Отчет о выполнении лабораторной работы

Лабораторная работа №12

Филипьева Ксения Дмитриевна

Содержание

# 1 Цель работы

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

# 2 Задание

Приобрести и отработать уже имеющиеся навыки программирования в оболочке ОС Linux.

# 3 Выполнение лабораторной работы

Создадим папку в которую будут сохраняться бэк-апы для первого задания. (рис. 1).

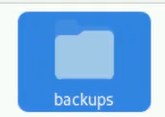


Рис. 1: создание папки

Создадим файл для первого задания и откроем его (рис. 2).

создание файла

Рис. 2: создание файла

Впишем в него код, который позволит выполнять нам поставленные задачи (рис. 3).

код задания

Рис. 3: код задания

Выдадим все права на файл и выполним его (тут я забыла сделать скрин вывода, но архив создался в папке) (рис. 4).

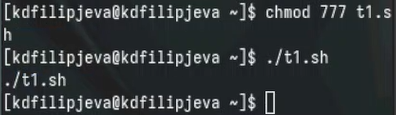


Рис. 4: выдача прав

Создадим файл для второго задания, выдадим все права и откроем его для редактирования (рис. 5).

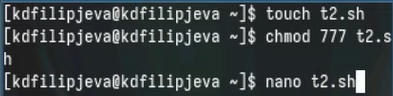


Рис. 5: создание файла

Впишем в него код, позволяющий выполнить поставленное задание (рис. 6).

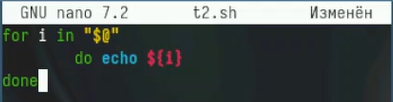


Рис. 6: код для задания

Работоспособность кода (рис. 7).

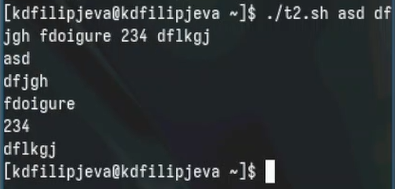


Рис. 7: работоспособность кода №2

Создадим файл для третьего задания, выдадим все права и откроем его для редактирования (рис. 8).

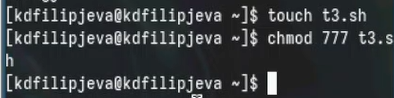


Рис. 8: создание файла

Впишем в него код, позволяющий выполнить поставленное задание (рис. 9).

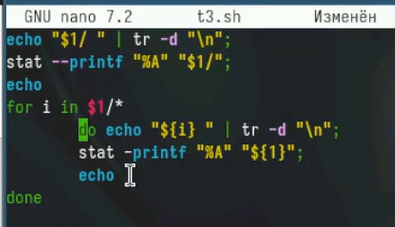


Рис. 9: код для задания

Работоспособность кода (рис. 10).

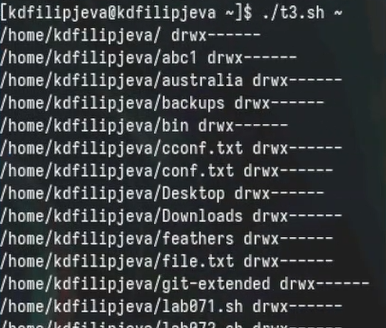


Рис. 10: работоспособность кода №3

Создадим файл для четвертого задания, выдадим все права и откроем его для редактирования (рис. 11).

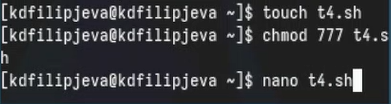


Рис. 11: создание файла

Впишем в него код, позволяющий выполнить поставленное задание (рис. 12).

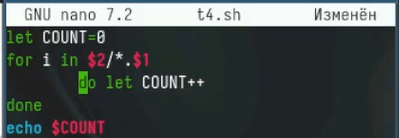


Рис. 12: код для задания

Работоспособность кода (рис. 13).

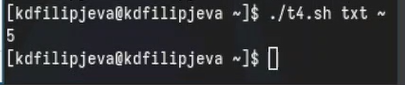


Рис. 13: работоспособность кода №4

# 4 Выводы

Мы получили новые и отработали уже имеющиеся навыки программирования в оболочке ОС Linux.

# 5 Ответы на вопросы

1. Командная оболочка (shell) - это программа, которая обеспечивает интерфейс для взаимодействия пользователя с операционной системой. Примеры: Bash (Bourne Again Shell), Zsh (Z shell), Fish (Friendly Interactive Shell). Они отличаются синтаксисом, встроенными функциями, расширенными возможностями автодополнения и настройками.
2. POSIX (Portable Operating System Interface) - это набор стандартов, определяющих интерфейсы между операционной системой и прикладными программами для обеспечения переносимости.
3. Переменные в Bash определяются как имя=значение. Массивы определяются как имя=(значение1 значение2 ...) или имя[индекс]=значение.
4. let позволяет выполнять арифметические операции, а read считывает ввод пользователя и сохраняет его в переменной.
5. Bash поддерживает основные арифметические операции: сложение (+), вычитание (-), умножение (\*), деление (/), остаток от деления (%), возведение в степень (\*\*).
6. Операция (( )) используется для выполнения арифметических операций и сравнений.
7. Стандартные переменные: HOME (домашний каталог), PATH (список каталогов для поиска команд), USER (имя текущего пользователя) и др.
8. Метасимволы - это специальные символы, имеющие особое значение для командной оболочки (например, \*, ?, [, ]).
9. Метасимволы экранируются с помощью обратного слэша \ или заключаются в кавычки.
10. Командные файлы создаются в текстовом редакторе и сохраняются с расширением .sh. Запуск: bash имя\_файла.sh или ./имя\_файла.sh (при наличии прав на исполнение).
11. Функции в Bash определяются как имя\_функции() { команды; }.
12. Для проверки типа файла используется команда test или [: [ -d файл ] (каталог) или [ -f файл ] (обычный файл).
13. set устанавливает опции командной оболочки, typeset объявляет переменные и их атрибуты, unset удаляет переменные или функции.
14. Параметры передаются в командные файлы через аргументы командной строки, доступные как $1, $2 и т.д.
15. Специальные переменные Bash: $0 (имя скрипта), $# (количество аргументов), $@ (все аргументы в виде отдельных слов), $\* (все аргументы в виде одной строки), $$ (PID текущего процесса) и др.