Отчёт по лабораторной работе №11

Серегин Денис Алексеевич

Содержание

# Цель работы

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

# Задания

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.

# Выполнение лабораторной работы

1. Написал скрипт, который при запуске делает резервную копию самого себя в другую директорию backup в своём домашнем каталоге. Файл архивировался предустановленным в системе архиватором gzip. Способ использования команд архивации узнал, изучив справку.

(рис 1. 1)

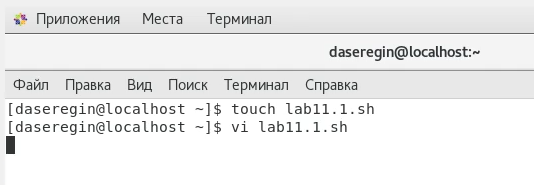


Figure 1: Создание файл

(рис 2. 2)

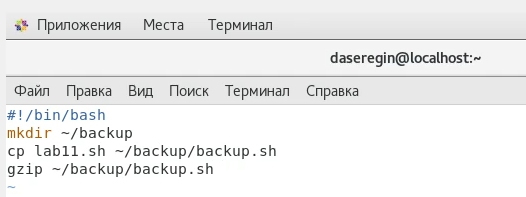


Figure 2: Код скрипта

(рис 3. 3)

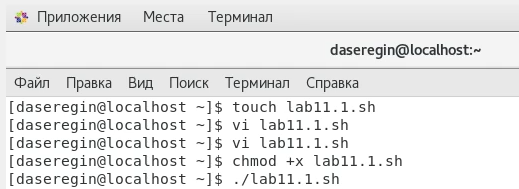


Figure 3: Создание исполняемого файла

(рис 4. 4)

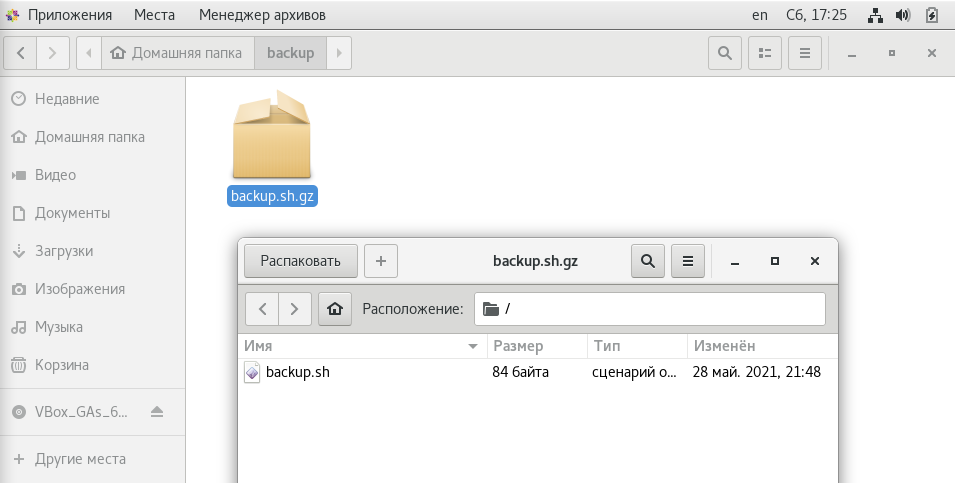


Figure 4: Получившаяся резервная копия

1. Написал пример командного файла, обрабатывающего любое произвольное число аргументов, в том числе превышающее десять. Скрипт последовательно распечатывал значения всех переданных аргументов.

(рис 5. 5)

Figure 5: Создание файла

Figure 5: Создание файла

(рис 6. 6)

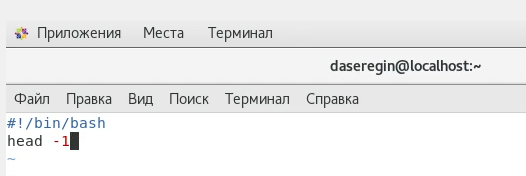


Figure 6: Код командного файла

(рис 7. 7)

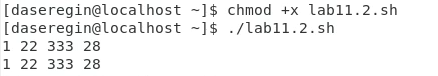


Figure 7: Создание исполняемого файла и проверка его работы

1. Написал командный файл - аналог команды ls. Файл выдаёт информацию о нужном каталоге и выводит информацию о возможностях доступа к файлам этого каталога.

(рис 8. 8)

Figure 8: Создание файла

Figure 8: Создание файла

(рис 9. 9)

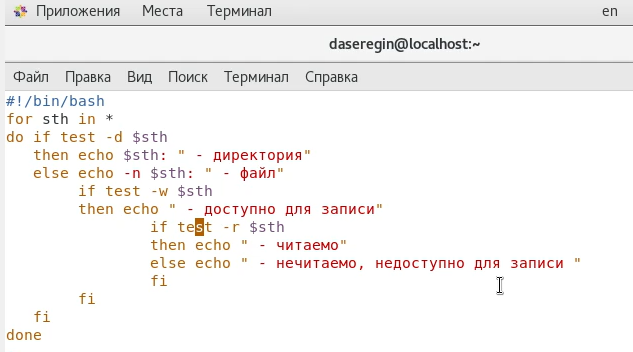


Figure 9: Код программы

(рис 10. 10)

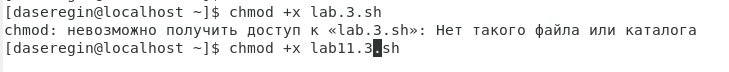


Figure 10: Создание исполняемого файла

(рис 11. 11)

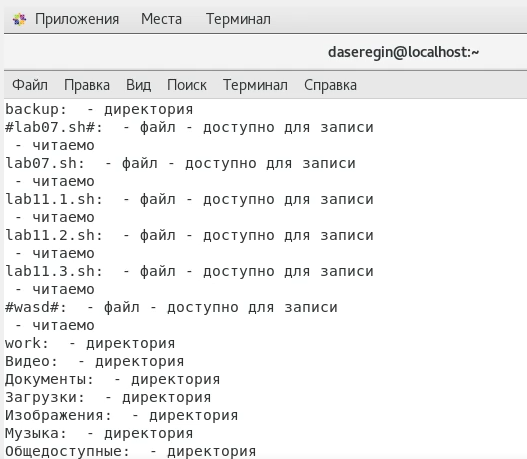
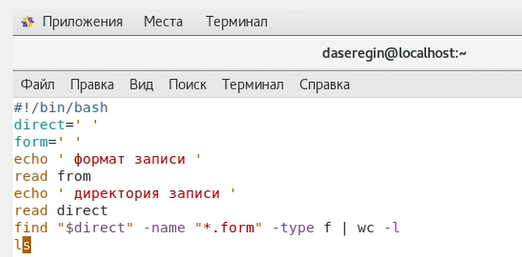
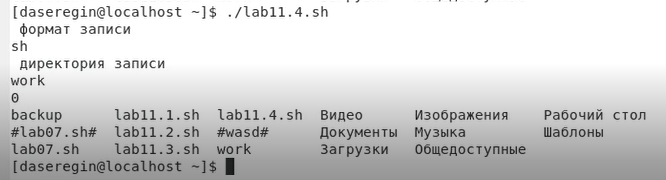


Figure 11: Результат работы кода

1. Написал командный файл который получает в качестве аргумента командной строки формат файла и вычисляет количество таких файлов в указанной директории. Путь к директории также передаётся в виде аргумента командной строки.

* (рис 12. 12)
* Figure 12: Создание файла
* Figure 12: Создание файла
* (рис 13. 13)
* 
* Figure 13: Код программы
* (рис 14. 14)
* Figure 14: Создание исполняемого файла
* Figure 14: Создание исполняемого файла
* (рис 15. 15)
* 
* Figure 15: Результат работы кода

# Выводы

В ходе лабораторной работы мне удалось изучить основы программирования в оболочке ОС UNIX/LINUX, а также научиться писать небольшие командные файлы.

# Использованные ресурсы

https://vps.ua/wiki/creating-unpacking-files-linux/

https://habr.com/ru/company/ruvds/blog/325522/