**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ  
ПО ЛАБОРАТОРНОЙ РАБОТЕ No 11**

*дисциплина: Операционные системы*

Студент: Джунусова Рузель

Группа: НПИбд-01-20

**МОСКВА**

2021 г.

**Цель работы:**

В данной лабораторной работе мне будет необходимо изучить основы программирования в оболочке ОС UNIX/Linux. Будет необходимо научиться писать небольшие командные файлы.

**Выполнение лабораторной работы:**

• ЗАДАНИЕ 1

1. Создала директорию в домашнем каталоге с помощью команды mkdir. С помощью команды emacs создала файл task1.sh. (рис. 4.1)

Изображение выглядит как текст

Автоматически созданное описание

Figure 4.1: Создание директории и файла

1. Cкрипт, который при запуске делает резервную копию самого себя (то есть файла, в котором содержится его исходный код) в другую директорию backup в вашем домашнем каталоге. Файл архивируется архиватором tar. В начале мы вызываем bash. Затем создаём переменную name в которую записываем название файла. Использую команду cp для копирования файла в директорию backup. Далее переходим в каталог и создаём архив.

(рис. 4.2)

Изображение выглядит как текст

Автоматически созданное описание

Figure 4.2: Скрипт

1. Команда для запуска скрипта. (рис. 4.3)



Figure 4.3: Запуск скрипта

1. Скрипт успешно создал резервную копию и архив. (рис. 4.4)

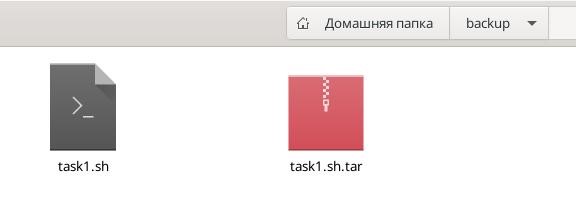


Figure 4.4: Демонстрация работы скрипта

• ЗАДАНИЕ 2

1. С помощью команды emacs создала файл task2.sh. (рис. 4.5)



Figure 4.5: Создание файла

1. Скрипт командного файла, обрабатывающего любое произвольное число аргументов командной строки, в том числе превышающее десять. Скрипт последовательно распечатывает значения всех переданных аргументов. В начале мы вызываем bash. Далее вызываем массив mas. Выводим надпись для запроса данных. Считываем введённые данные. Выводим элементы массива. (рис. 4.6)

Изображение выглядит как текст

Автоматически созданное описание

Figure 4.6: Скрипт

1. Командный файл успешно распечатала значения переданных аргументов.

(рис. 4.7)



Figure 4.7: Демонстрация работы

• ЗАДАНИЕ 3

1. С помощью команды emacs создала файл task3.sh. (рис. 4.8)



Figure 4.8: Создание файла

1. Написала командный файл — аналог команды ls (без использования самой этой команды и команды dir). Скрипт выдаёт информацию о нужном каталоге и выводит информацию о возможностях доступа к файлам этого каталога. В начале мы вызываем bash. Выводим надпись для запроса названия каталога. Считываем введённые данные. Переходим в необходимый каталог. Выводим содержимое каталога. Ключ -с выводит файлы построчно. %A выводит права доступа, а %n выводит названия файлов. (рис. 4.9)

Изображение выглядит как текст

Автоматически созданное описание

Figure 4.9: Скрипт

1. Командный файл успешно вывел названия файлов и права доступа введённого каталога. (рис. 4.10)

Изображение выглядит как текст

Автоматически созданное описание

Figure 4.10: Демонстрация работы

• ЗАДАНИЕ 4

1. С помощью команды emacs создала файл task4.sh. (рис. 4.11)

Изображение выглядит как текст

Автоматически созданное описание

Figure 4.11: Создание файла

1. Написала командный файл, который получает в качестве аргумента командной строки формат файла (.txt,.doc,.jpg,.pdf ит.д.) и вычисляет количество таких файлов в указанной директории. Путь к директории также передаётся в виде аргумента командной строки. В начале мы вызываем bash. Объявляем переменную directory и считываем название дирректории. Объявляем переменную format и считываем необходимый формат файла для поиска. Затем переходим в заданную директорию и находим количество файлов с нужным форматом. (рис. 4.12)

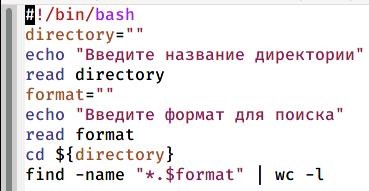


Figure 4.12: Скрипт

1. Командный файл успешно вывел количество файлов с заданным форматом в необходимой директории. (рис. 4.13)

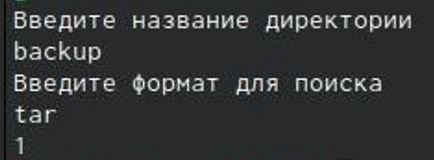


Figure 4.13: Демонстрация работы

**Выводы:**

В данной лабораторной работе мне успешно удалось изучить основы программирования в оболочке ОС UNIX/Linux. Получилось научиться писать небольшие командные файлы.