

Лабораторная работа №9

Архитектура ОС

Демидович. Н. М.

15 апреля 2023

Российский университет дружбы народов, Москва, Россия

Докладчик

- Демидович Никита Михайлович
- Студент группы НКАбд-01-22
- Кафедра прикладной информатики и теории вероятностей
- Российский университет дружбы народов
- 1132221550@pfur.ru
- <https://github.com/nikdem1>



Цели и задачи

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

Задачи

Последовательность выполнения работы:

1. Написать скрипт, который при запуске будет делать резервную копию самого себя (то есть файла, в котором содержится его исходный код) в другую директорию backup в вашем домашнем каталоге. При этом файл должен архивироваться одним из архиваторов на выбор zip, bzip2 или tar. Способ использования команд архивации необходимо узнать, изучив справку.
2. Написать пример командного файла, обрабатывающего любое произвольное число аргументов командной строки, в том числе превышающее десять. Например, скрипт может последовательно распечатывать значения всех переданных аргументов.
3. Написать командный файл — аналог команды ls (без использования самой этой команды и команды dir). Требуется, чтобы он выдавал информацию о нужном каталоге и выводил информацию о возможностях доступа к файлам этого каталога.

Теоретическое введение

Командный процессор (командная оболочка, интерпретатор команд shell) — это программа, позволяющая пользователю взаимодействовать с операционной системой компьютера. В операционных системах типа UNIX/Linux наиболее часто используются следующие реализации командных оболочек:

- оболочка Борна (Bourne shell или sh) — стандартная командная оболочка UNIX/Linux, содержащая базовый, но при этом полный набор функций;
- C-оболочка (или csh) — надстройка на оболочкой Борна, использующая C-подобный синтаксис команд с возможностью сохранения истории выполнения команд;
- оболочка Корна (или ksh) — напоминает оболочку C, но операторы управления программой совместимы с операторами оболочки Борна;
- BASH — сокращение от Bourne Again Shell (опять оболочка Борна), в основе своей совмещает свойства оболочек C и Корна (разработка компании Free Software

Выполнение лабораторной работы

Далее описан ход выполнения данной лабораторной работы.

На первом этапе выполнения работы я создал командный файл programm1.sh и приступил к написанию скрипта, который при запуске делает резервную копию самого себя в директорию backup в домашнем каталоге. Затем я создал данную директорию, после чего запустил скрипт и проверил его работу.

```
tar -cvf ~/backup/backup.tar programm1.sh
```

После этого я приступил к выполнению второго задания. Мною был написан командный файл programm2.sh, который обрабатывает любое произвольное число аргументов командной строки, в том числе превышающее десять (10).

```
# echo 'Введите любые числа'
# read n
for A in $*
do echo $A
done
```

Далее я написал командный файл programm3.sh, который является аналогом команды ls (без использования самой этой команды и команды dir): он выдает информацию о нужном каталоге и выводит информацию о возможностях доступа к файлам этого каталога.


```
for A in *
do
    if test -d "$A"
    then
        echo -n "$A: is a directory"
    else
        echo -n "$A: is a file and "
        if test -w $A
        then
            echo writeable
            if test -r $A
            then
                echo "readable"
            else
```

И на финальном этапе выполнения работы я создал командный файл programm4.sh, который получает в качестве аргумента командной строки формат файла (.txt, .doc, .jpg, .pdf и т.д.) и вычисляет количество таких файлов в указанной директории.

```
format=""
directory=""
echo "Введите формат файла: "
read format
echo "Введите директорию: "
read directory
find "${directory}" -name "*.${format}" -type f | wc -l
ls
```

Результаты

В результате выполнения данной лабораторной работы я изучил основы программирования в оболочке ОС UNIX/Linux и научился писать небольшие командные файлы.

Список источников

Лабораторная работа №10 (Архитектура ОС).