Министерство образования Республики Беларусь

Учреждение образования

“Гомельский государственный университет имени Ф. Скорины”

Отчет по лабораторной работе

Лабораторная работа №5. **Командные файлы и язык Shell**

Выполнил:

Студент группы КФ-17

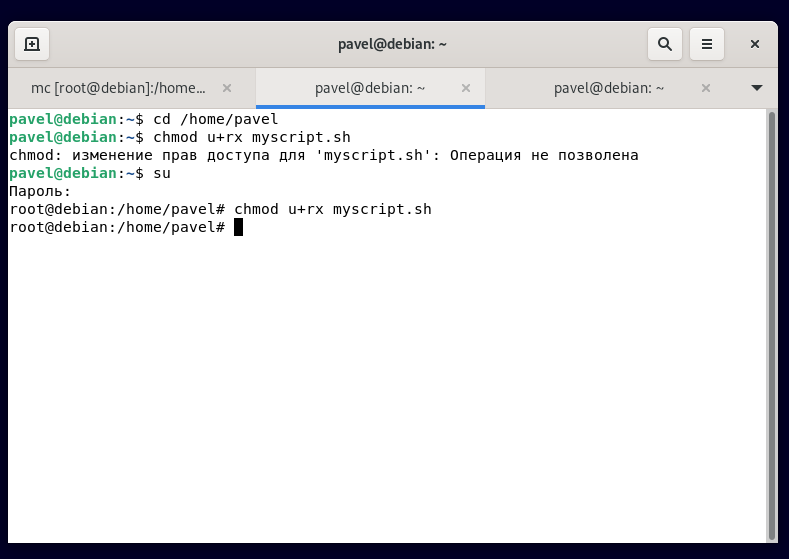
Гуревич П.А.

Гомель 2022

**Цель:** изучение языка сценариев Unix Shell

Программы на языке Shell (скрипты, сценарии) представляют собой обычные текстовые файлы, содержащие набор команд и управляющих конструкций, а также комментарии. Для того чтобы файл сценария можно было выполнить в командном интерпретаторе нужно установить для него права доступа на чтение и выполнение для владельца файла:

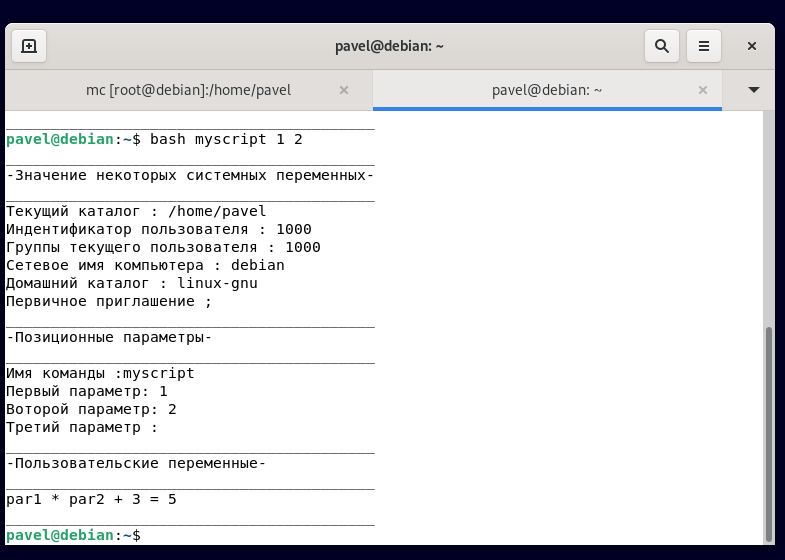
$ chmod u+rx myscript.sh



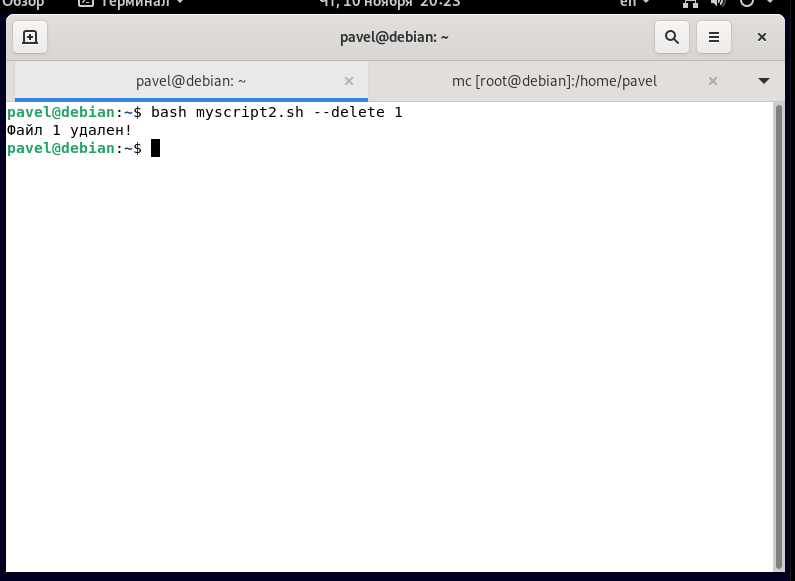


1. Выполните на рабочем компьютере все практические примеры из данной лабораторной работы.

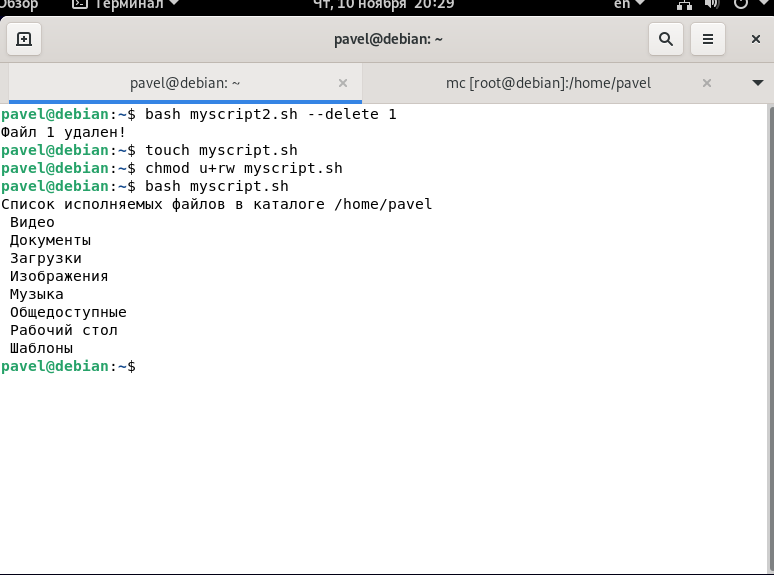
Пример1



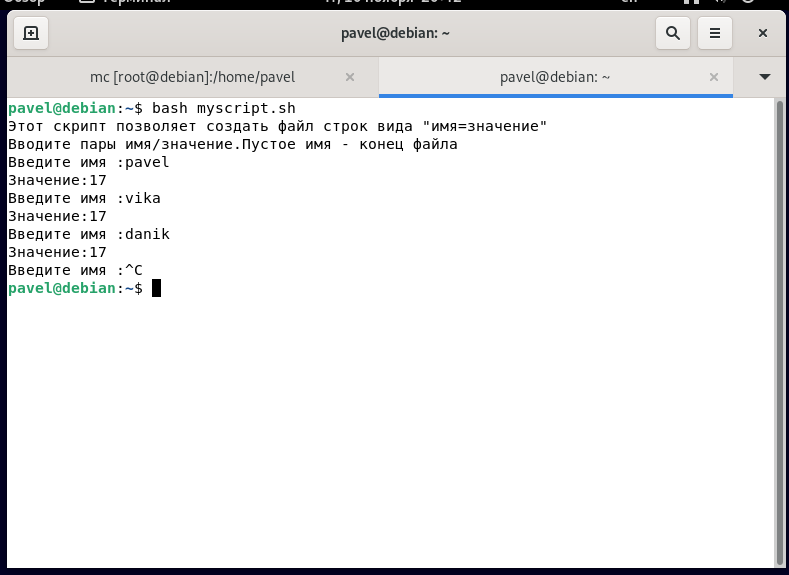
Пример 2

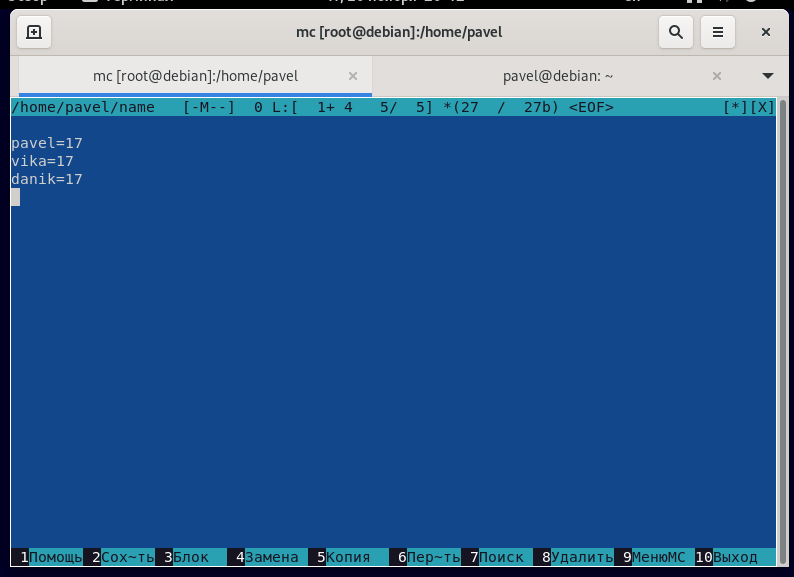


Пример 3

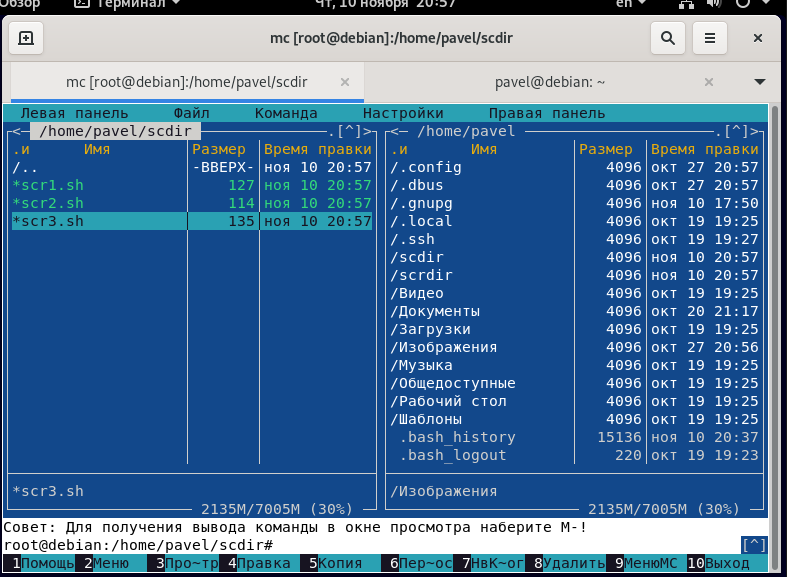


Пример 4

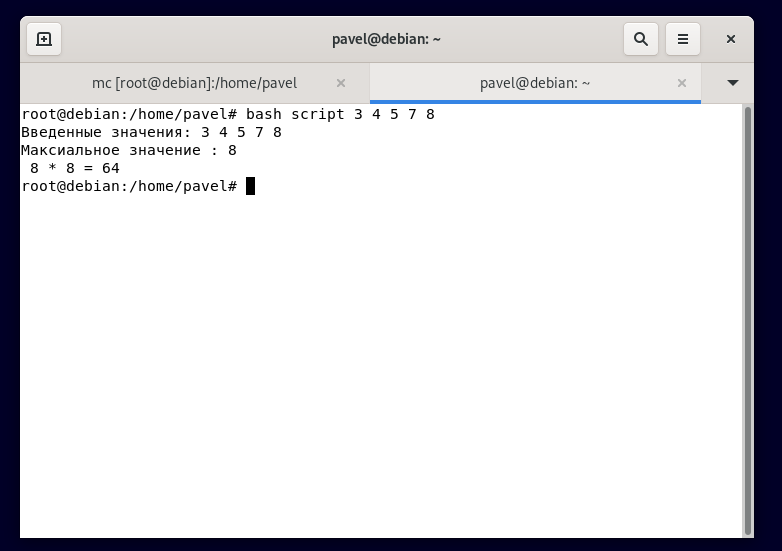




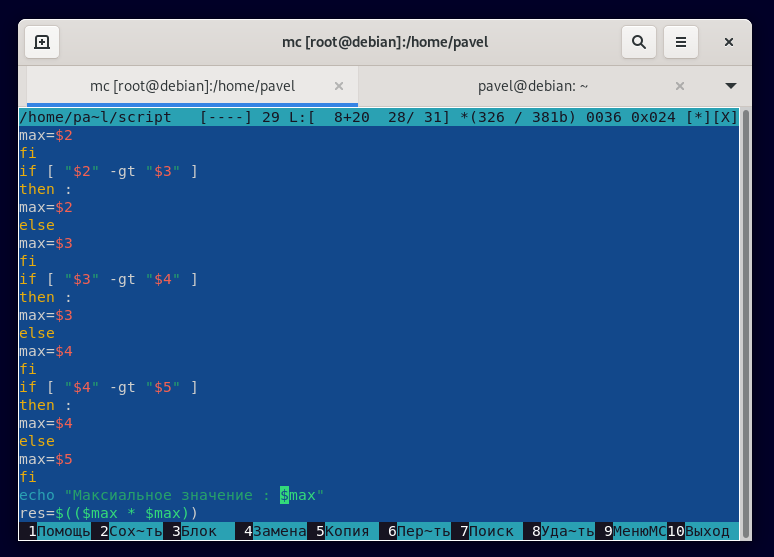
Пример 5



1. Разработайте скрипт, выполняющий следующие действия. Через параметры командной строки ему передаются от одного до пяти чисел. Требуется вывести максимальное значение и сумму квадратов введенных чисел.

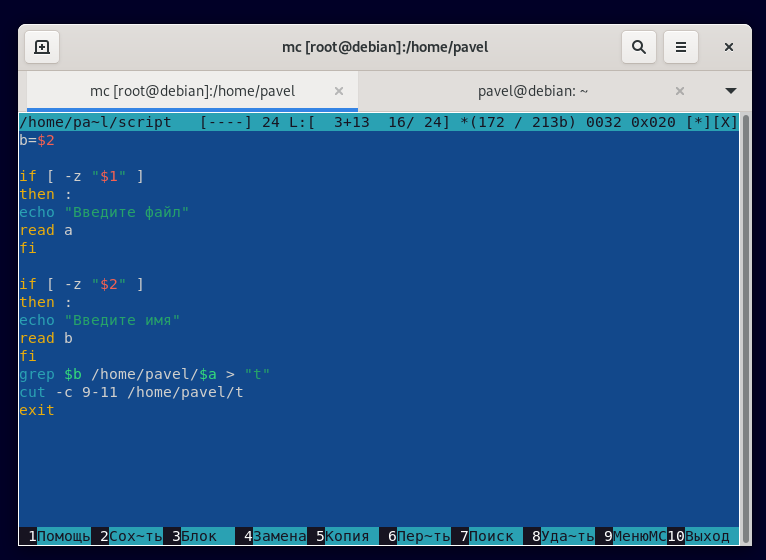


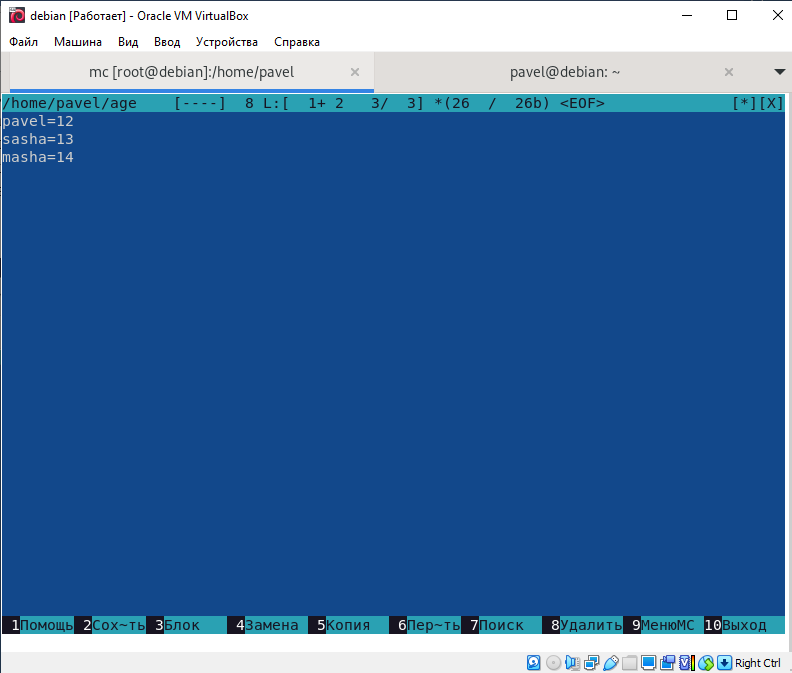
Скрипт :



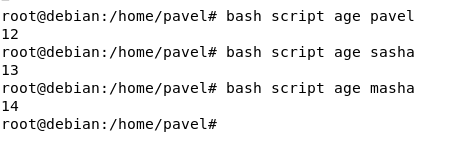
1. Пусть существует файл, содержащий строки вида «имя=значение». Разработайте скрипт, который будет по заданному имени находить соответствующее значение из правой части строки. Имя файла и имя-ключ передаются через параметры командной строки. Если один или оба параметра не заданы, то они запрашиваются у пользователя в интерактивном режиме.

Скрипт :



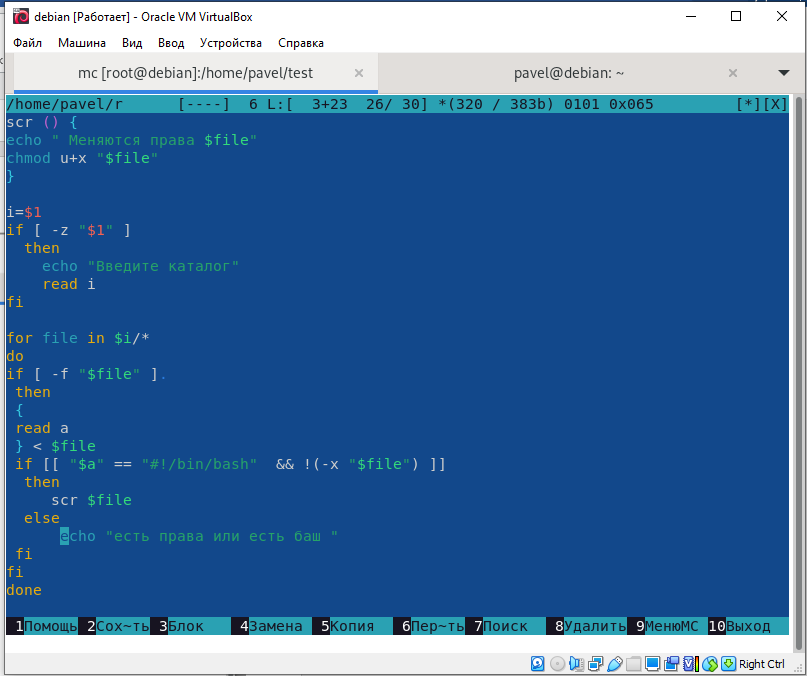


Решение :

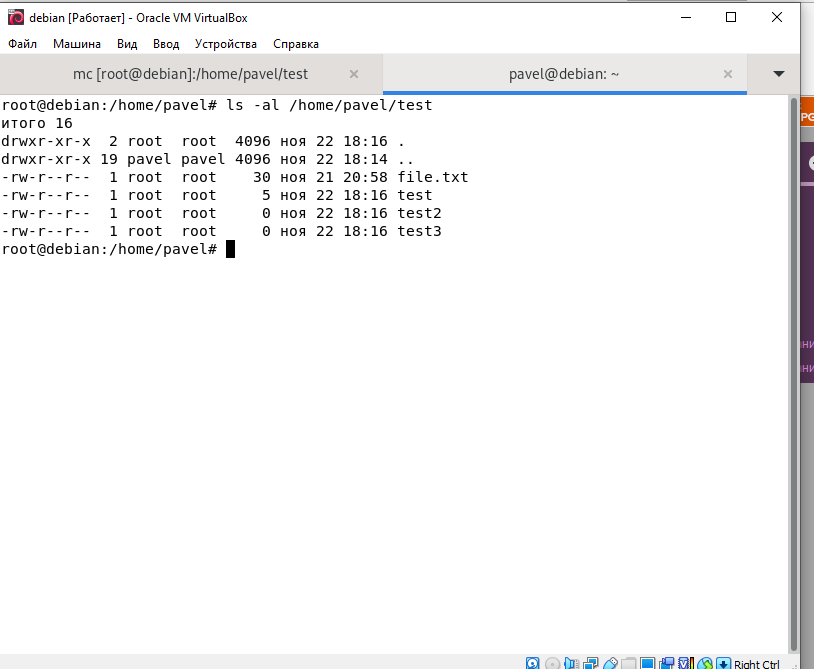


1. Разработайте скрипт, который просматривает все файлы в заданном каталоге. Если первой строкой файла является «#!/bin/bash» и файл не обладает правами на исполнение владельцем, то информация о таком файле выводится на экран и файлу устанавливается право на исполнение владельцем. Каталог передается через первый параметр командной строки. Действия по анализу и изменению прав доступа для некоторого файла оформите в виде функции.

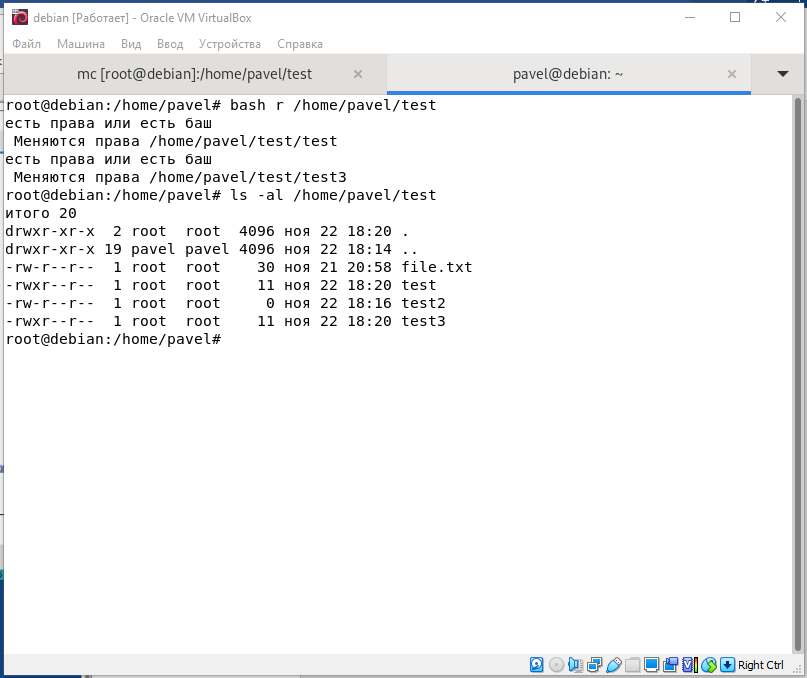
Скрипт :



Пример :



Решение :



**КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Что собой представляет программа на языке Shell? В каких случаях лучше не использовать язык Shell?

Shell — это своеобразный «раствор» для скрепления крупных строительных блоков, которыми являются стандартные утилиты Linux и программы

Если вам нужно написать программу, которая будет выполнена один раз, или быстро создать прототип программы для проверки какой-то идеи, то Shell и другие скриптовые языки — ваш естественный выбор. Основная задача языка Shell — *автоматизация* повторяющихся рутинных операций при работе с компьютером и его настройке. Скрипты инициализации системы написаны на языке Shell.

Сильная сторона языка Shell — небольшое время разработки программ. Часто одну и ту же задачу можно решить, написав за несколько дней программу на C++ размером в несколько тысяч строк кода или написав за полчаса скрипт на Shell, содержащий 30 строк кода.

1. В чем особенность переменных в языке Shell ? Каким образом переменные определяются в программе? Как осуществляется присвоение значения переменной? Как выполняются операции ввода-вывода?

В языке Shell переменные не имеют типа. Все значения хранятся встроковом виде. Числовые значения тоже хранятся в строковом виде и  
обрабатываются с помощью специальных команд.  
Так как переменные не имеют типа, то специально описыватьпеременные не нужно. Нужно просто присвоить переменной значение:  
var=1  
Важным моментом здесь является то, что данное выражение нужно  
писать без пробелов. Если вы напишете присвоение как «var = 1», то оно  
будет трактоваться как запуск команды с именем var и параметрами «=» и  
«1».

1. Что собой представляют позиционные и специальные параметры, переменные окружения?

При запуске любой программы из командной строки мы можем передавать ей ключи и параметры. Точно так же мы можем передавать ключи и параметры скрипту при его запуске. Доступ к значениям ключей и параметров командной строки в скрипте осуществляется с помощью специальных переменных, называемых *позиционными параметрами*. Позиционные параметры имеют имена, совпадающие с номером соответствующего параметра — **$1**, **$2** и т.д. Если номер параметра состоит из двух цифр, то их нужно взять в фигурные скобки, например, **${10}**. В параметре **$0** передается имя скрипта.

Кроме позиционных параметров есть еще один вид встроенных переменных, имеющих специальные значения – специальные параметры. Во время выполнения служебные параметры заменяются некоторым значением.

1. В чем особенности использования различных скобок и кавычек в языке Shell? Как можно вычислить значение арифметического выражения в языке Shell?

При задании строковых констант важным моментом является то, в какие скобки или кавычки они заключены. В зависимости от вида скобок и кавычек строковая константа обрабатывается тем или иным образом. В языке shell строковые константы могут заключаться:

* в двойные кавычки **«“»**,
* в одинарные кавычки (апострофы) **«’»**,
* в обратные кавычки (данный символ расположен на клавиатуре на клавише «тильда» «~») **«`»**.

**Двойные кавычки**(**нестрогие кавычки**) предназначены для объединения в один параметр строк, содержащих пробелы. При этом если строка содержит подстроку **$имя\_переменной**, то в это место строки подставляется значение данной переменной. Одиночные подстановки значений переменных так же рекомендуется заключать в двойные кавычки.

**Одинарные кавычки**(**строгие кавычки**) в точности передают в качестве параметра то значение, которое в них заключено. Никакие подстановки не выполняются.

**Обратные кавычки**предназначены для выполнения команд. Строка, заключенная в обратные кавычки, заменяется выводом команды, содержащейся в обратных кавычках.

Сумма :

$ echo $(((2+3)\*5))   
25

1. Для чего предназначены команды if и test языка Shell? Перечислите основные - 6 - параметры команды test.

Оператор **if** работает следующим образом. Он проверяет коды завершения, возвращаемые командами из **список\_команд1**. Если в этом списке несколько команд, то проверяется код завершения последней команды списка. Если значение кода завершения равно 0 (успешное завершение), то будут выполняться команды из **список\_команд2**, если это значение не нулевое, то будут выполнены команды из **список\_команд3**. Код завершения, возвращаемый таким составным оператором **if**, совпадает с кодом завершения, выдаваемым последней командой выполняемой последовательности.

Команда **test** анализирует переданное ей в качестве параметра выражение, и если оно *истинно*, то команда завершается с кодом 0. Если же выражение *ложно*, то команда завершается с кодом 1.

Команда **test** может выполнять проверку нескольких типов простых условий:

* проверки, связанные с файлами;
* проверки, связанные со строками;
* арифметические сравнения.

1. Для чего предназначены команды for, while и until языка Shell? В чем особенности циклов языка Shell по сравнению с другими языками программирования?

For

Оператор for в языке Shell несколько отличается от одноименных  
аналогов в алгоритмических языках программирования. Он перебирает  
значения из заданного списка и для каждого из значений выполняет  
некоторую последовательность команд. Поэтому аналогом данного  
125  
оператора в других языках является не оператор for, а оператор foreach.

While Until

Перед каждой итерацией цикла выполняется список\_команд1. Если  
код завершения последней команды в нем равен 0 (условие истинно), то  
выполняется тело цикла, представленное списком команд 2. Если код  
завершения последней команды в списке команд 1 не равен нулю, то цикл  
завершается.

Отличие оператора until от оператора while состоит в том, что  
список\_команд1 задает не условие продолжения цикла, а условие его  
окончания. Т.е. когда последняя выполненная команда в списке команд 1  
завершится с кодом 0, цикл завершится.  
Обратите внимание – цикл until проверяет условие передвыполнением очередной итерации, а не после ее завершения, как принято,  
например, в языке Pascal.

1. Как реализованы функции в языке Shell? Каким способом функциям передаются параметры?

Функции так же присутствуют в языке Shell, правда, обладая несколько  
урезанной функциональностью по сравнению с другими языками  
программирования. Внешне функции в языке Shell напоминают  
определенные пользователем команды.

Важным моментом является передача параметров в функцию и  
возврат из нее значения.  
В большинстве языков программирования требуется описывать имена  
и типы параметров функции заранее. В языке Shell нет типов данных,  
поэтому параметры функции описывать не нужно. Более того, мы можем  
передать в функцию различное количество параметров.