Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра автоматизированные системы управления

Лабораторная работа №4

по Linux-server

«Создание и использование сценариев, скриптов в Linux»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент |  |  |  | Кретов И.О. |
| Группа С-АС-22-1 |  |  |  |  |
| Руководитель |  |  |  | Кургасов В.В. |

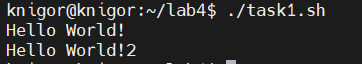
Липецк 2023 г.

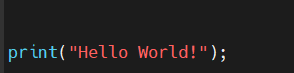
# Цель работы:

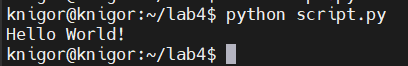
Изучить основные возможности языка программирования высокого уровня Shell. Получить навыки написания и использования скриптов.

1. Используя команды echo, printf, вывести информацию на экран.

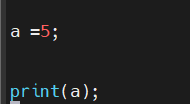






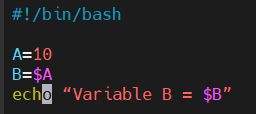


1. Присвоить переменной A целочисленное значение. Посмотреть значение переменной A.





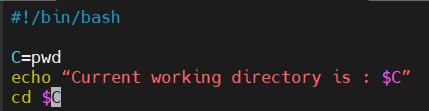
1. Присвоить переменной B значение переменной A.

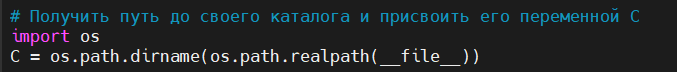






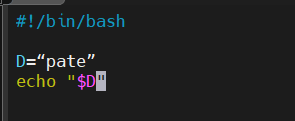
1. Присвоить переменной C значение “путь до своего каталога”, перейти в этот каталог с использованием переменной







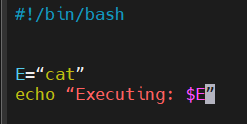
1. Присвоить переменной D значение “имя команды”, а именно команды PATE. Выполнить команду, используя значение переменной

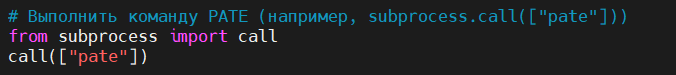






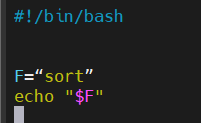
1. Присвоить переменной E значение “имя команды”, а именно команды просмотра содержимого файла, посмотреть содержимое переменной. Выполнить эту команду, используя значение переменной





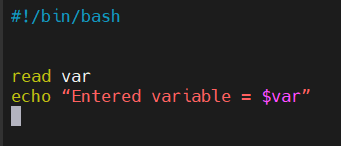


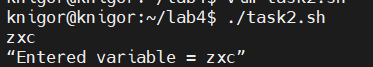
1. Присвоить переменной *F* значение “имя команды”, а именно, сортировки содержимого текстового файла. Выполнить эту команду, используя значение переменной.



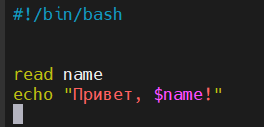


1. Программа запрашивает значение переменной, а затем выводит значение этой переменной.



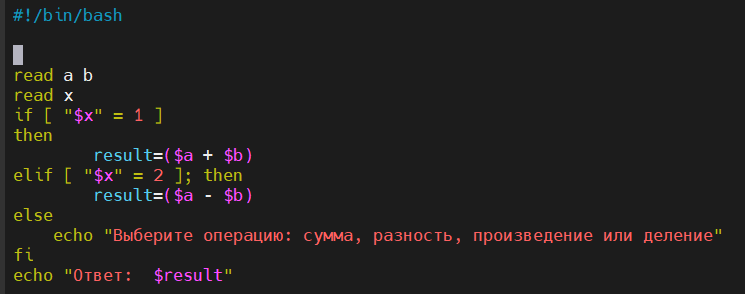


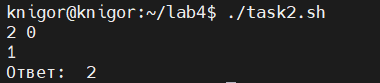
1. Программа запрашивает имя пользователя, затем здоровается с ним, используя значение введенной переменной.



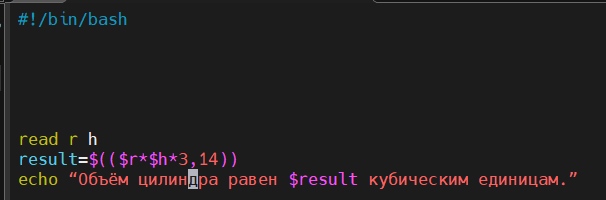


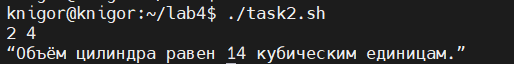
1. Программа запрашивает значения двух переменных, вычисляет сумму (разность, произведение, деление) этих переменных. Результат выводится на экран (использовать команды: a) EXPR; б) ВС).



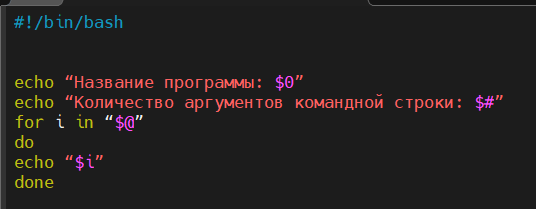


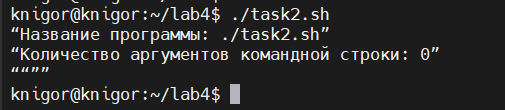
1. Вычислить объем цилиндра. Исходные данные запрашиваются программой. Результат выводится на экран.



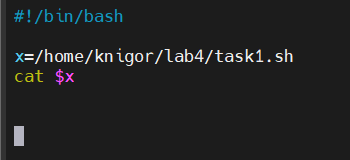


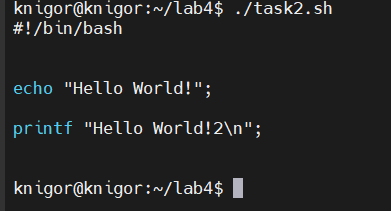
1. Используя позиционные параметры, отобразить имя программы, количество аргументов командной строки, значение каждого аргумента командной строки.



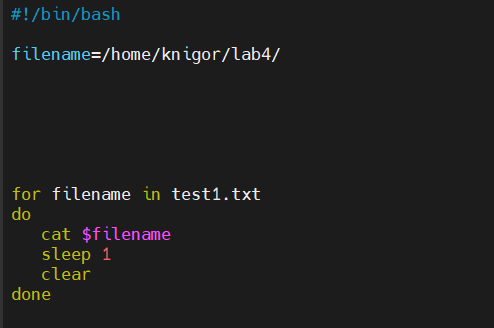


1. Используя позиционный параметр, отобразить содержимое текстового файла, указанного в качестве аргумента командной строки. После паузы экран очищается.



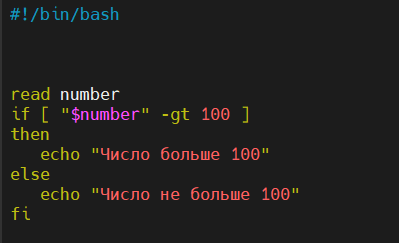


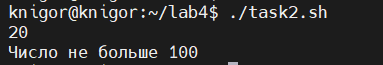
1. Используя оператор FOR, отобразить содержимое текстовых файлов текущего каталога поэкранно.





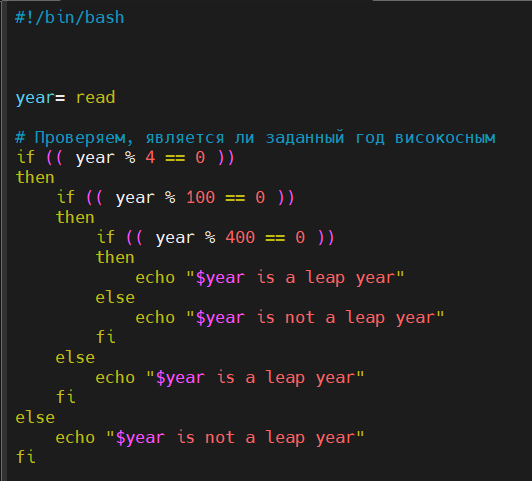
1. Программой запрашивается ввод числа, значение которого затем сравнивается с допустимым значением. В результате этого сравнения на экран выдаются соответствующие сообщения.

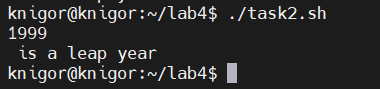




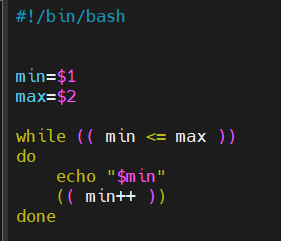
1. Программой запрашивается год, определяется, високосный ли он.

Результат выдается на экран.



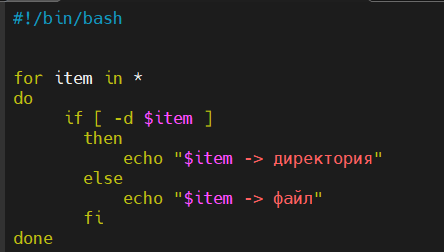


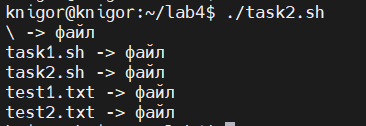
1. Вводятся целочисленные значения двух переменных. Вводится диапазон данных. Пока значения переменных находятся в указанном диапазоне, их значения инкрементируются.



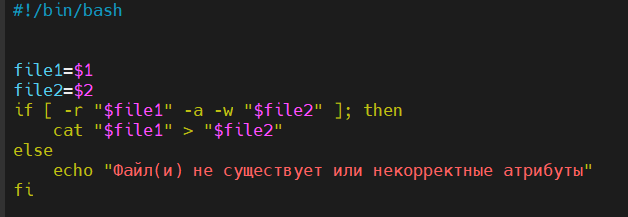


1. В качестве аргумента командной строки указывается пароль. Если пароль введен верно, постранично отображается в длинном формате с указанием скрытых файлов содержимое каталога /etc.
2. Проверить, существует ли файл. Если да, выводится на экран его содержимое, если нет - выдается соответствующее сообщение
3. Если файл есть каталог и этот каталог можно читать, просматривается содержимое этого каталога. Если каталог отсутствует, он создается. Если файл не есть каталог, просматривается содержимое файла



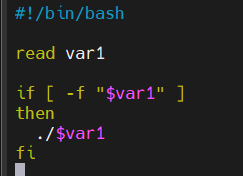


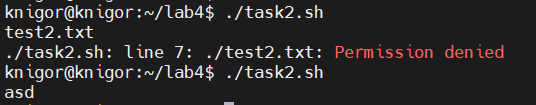
1. Анализируются атрибуты файла. Если первый файл существует и используется для чтения, а второй файл существует и используется для записи, то содержимое первого файла перенаправляется во второй файл. В случае несовпадений указанных атрибутов или отсутствия файлов на экран выдаются соответствующие сообщения (использовать имена файлов и/или позиционные параметры)



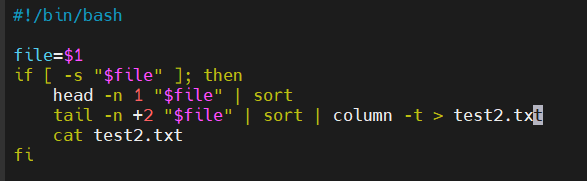


1. Если файл запуска программы найден, программа запускается (по выбору)





1. В качестве позиционного параметра задается файл, анализируется его размер. Если размер файла больше нуля, содержимое файла сортируется по первому столбцу по возрастанию, отсортированная информация помещается в другой файл, содержимое которого затем отображается на экране.





Листинг для задач на языке Python

# Присвоить значение A переменной a

a = 10

# Присвоить переменную B значение переменной A (B = A)

b = a

# Получить путь до своего каталога и присвоить его переменной C

import os

C = os.path.dirname(os.path.realpath(\_\_file\_\_))

# Перейти в каталог, указанный в переменной C (используя os.chdir)

os.chdir(C)

# Выполнить команду PATE (например, subprocess.call(["pate"]))

from subprocess import call

call(["pate"])

# Просмотреть содержимое файла (например, print(open("file.txt").read()))

def view\_file\_content(file\_name):

with open(file\_name, 'r') as file:

print(file.read())

# Просмотреть содержимое файла "file.txt" (view\_file\_content("file.txt"))

view\_file\_content("file.txt")

# Сортировать содержимое текстового файла (например, sorted(file.readlines()))

def sort\_text\_file(file\_name):

with open(file\_name) as file:

return sorted(file.readlines())

# Отсортировать содержимое текстового файл “file.txt” и вывести результат на экран (print(sort\_text\_file(“file.txt”)))

print(sort\_text\_file(“file.txt”))

# Отсортировать содержимое текстового файл “file.txt” и вывести результат на экран (print(sort\_text\_file(“file.txt”)))

print(sort\_text\_file(“file.txt”))

#Запросить значение переменной и вывести его на экран

def get\_and\_print\_var(var\_name):

try:

var\_value = eval(var\_name)

except NameError:

var\_value = “Variable does not exist”

finally:

print("Variable " + var\_name + " has value " + str(var\_value))

get\_and\_print\_var(“a”)

#Запрос имени пользователя и приветствие

def greet\_user(user\_name):

print("Hello, " + user\_name + “!”)

greet\_user(input("Enter your name: "))

#Расчет суммы, разности, произведения, деления двух переменных и вывод результата

def calculate\_with\_vars(operation, var\_a, var\_b):

var\_a = float(eval(var\_a))

var\_b = float(eval(var\_b))

if operation == “sum”:

result = var\_a + var\_b

elif operation == “diff”:

result = var\_a - var\_b

elif operation == “prod”:

result = var\_a \* var\_b

elif operation == “div”:

if var\_b == 0:

print(“Division by zero is not allowed.”)

else:

result = var\_a / var\_b

var\_a = input("Enter variable a: ")

var\_b = input("Enter variable b: ")

operation = input("Select operation (+, -, \*, /): ")

result = calculate\_with\_vars(operation, var\_a, var\_b)

print("Result is: " + str(result))