РУДН. Операционные системы

Отчёт по лабораторной работе №9

Косинов Никита Андреевич, НПМбв-02-20

Содержание

# 1 Цель работы

Основой пользования ЭВМ и его работы явлются программы - блоки последовательно выполняемых простейших команд. Мы сталкивались ранее с написанием простейших команд и конвейеров в терминале. Но что делать, если нам нужно выполнить множество одинаковых, или зависящих от условия команд, или чтобы они выполнялись автоматически?

Оболочка ОС **Linux** позволяет базово программировать прямиком в терминале и даже сохранять блоки команд в текстовых, но исполняемых файлах. Цель данной работы - познакомиться с основами, предлагаемыми терминалом **Linux** для программирования.

# 2 Ход работы

Лабораторная работа выполнена в терминале **OC Linux**, командная оболочка **bash** и хостинге хранения проектов **Github**. Действия по лабораторной работе представлены в следующем порядке:

1. написание простейшего скрипта;

1. написание командного файла, обрабатывающего аргументы;
2. написание командного файла - аналог команды *ls*;
3. написание командного файла, вычисляющего количество файлов по формату.

По завершении отчёта, вся рабочая папка отправляется на репозиторий на *github*.

# 3 Написание простейшего скрипта

Перед началом работы создадим новый рабочий каталог **lab09** и перейдём внутрь. Также не забываем синхронизироваться с нашим **Git**.

Первое задание состоит в том, чтобы написать скрипт, копирующий архив себя в папку **backup**.

1. Создадим новый текстовый файл командой *touch*. Назовём его *saveself*. Откроем его с помощью *gedit* на запись.
2. Введём последовательность команд. Сначала скопируем файл *saveself* в нужную папку, затем скопированный файл заархивируем с помощью команды *tar*. Получим следующий код:

’’’ cp saveself backup tar -cf backup/saveself saveself ’’’

1. Сохраним файл, сделаем его исполняемым командой *chmod* и вызовем с помощью *bash*.
2. Получим требуемый результат.

# 4 Написание командного файла, обрабатывающего аргументы

Второе задание: написать командный файл, распечатывающий последовательно все введённые аргументы. Сложность задания в том, что аргументов может быть более 10

1. Создаём новый текстовый файл *replic* и открываем его на редактирование.
2. Идея решения состоит в том, чтобы пройтись циклом *for* по всей строке аргументов, которая воспринимается циклом как массив. Напишем следующий код:

’’’ for s in $\* do echo $s done ’’’

1. Запустим скрипт, введя много переменных и 0 переменных.
2. Результат удовлетворительный.

# 5 Написание командного файла - аналог команды *ls*

Третье задание состоит в том, чтобы написать командный файл, выводящий список файлов и каталогов данного с указанием прав доступа.

1. Создаём файл *new\_ls*.
2. Записываем в него следующий код. Идея состоит в том, чтобы сохранить список файлов, посчитать их количество, а далее вевести только названия и права доступа.

’’’ list=($(ls (ls -l {#list[\*]} echo ‘Files and papkas:’ $n x=0 t=2 while ((x<n)) do echo $((x+1)) ‘:’ ${pr[t]} ${list[$x]} x=((t+9)) done ’’’

1. Проверяем полученный результат.
2. Сравниваем его со стандартной функцией *ls -l*

# 6 Написание командного файла, вычисляющего количество файлов по формату

Четвёртая задача заключается в написании файла, принимающему на вход 2 аргумента: путь к директории и формат файлов, количество которых нужно посчитать.

1. Создаём файл *format*

# 7 Выводы

Встроенный текстовый редактор **vi** имеет множество функций по редактированию текста. Множество команд позволяют быстро и без использования мыши изменять текст, упрощают навигацию по файлу и пр.