Лабораторная работа № 10

Программирование в командном процессоре ОС UNIX. Командные файлы

Шулуужук Айраана Вячеславовна НПИбд-02-22

Содержание

# 1 Цель работы

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

# 2 Задание

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 ар- хиваторов 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 ко- манды 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 ко- мандной строки.

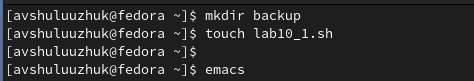
# 3 Теоретическое введение

## 3.1 Командные процессоры (оболочки)

Командный процессор (командная оболочка, интерпретатор команд shell) — это про- грамма, 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; – С-оболочка (или csh) — надстройка на оболочкой Борна, использующая С-подобный синтаксис команд с возможностью сохранения истории выполнения команд; – оболочка Корна (или ksh) — напоминает оболочку С, но операторы управления програм- мой совместимы с операторами оболочки Борна; – BASH — сокращение от Bourne Again Shell (опять оболочка Борна), в основе своей сов- мещает свойства оболочек С 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 опера- ционных 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 оболоч- ках большинство команд будет совпадать с описанными ниже.

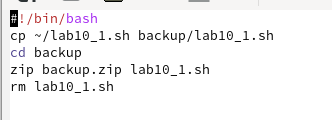
# 4 Выполнение лабораторной работы

Создадим каталог backup и файл lab10\_1.sh. Отроем редактор emacs (рис. ??)



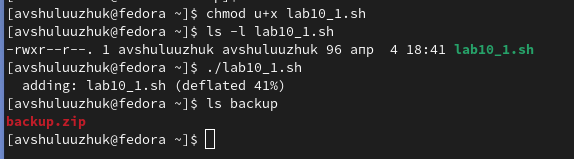
создание каталога и файла

Напишем скрипт, который при запуске будет делать резервную копию самого себя (рис. ??)



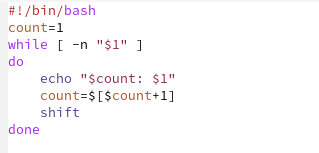
скрипт lab10\_1

Присвоим данному файлу права на исполнение и запустим его (рис. ??)



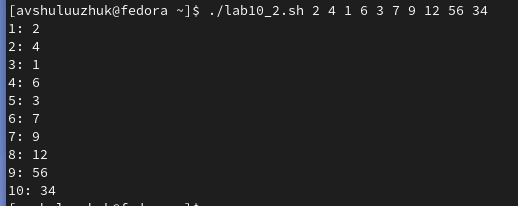
результат запуска скрипта 1

Создаем новый файл lab10\_2.sh и напишем пример командного файла, обрабатывающее любое число аргументов командной строки (рис. ??)



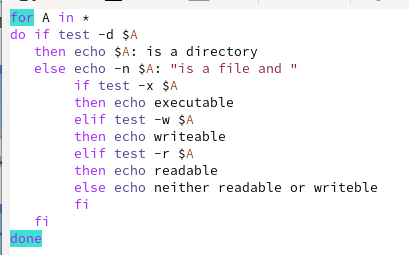
скрипт lab10\_2.sh

Запустим следующий исполняемый файл и введем 10 или более произвольных чисел (рис. ??)



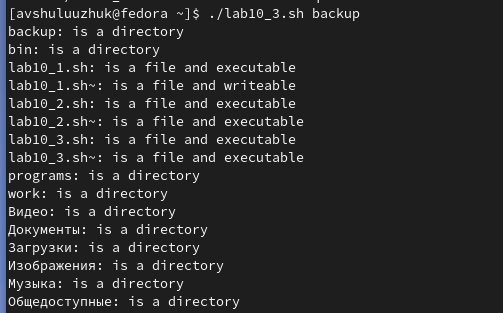
результат запуска скрипта 2

Создаем новый файл lab10\_3.sh для скрипта 3. Напишем командный файл, который воспроизводит аналог команды ls (рис. ??)



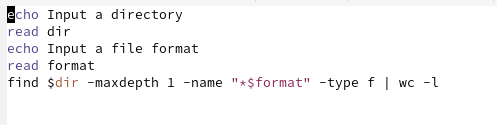
скрипт lab10\_3.sh

Запустим этот файл и просмотрим содержимое домашнего каталога (рис. ??)



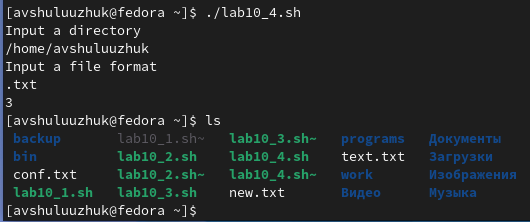
результат запуска скрипта 3

В файле lab10\_4 напишем скрипт, который получает в качестве аргумента командной строки формат файла (.txt, .doc, .jpg, .pdf и т.д.) и вычисляет количество таких файлов в каталоге (рис. ??)



скрипт lab10\_4.sh

Запустим файл и просмотрим количество файлов формата .txt в домашнем каталоге (рис. ??)



результат запуска скрипта 4

# 5 Выводы

В ходе выполнения работы мы изучили основы программирования в оболочке ОС UNIX/Linux и научились писать небольшие командные файлы