

Front matter

lang: ru-RU title: Программирование в командном процессоре ОС UNIX. Командные файлы. subtitle: ЛР по ОС №11 author: Танрибергенов Эльдар Марсович group: НПИбд-02-20

Отчёт

о выполнении лабораторной работы № 11.

Программирование в командном процессоре ОС UNIX. Командные файлы.

Выполнил:

студент группы НПИбд-02-20

Танрибергенов Эльдар Марсович.

Студ. билет № 1032208074

Москва 2021 г.

Цель работы:

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

Задание:

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.
4. Написать командный файл, который получает в качестве аргумента командной строки формат файла (.txt, .doc, .jpg, .pdf и т.д.) и вычисляет количество таких файлов в указанной директории. Путь к директории также передаётся в виде аргумента командной строки.

Теоретическое введение

[\[1\]](#)

Командный файл (скрипт или сценарий) - это текстовый файл, состоящий из команд интерпретатора.

При запуске этого файла последовательно выполняются все команды, содержащиеся в нем.

Как файл его можно создать командой `cat` или использовать текстовый редактор, например, `vi`.

Для тех, кто привык работать с текстовым редактором, встроенным в файловый менеджер Midnight Commander (МС - запуск), в котором имеется встроенный текстовый редактор с более привычным интерфейсом.

Краткий справочник команд Linux [\[2\]](#)

Ход работы:

Для начала нужно узнать адрес `bash` для того чтобы прописать его в скрипте. Делаю это командой `whereis bash`. Открыл файл `~/os/progs/p1` с помощью `vi`. Написал скрипт.

https://github.com/emtanribergenov/OS_labs/blob/master/11/screenshots/1.png

Командный файл 1.

https://github.com/emtanribergenov/OS_labs/blob/master/11/screenshots/2.png

Далее сделал файл `p1` исполняемым. Проверил права. Запустил скрипт. Проверил результат работы скрипта. Видно, что скрипт написан верно: в директории `backup` появился архив.

https://github.com/emtanribergenov/OS_labs/blob/master/11/screenshots/3.png

Просмотрел содержимое архива с помощью графического интерфейса.

https://github.com/emtanribergenov/OS_labs/blob/master/11/screenshots/4.png

Открыл файл `p2` в `vi`. Прописал скрипт.

Командный файл 2.

https://github.com/emtanribergenov/OS_labs/blob/master/11/screenshots/5.png

Сделал файл `p2` исполняемым при помощи `chmod`. Запустил скрипт с аргументами `>10` штук. Результат работы скрипта.

https://github.com/emtanribergenov/OS_labs/blob/master/11/screenshots/6.png

https://github.com/emtanribergenov/OS_labs/blob/master/11/screenshots/7.png

Командный файл 3.

https://github.com/emtanribergenov/OS_labs/blob/master/11/screenshots/8.png

Сделал исполняемым и запустил его.

https://github.com/emtanribergenov/OS_labs/blob/master/11/screenshots/9.png

https://github.com/emtanribergenov/OS_labs/blob/master/11/screenshots/10.png

Командный файл 4.

https://github.com/emtanribergenov/OS_labs/blob/master/11/screenshots/11.png

Испытал его при помощи каталога `test`. Результаты абсолютно верные.

https://github.com/emtanribergenov/OS_labs/blob/master/11/screenshots/12.png

Вывод:

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

Ответы на вопросы:

1. Командная оболочка — это отдельный программный продукт, который обеспечивает прямую связь между пользователем и операционной системой. Текстовый пользовательский интерфейс командной строки предоставляет среду, в которой выполняются приложения и служебные программы с текстовым интерфейсом.
2. **POSIX** (произносится как «позикс») — это интерфейс портативных операционных систем. Но что это значит? Во-первых, нужно обозначить область действия понятия «портативность», в этом конкретном случае, и определиться с понятием «интерфейс».
3. В BASH могут быть определены одномерные массивы - переменные специального вида, доступ к которым осуществляется с одним именем, но разными индексами. Для управления массивами переменных используются конструкции. `name[n]=value` - задание n-го элемента массива `name==(val1 val2...`
4. Присваивает какое-либо значение переменной.
5. Все стандартные АО.
6. Действие.
7. Если не про встроенные, то: `var`, `i`, `counter` и т.д.
8. Метасимволы — это специальные символы, являющиеся важнейшим понятием в регулярных выражениях.
9. Экранировать метасимволы можно с помощью метасимвола `\` (обратная косая черта), или заключив в одинарные или двойные кавычки.
10. Создание файла > написание скрипта > наделение правом на исполнение > запуск `./`
11. имя функции() {список команд}
12. команда `file`
13. Изменяет значения внутренних переменных сценария, изм. типа
14. через пробел
15. переменные окружения, позиционные параметры, `shift`