Лабораторная работа №10

Программирование в командном процессоре ОС UNIX. Командные файлы

Панченко Денис Дмитриевич

Содержание

# 1 Цель работы

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

# 2 Выполнение лабораторной работы

Создаем файл для написания первого скрипта (рис. [1](#fig:001)).

Figure 1: Файл

Figure 1: Файл

Узнаем об архиваторе tar (рис. [2](#fig:002)).

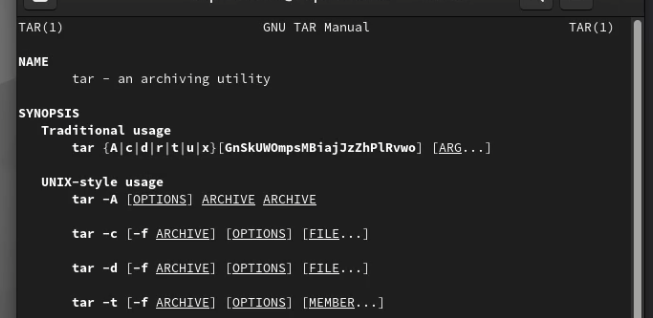


Figure 2: Архиватор

Пишем скрипт, который при запуске будет делать резервную копию самого себя в другую директорию backup в нашем домашнем каталоге (рис. [3](#fig:003)).

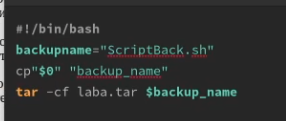


Figure 3: Скрипт

Создаем файл для второго скрипта (рис. [4](#fig:004)).

Figure 4: Файл

Figure 4: Файл

Пишем скрипт, обрабатывающий любое произвольное число аргументов командной строки (рис. [5](#fig:005)).

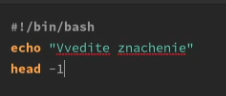


Figure 5: Скрипт

Проверяем скрипт (рис. [6](#fig:006)).

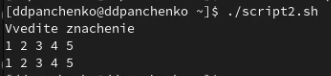


Figure 6: Проверка

Создаем третий файл для скрипта (рис. [7](#fig:007)).

Figure 7: Файл

Figure 7: Файл

Пишем командный файл — аналог команды ls (рис. [8](#fig:008)).



Figure 8: Вырезка

Проверяем скрипт (рис. [9](#fig:009)).

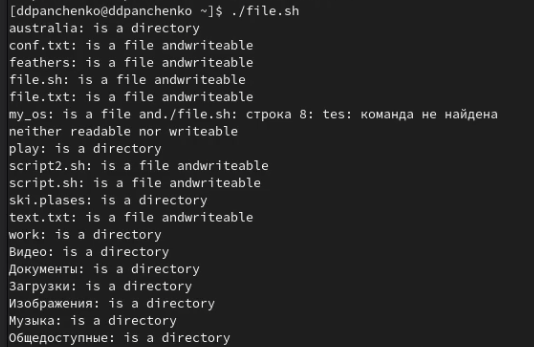


Figure 9: Проверка

Создаем файл для четвертого скрипта (рис. [10](#fig:010)).

Figure 10: Файл

Figure 10: Файл

Пишем командный файл, который получает в качестве аргумента командной строки формат файла и вычисляет количество таких файлов в указанной директории. (рис. [11](#fig:011)).

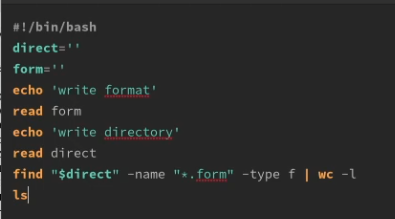


Figure 11: Скрипт

Проверяем скрипт (рис. [12](#fig:012)).

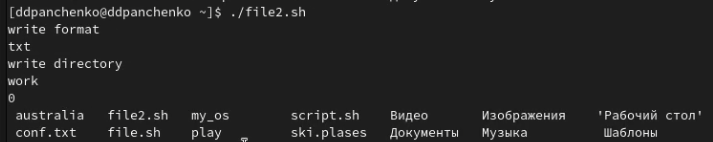


Figure 12: Проверка

# 3 Вывод

Я изучил основы программирования в оболочке ОС UNIX/Linux. Научиляся писать небольшие командные файлы.

# 4 Контрольные вопросы

1. Командная оболочка (shell) - это интерпретатор, который принимает команды пользователя и выполняет их на уровне операционной системы. Примеры командных оболочек: bash, sh, csh, ksh, zsh. Они отличаются синтаксисом и набором возможностей.
2. POSIX (Portable Operating System Interface) - это набор стандартов, которые определяют поведение операционной системы и интерфейс прикладных программ. Это позволяет написать переносимый код, который будет работать на разных операционных системах (Linux, macOS, Unix, Windows и др.).
3. Переменные задаются в виде name=value, например x=10. Доступ к переменной осуществляется через знак $, например $x. Массивы задаются в виде array=(value1 value2 value3), а доступ к элементу массива осуществляется через индекс в квадратных скобках, например ${array[0]}.
4. Оператор let используется для выполнения арифметических операций и присваивания результата переменной, например let x=1+2. Оператор read используется для чтения ввода пользователя в переменную, например read x.
5. В bash можно использовать арифметические операции +, -, \*, /, %, \*\*.
6. Операция (( )) используется для выполнения арифметических операций и вычисления выражений в скобках, например ((x=1+2)).
7. Некоторые стандартные имена переменных: $HOME (домашняя директория), $PATH (список директорий для поиска исполняемых файлов), $USER (имя текущего пользователя), $PWD (текущий рабочий каталог).
8. Метасимволы - это специальные символы, которые имеют особое значение в командной строке. Примеры: \* (соответствует нулю или более символов), ? (соответствует одному символу), [ ] (соответствует символам из заданного набора).
9. Метасимволы можно экранировать с помощью обратной косой черты \. Например, \$.
10. Командный файл создается в текстовом редакторе и сохраняется с расширением .sh. Запустить его можно с помощью команды ./script.sh. Необходимо предварительно дать файлу право на выполнение с помощью команды chmod +x script.sh.
11. Функция задается в виде function name { commands }. Вызвать функцию можно, например, name.
12. Для проверки типа файла можно использовать команду test -d filename для проверки, является ли файл каталогом, и test -f filename для проверки, является ли файл обычным файлом.
13. set используется для задания опций командной строки и установки значений специальных переменных. typeset используется для определения типа переменной (например, числовая или строковая). unset используется для удаления переменной.
14. Параметры передаются в виде аргументов командной строки. В командном файле к ним можно обратиться с помощью переменных $1, $2 и т.д.
15. Специальные переменные: $0 (имя командного файла), $# (количество аргументов командной строки), $@ (список всех аргументов командной строки), $? (код возврата последней выполненной команды).