Лабораторная работа №10

Операционные системы

Александрова Ульяна

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 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

# 3 Теоретическое введение

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; - 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 программой совместимы с операторами оболочки Борна; - BASH — сокращение от Bourne Again Shell (опять оболочка Борна), в основе своей совмещает 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 операционных систем и переносимости прикладных программ на уровне исходного кода.  
POSIX-совместимые оболочки разработаны на базе оболочки Корна.

# 4 Выполнение лабораторной работы

Я создаю файл программы и прописываю в ней необходимый код (рис. fig. 1).

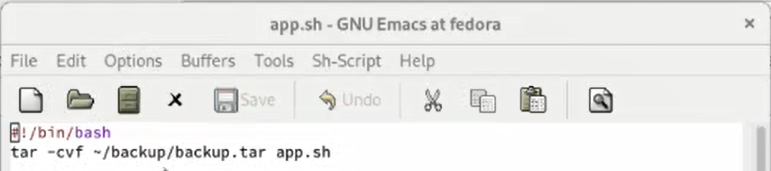


Рис. 1: Программа 1

Я запускаю программу и проверяю выполнение. Программа работает (рис. fig. 2), (рис. fig. 3).

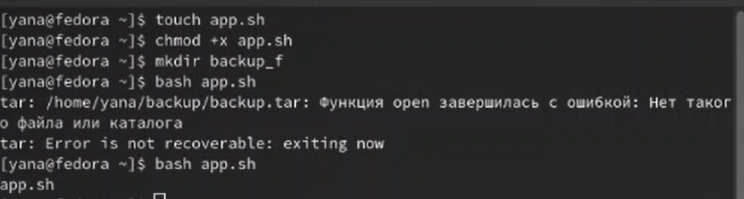


Рис. 2: Запуск программы 1

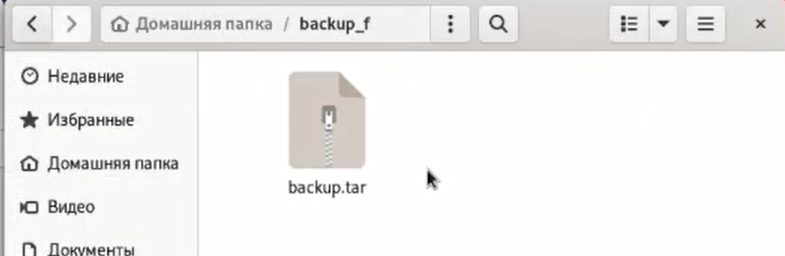


Рис. 3: Архив

Создаю файл для второй программы и также ввожу код (рис. fig. 4).

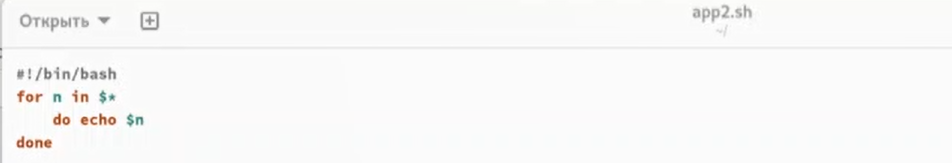


Рис. 4: Создание программы 2

Проверяю работу командного файла. Все работает (рис. fig. 5).

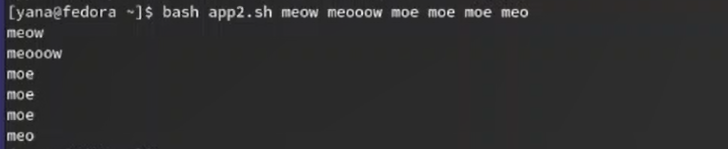


Рис. 5: Проверка программы 2

Создаю третий командный файл, редактирую (рис. fig. 6).

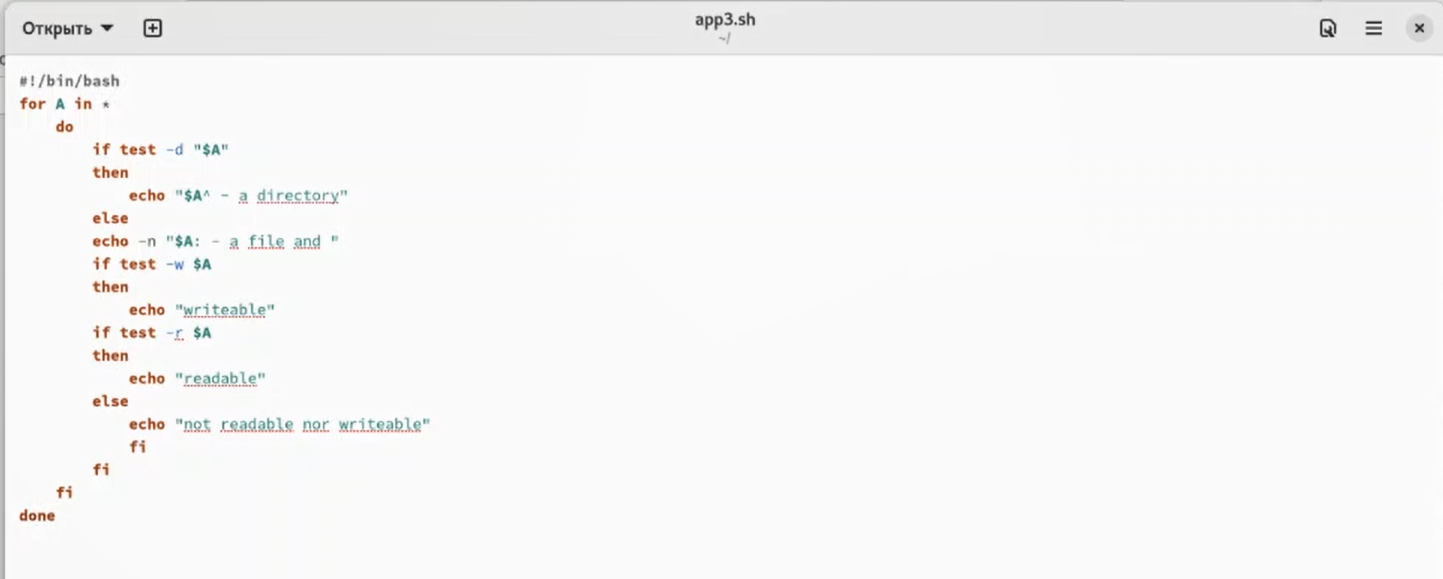


Рис. 6: Написание программы 3

Проверяю работу. Все корректно (рис. fig. 7).

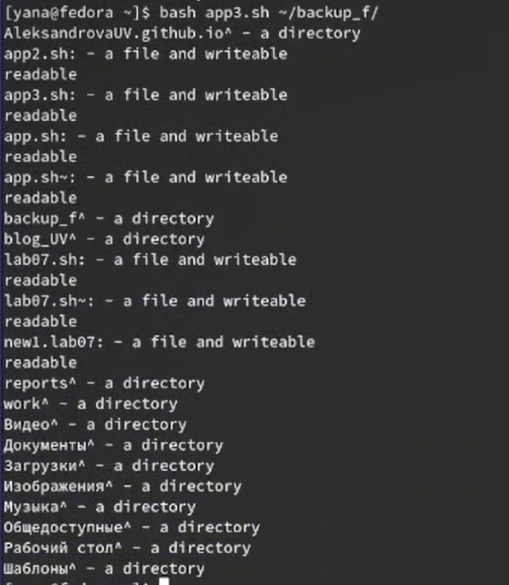


Рис. 7: Запуск программы 3

Создаю четвертый файл, заполняю (рис. fig. 8).

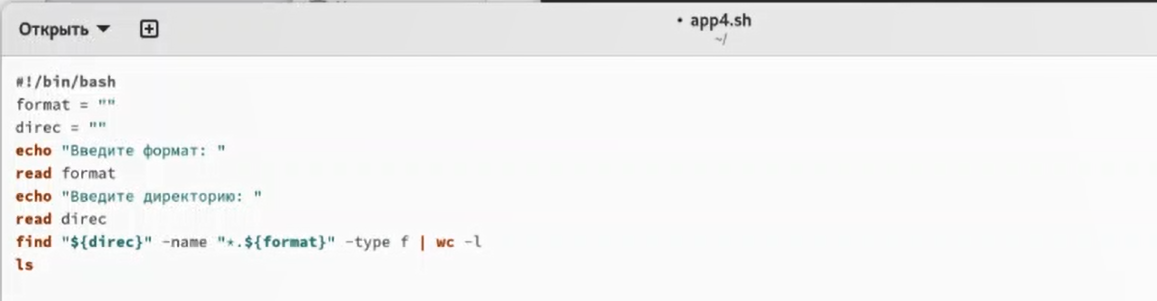


Рис. 8: Создание программы 4

Файл работает корректно (рис. fig. 9).

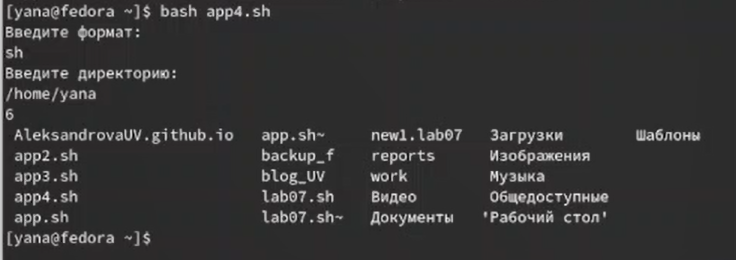


Рис. 9: Проверка программы 4

# 5 Выводы

Я изучила основы программирования в оболочке ОС UNIX/Linux, научилась писать небольшие командные файлы.

:::