
Front matter

lang: ru-RU title: "Отчет по лабораторной работе 12" subtitle: "Операционные системы" author: "Дорофеева Алёна Тимофеевна" group: "НПИбд-01-20"

Formatting

fontsize: 12pt linestretch: 1.5

papersize: a4paper

Цель работы

Изучить основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Задание

Проследовав плану выполнения работы, научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Выполнение лабораторной работы

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию

1. Создаем файл и заходим в редактор emacs. После работы в emacs необходимо изменить права доступа (рис.1)

□

Рис. 1: Работа с первым файлом в строке

- 1.1. Используя команды `getopts` `grep`, написать командный файл, который анализирует командную строку с ключами:

- `-i` — прочитать данные из указанного файла;
- `-o` — вывести данные в указанный файл;
- `-r` — указать шаблон для поиска;
- `-C` — различать большие и малые буквы; –
- `-n` — выдавать номера строк.

(рис.1.1.1, рис.1.1.2, рис. 1.1.3)

□

Рис. 1.1.1: Работа в emacs

□

Рис. 1.1.2: Продолжение работы в emacs

□

Рис. 1.1.3: Продолжение работы в emacs

- 1.2. Создание `.txt` файлов для работы, заполняем первый (рис.1.2)

□

Рис. 1.2: Создание и заполнение .txt файлов

- 1.3. Проверяем работу написанного скрипта, ищущего в указанном файле нужные строки, определяемые ключом `-r` (рис.1.3)

□

Рис. 1.3: Исправная работа скрипта

2. Создаем второй файл для работы с ним, открываем `grep`. После работы в `grep` (рис.2.1), создаем файл, открываем `emacs` и пишем скрипт (рис. 2.2), меняем права доступа и запускаем - наблюдаем исправную работу (рис.2)

□

Рис. 2: Выполнение второго задания в строке

□

Рис. 2.1: Работа в grep

□

Рис. 2.2: Работа в etacs

3. Создаем третий файл, запускаем etacs и меняем права доступа (рис. 3)

□

Рис. 3: Работа с файлами в строке

3.1. Написан скрипт, создающий указанное число файлов, пронумерованных последовательно от 1 до N, число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же скрипт удаляет все созданные им файлы (рис.3.1)

□

Рис. 3.1: Работа с файлом в etacs

3.2. Проверяем работу команды в строке (рис. 3.2)

□

Рис. 3.2: Проверяем исправную работу написанного скрипта

4. Создаем четвертый файл, запускаем etacs и меняем права доступа (рис.4)

□

Рис. 4: Работа в консоли

4.1. Написан командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории. Модифицирован так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (рис.4.1)

□

Рис. 4.1: Работа в etacs

4.2. Запускаем и проверяем работу программы в строке (рис. 4.2)

□

Рис. 4.2: Запуск программы в строке

4.3. Наблюдаем создание архива (рис.4.3)

□

Рис. 4.3: Исправная работа написанного скрипта

Выводы

Работа была проделана полностью, как и с лабораторной работой №11, возникли некоторые сложности с программами, пришлось проводить вне записи некоторые установки и проверять работу кодов.