Лабораторная работа №12

Программирование в командном процессоре ОС UNIX. Командные файлы

Буллер Татьяна Александровна

Содержание

# 1 Цель работы

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

# 2 Выполнение лабораторной работы

## 2.1 Резервное копирование

Скрипт для резервного копирования можно написать в одну строку: для этого используем команду архиватора, которой укажем создать файл в заранее созданной директории backup, и подадим в качестве второго аргумента имя исполняемого файла ($0). Для того, чтобы все резервные копии различались можду собой, пропишем первому аргументу сохранять файл не просто со статичным названием, а добавляя дату создания в конец (переменная дата в формате год+месяц+число-час+минута+секунда).

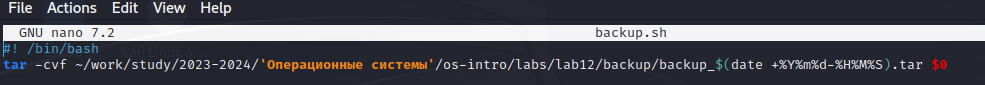


Рис. 1: Скрипт резервного копирования

После исполнения скрипта проверим директорию backup командой ls: видим, что создался файл архива резервной копии, в названии которого прописана дата его создания.

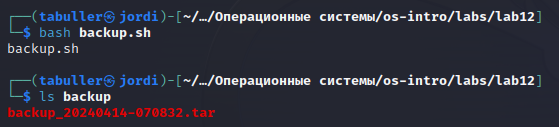


Рис. 2: Результат выполнения скрипта

## 2.2 Последовательная печать аргументов

Мы уже обрабатывали аргумент $0 в предыдущем скрипте. Теперь обработаем все, что будет подано вместе с командой запуска скрипта: проще всего для этого использовать цикл. Пока значение i существует, будет печататься аргумент с соответствующим номером.

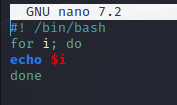


Рис. 3: Скрипт печати аргументов

Подадим на печать сколько угодно много аргументов и проверим, справляется ли скрипт с поставленной задачей:

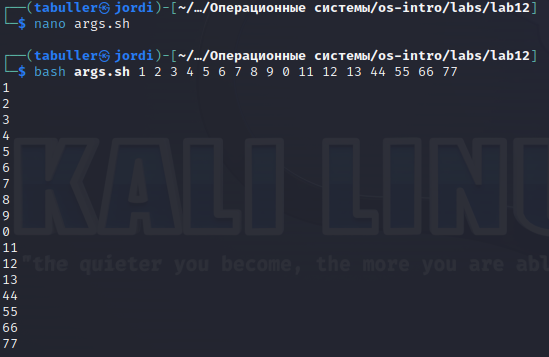


Рис. 4: Результат выполнения скрипта

## 2.3 Аналог ls

Работая с этим скриптом будем использовать цикл для всех файлов текущего каталога. Сперва проверим, является ли файл директорией (test -d) и если да, то выведем информацию об этом, выделив название файла жирным шрифтом. Все остальные файлы, очевидно, являются файлами. Проверим их на возможность исполнения, записи или чтения (-х -w -r) и выведем соответствующие сообщения. Если файл с разрешением на запись - выделим его зеленым, если ни одного из разрешений у файла нет - выделим красным.

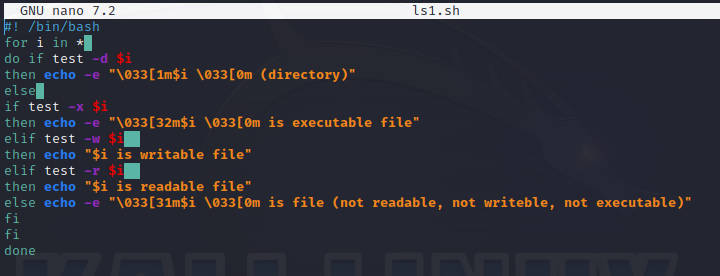


Рис. 5: Скрипт распечатки файлов

Проверим: сам скрипт выполняется. Для того, чтобы проверить корректность отображения цветов, изменим разрешения у пары файлов и снова посмотрим на вывод.

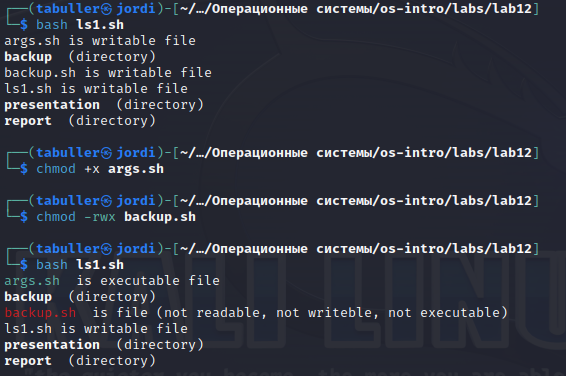


Рис. 6: Результат выполнения скрипта

## 2.4 Счетчик-фильтр

Снова работаем с аргументами на входе. Первым аргументом принимаем формат, вторым - зону поиска. Для реализации скрипта сохраним в массив вывод команды grep по результату ls нужного каталога и выведем на экран количество элементов массива.

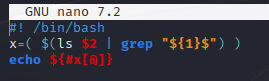


Рис. 7: Скрипт счетчика

Проверим для рабочей директории (не подавая второго аргумента) и для случайно выбранной директории. В обоих случаях скрипт справляется с поставленной задачей корректно.

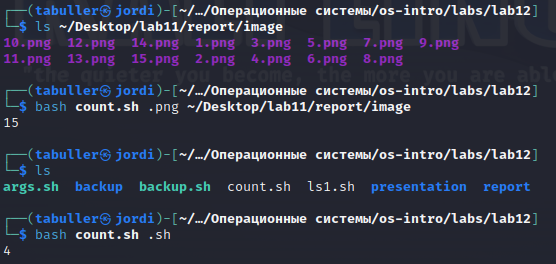


Рис. 8: Результат выполнения скрипта

# 3 Выводы

Изучены основы программирования в оболочке ОС UNIX/Linux. Приобретены навыки написания небольших командных файлов.