Отчёт по лабораторной работе №10

Программирование в командном процессоре ОС UNIX. Командные файлы.

Грузинова Елизавета Константиновна; НКНбд-02-21

Содержание

# 1 Цель работы

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

# 2 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.

# 3 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;

– С-оболочка (или csh) — надстройка на оболочкой Борна, использующая С-подобный синтаксис команд с возможностью сохранения истории выполнения команд;

– оболочка Корна (или ksh) — напоминает оболочку С, но операторы управления программой совместимы с операторами оболочки Борна;

– BASH — сокращение от Bourne Again Shell (опять оболочка Борна), в основе своей совмещает свойства оболочек С и Корна (разработка компании Free Software Foundation). POSIX (Portable Operating System Interface for Computer Environments) — набор стандартов описания интерфейсов взаимодействия операционной системы и прикладных программ. [1]

# 4 Выполнение лабораторной 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. (рис. 1-4)

Рис. 1: Создание файла pr1.sh

Рис. 1: Создание файла pr1.sh

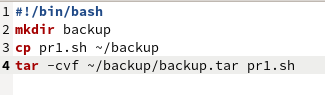


Рис. 2: Программный код файла pr1.sh

Рис. 3: Файл в действии

Рис. 3: Файл в действии

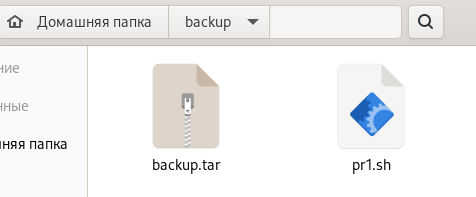


Рис. 4: Созданный программой архив файла

1. 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. (рис. 5, 6)

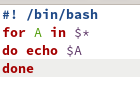


Рис. 5: Программный код файла pr2.sh

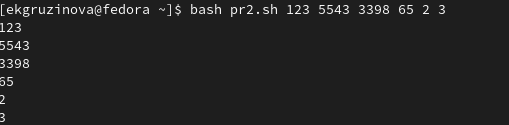


Рис. 6: Скрипт в действии, в данном случае, он записывает необходимые аргументы и распечатывает их обратно

1. 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. (рис. 7, 8)

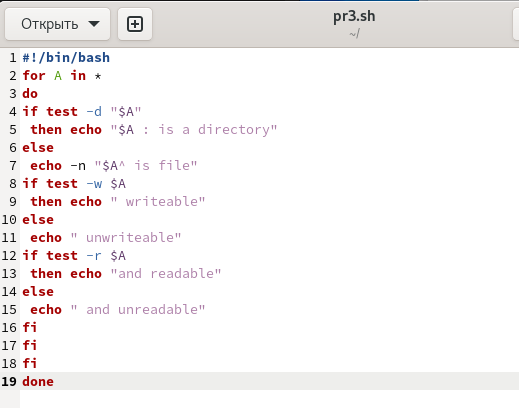


Рис. 7: Программный код файла pr3.sh

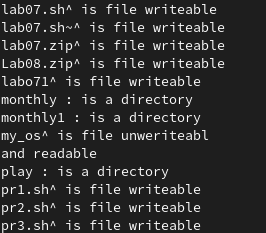


Рис. 8: Вывод информации о каждом файле или каталоге

1. 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. (рис. 9, 10)

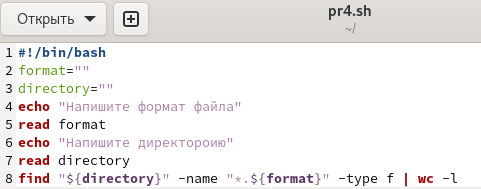


Рис. 9: Программный код файла pr4.sh

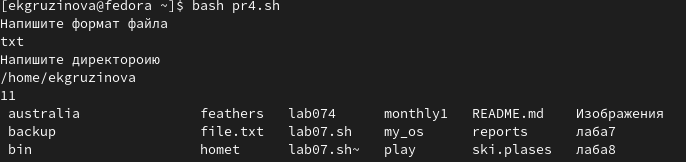


Рис. 10: Вывод количества файлов в /home/gruzinova формата .txt

# 5 Контрольные вопросы

1. Объясните понятие командной оболочки. Приведите примеры командных оболочек. Чем они отличаются?

Командная оболочка - это программа, позволяющая пользователю взаимодействовать с операционной системой компьютера.

Примеры: MC, BASH, VI и т.д.

Они отличаются своей реализацией командных оболочек, т.е. некоторые из них построенные на оболочке bash, Борнаб Корна и т.д.

1. Что такое POSIX?

POSIX (Portable Operating System Interface for Computer Environments) — набор стандартов описания интерфейсов взаимодействия операционной системы и прикладных программ. Разработан для обеспечения совместимости различных UNIX/Linux-подобных операционных систем и переносимости прикладных программ на уровне исходного кода.

1. Как определяются переменные и массивы в языке программирования bash?

Для определения переменной используют имя переменной. присваемое значени и знак “=” между ними.

Для создания массива используется команда set с флагом -A. За флагом следует имя переменной, а затем список значений, разделённых пробелами.

1. Каково назначение операторов let и read?

Команда let берет два операнда и присваивает их переменной.

Команда read присваивает значение, введеное пользователем, той переменной, указанной после команды.

1. Какие арифметические операции можно применять в языке программирования bash?

Сложение (+), вычитание (-), умножение (\*), целочисленное деление (/) и целочисленный остаток от деления (%) и т.д.

1. Что означает операция (( ))?

Вычисление арифметических выражений.

1. Какие стандартные имена переменных Вам известны?

X, Y, Z, trash.

1. Что такое метасимволы?

Метасимволы - это символы вида < > \* ? |  ” &, имеющие определенный смысл для командного процессора.

1. Как экранировать метасимволы?

Для этого необходимо снять специальный символ с матесимвола, то есть заключить в одинарые ковычки. Это не работает с символами $, ’ , , “.

1. Как создавать и запускать командные файлы?

Для создания файла используем touch имя файла. Для запуска пишем bash командный\_файл [аргументы].

1. Как определяются функции в языке программирования bash?

Функция - это группа команд.

1. Каким образом можно выяснить, является файл каталогом или обычным файлом?

С помощью команды test -f имя \_файла.

1. Каково назначение команд set, typeset и unset?

Команда typeset задаёт и/или накладывает ограничения на переменные.

Команда set изменяет значения внутренних переменных сценария.

Команда unset удаляет переменную, фактически – устанавливает ее значение в null.

1. Как передаются параметры в командные файлы?

Часто сценарий написан так, что аргументы могут быть переданы в любом порядке с использованием флагов.

1. Назовите специальные переменные языка bash и их назначение

Переменные - ? # $ !. Они позволяют с помощью команды echo получить следующую информацию:

* – текущие флаги интерпретатора (установка флагов может быть изменена командой set);

2 # – число аргументов, которое было сохранено интерпретатором при выполнении какой-либо команды;

? – код возврата последней выполняемой команды;

# 6 Выводы

В процессе выполнения лабораторной работы изучила основы программирования в оболочке ОС UNIX/Linux и научилась писать небольшие командные файлы.

# Список литературы

1. Лабораторная работа №10. Программирование в командном процессоре ОС UNIX. Командные файлы [Электронный ресурс]. URL: <https://esystem.rudn.ru/pluginfile.php/1383185>.