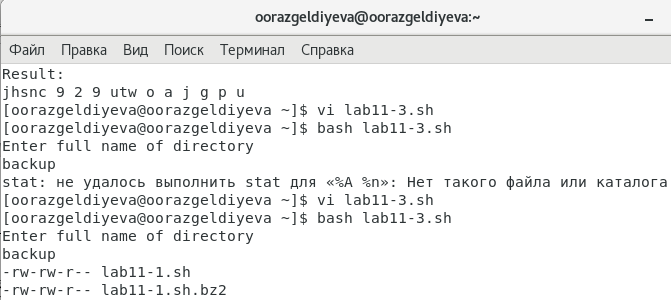


Рисунок 10. Выполнение программы

Как видно по рис. 10, для примера ввела название ранее созданного каталога *backup* (задание 1) и проверила файлы и права доступа к этим файлам, находящихся внутри этого каталога.

1. Создадим и откроем командный файл *lab11-4.sh* при помощи текстового редактора vi и напишем программу. (см. рис. 11)

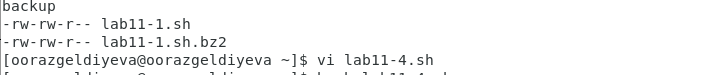


Рисунок 11. Создание файла

Напишем программу, 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. (см. рис. 12)

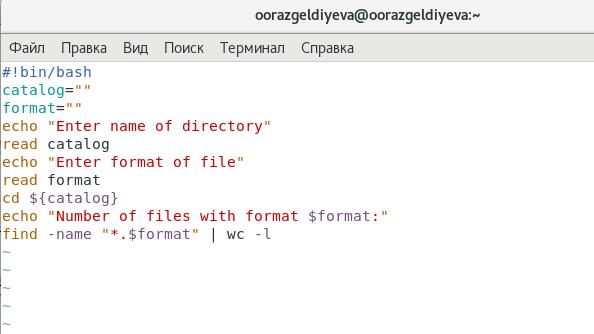


Рисунок 12. Программа к заданию 4

**Пояснения:** создаем переменные для названия каталога и формата файла (*catalog и format* соответственно); запрос на ввод названия каталога с помощью echo и его чтение; аналогично вводим формат. Переходим в заданны каталог и с помощью команды *find*, конвейера (|) и команды *wc -l* находим (find) и выводим количество файлов (wc -l) с указанным форматом.

Выполняем программу с помощью *bash*. Вводим название каталога (backup) и формат bz2. (см. рис. 13) (в первом случае сначала вводили формат, а потом выводился запрос на формат, поэтому исправила программу и выполнила еще раз)

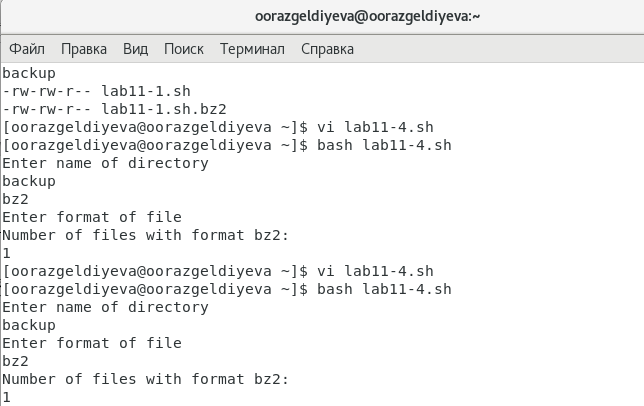


Рисунок 13. Выполнение программы

Также для примера посмотрим, сколько файлов с расширением *.txt* в каталоге *work*, с расширением *.sh* в *backup*. (см. рис. 14)

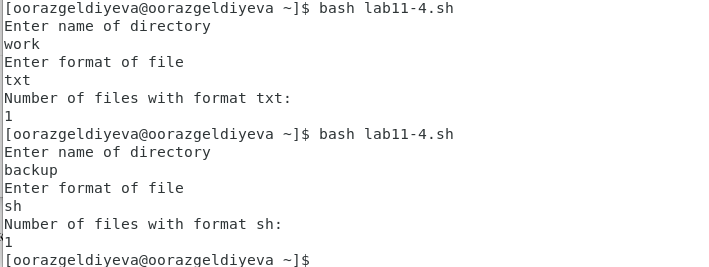


Рисунок 14. Выполнение программы

**Вывод:** на лабораторной работе изучила основы программирования в оболочке ОС UNIX/Linux; научилась писать небольшие командные файлы.

**Библиография** [[1] - РУДН, Операционные системы, Программирование в командном процессоре ОС UNIX. Командные файлы](https://esystem.rudn.ru/pluginfile.php/1142377/mod_resource/content/2/008-lab_shell_prog_1.pdf)

[[2] - Linux. Bash.Что это? Основы bash](https://eternalhost.net/blog/sistemnoe-administrirovanie/linux-bash-chto-eto)

[[3] - Команда stat в Linux](https://losst.ru/komanda-stat-v-linux)