Оглавление

[Введение 2](#_Toc122457225)

[Первый вариант сравнения 3](#_Toc122457226)

[1.Ввод имен файлов и их проверка 3](#_Toc122457227)

[2.Создание последовательностей целых чисел для каждого из файлов 3](#_Toc122457228)

[3.Вывод результатов 4](#_Toc122457229)

[4.Тесты 4](#_Toc122457230)

[Второй вариант сравнения 4](#_Toc122457231)

[1.Ввод имен файлов и их проверка 4](#_Toc122457232)

[2.Нахождение подстроки и считывание текста 5](#_Toc122457233)

[3.Результат 6](#_Toc122457234)

[4.Тесты 6](#_Toc122457235)

[Третий вариант сравнения 6](#_Toc122457236)

[1.Ввод имен файлов и их проверка 6](#_Toc122457237)

[2.Создание последовательностей целых чисел для каждого из файлов 7](#_Toc122457238)

[3.Вывод результатов 7](#_Toc122457239)

[4.Тесты 7](#_Toc122457240)

[Четвертый вариант сравнения 8](#_Toc122457241)

[1.Ввод имен файлов и их проверка 8](#_Toc122457242)

[2.Создание последовательностей целых чисел для каждого из файлов 8](#_Toc122457243)

[3.Вывод результатов 8](#_Toc122457244)

[4.Тесты 9](#_Toc122457245)

[Заключение 10](#_Toc122457246)

# Введение

Цель задания: разработать и протестировать скрипт командной оболочки для сравнения содержимого двух текстовых файлов по определённым правилам.

1. Сравнивание последовательностей целых чисел в файлах.
2. Сравнивание текста в файлах после первого вхождения подстроки «string:».
3. Сравнивание последовательностей чисел с плавающей точкой, записанных не в экспоненциальной форме, в файлах.
4. Сравнивание последовательностей чисел с плавающей точкой, в том числе записанных в экспоненциальной форме, в файлах.

# Первый вариант сравнения

Необходимо найти все целые числа в каждом из файлов (каждое отделено пробельным символом), а затем сравнить полученные последовательности.

## 1.Ввод имен файлов и их проверка

#!/bin/bash

file1=$1

file2=$2

verb=$3

**if** [ -z "$file1" ] || [ -z "$file2" ] || [ **!** -f "$file1" ] || [ **!** -f "$file2" ]; **then**

**if** [[ $verb == "-v" ]]; **then**

**echo** "Please enter correct file names"

**fi**

**exit** 2

**elif** [ **!** -r "$file1" ] || [ **!** -r "$file2" ]; **then**

**if** [[ $verb == "-v" ]]; **then**

**echo** "Permission error"

**fi**

**exit** 3

**fi**

Значения позиционных параметров с именами сравниваемых файлов присваиваются переменным file1 и file2, а затем проверяется, были ли введены

имена, существуют ли данные файлы и предоставлен ли к ним доступ.

## 2.Создание последовательностей целых чисел для каждого из файлов

nums1=""

nums2=""

**while** **read** -r line1; **do**

line1=${line1//"."/"a"}

tmp="$(grep -owE "[-]?[0-9]+" <<< "$line1")"

**if** [[ -n $tmp ]]; **then**

nums1="$nums1

$tmp"

**fi**

**done** < "$file1"

**while** **read** -r line2; **do**

line2=${line2//"."/"a"}

tmp="$(grep -owE "[-]?[0-9]+" <<< "$line2")"

**if** [[ -n $tmp ]]; **then**

nums2="$nums2

$tmp"

**fi**

**done** < "$file2"

Построчное чтение файлов и запись нужных чисел.

## 3.Вывод результатов

Если последовательности равны, код возврата нулевой, иначе ненулевой.

**if** [[ "$nums1" == "$nums2" ]]; **then**

**if** [[ $verb == "-v" ]]; **then**

**echo** "files match"

**fi**

**exit** 0

**else**

**if** [[ $verb == "-v" ]]; **then**

**echo** "files do not match"

**fi**

**exit** 1

**fi**

## 4.Тесты

|  |  |
| --- | --- |
| **Тест** | **Результат** |
| Два одинаковых файла | Пройден |
| Два файла с одинаковыми числовыми последовательностями | Пройден |
| Последовательности одинаковы, некоторые числа во втором файле находятся в начале строк. | Пройден |
| Последовательности одинаковы, некоторые числа во втором файле находятся в конце строк. | Пройден |
| Некоторые числа не отделены пробелами – последовательности не одинаковы | Пройден |
| Разные последовательности. | Пройден |
| Второй файл пуст | Пройден |
| Во втором файле нет чисел | Пройден |

# Второй вариант сравнения

Необходимо найти первое вхождение подстроки «string:» в каждом из файлов, затем сравнить отрывки текста после этих вхождений.

## 1.Ввод имен файлов и их проверка

#!/bin/bash

file1=$1

file2=$2

verb=$3

**if** [ -z "$file1" ] || [ -z "$file2" ] || [ **!** -f "$file1" ] || [ **!** -f "$file2" ]; **then**

**if** [[ $verb == "-v" ]]; **then**

**echo** "Please enter correct file names"

**fi**

**exit** 2

**elif** [ **!** -r "$file1" ] || [ **!** -r "$file2" ]; **then**

**if** [[ $verb == "-v" ]]; **then**

**echo** "Permission error"

**fi**

**exit** 3

**fi**

Значения позиционных параметров с именами сравниваемых файлов присваиваются переменным file1 и file2, а затем проверяется, были ли введены

имена, существуют ли данные файлы и предоставлен ли к ним доступ.

## 2.Нахождение подстроки и считывание текста

Сначала построчно перебирается первый файл. Как только встречается строка, содержащая нужную подстроку, начинается перебор второго файла. Если подстрока была найдена и во втором файле, сравниваются фрагменты строк, содержащих подстроку, после нее. Если они равны, начинается запись всех остальных строк в соответствующие переменные.

arr1=""

arr2=""

delimiter="string:"

weregood=0

**while** **read** -r line1; **do**

**if** [ $weregood -eq 1 ]; **then**

arr1="$arr1

$line1"

**elif** [[ $line1 == \*"string:"\* ]]; **then**

arr1=${line1#\*$delimiter}

**while** **read** -r line2; **do**

**if** [ $weregood -eq 1 ]; **then**

arr2="$arr2

$line2"

**elif** [[ $line2 == \*"string:"\* ]]; **then**

#arr2=$(grep -Eo "string:[[:graph:]]\*" <<< "$line2")

arr2=${line2#\*$delimiter}

**if** [[ "$arr1" != "$arr2" ]]; **then**

**if** [ "$verb" == "-v" ]; **then**

**echo** "files do not match"

**fi**

**exit** 1

**else**

arr1=""

arr2=""

weregood=1

**fi**

**fi**

**done** < "$file2"

**fi**

**done** < "$file1"

## 3.Результат

**if** [[ "$arr1" == "$arr2" ]]; **then**

**if** [ "$verb" == "-v" ]; **then**

**echo** "files match"

**fi**

**exit** 0

**else**

**if** [ "$verb" == "-v" ]; **then**

**echo** "files do not match"

**fi**

**exit** 1

**fi**

Если отрывки равны, код возврата нулевой, иначе ненулевой.

## 4.Тесты

|  |  |
| --- | --- |
| **Тест** | **Результат** |
| Два одинаковых файла | Пройден |
| Одинаковые отрывки | Пройден |
| Разные отрывки | Пройден |
| Проверка, берет ли программа первое вхождение «string:» | Пройден |
| Во втором файле отсутствует «string:» | Пройден |

# Третий вариант сравнения

Необходимо найти все ЧПТ, записанные не в экспоненциальной форме, в каждом из файлов (каждое отделено пробельным символом), а затем сравнить полученные последовательности.

## 1.Ввод имен файлов и их проверка

#!/bin/bash

file1=$1

file2=$2

verb=$3

**if** [ -z "$file1" ] || [ -z "$file2" ] || [ **!** -f "$file1" ] || [ **!** -f "$file2" ]; **then**

**if** [[ $verb == "-v" ]]; **then**

**echo** "Please enter correct file names"

**fi**

**exit** 2

**elif** [ **!** -r "$file1" ] || [ **!** -r "$file2" ]; **then**

**if** [[ $verb == "-v" ]]; **then**

**echo** "Permission error"

**fi**

**exit** 3

**fi**

Значения позиционных параметров с именами сравниваемых файлов присваиваются переменным file1 и file2, а затем проверяется, были ли введены

имена, существуют ли данные файлы и предоставлен ли к ним доступ.

## 2.Создание последовательностей целых чисел для каждого из файлов

nums1=$(grep -owE "[-]?[0-9]+(\.[0-9]+)?" "$file1")

nums2=$(grep -owE "[-]?[0-9]+(\.[0-9]+)?" "$file2")

Запись последовательностей в переменные с помощью регулярных выражений.

## 3.Вывод результатов

**if** [[ "$nums1" == "$nums2" ]]; **then**

**if** [[ $verb == "-v" ]]; **then**

**echo** "files match"

**fi**

**exit** 0

**else**

**if** [[ $verb == "-v" ]]; **then**

**echo** "files do not match"

**fi**

**exit** 1

**fi**

Если последовательности равны, код возврата нулевой, иначе ненулевой.

## 4.Тесты

|  |  |
| --- | --- |
| **Тест** | **Результат** |
| Два одинаковых файла | Пройден |
| Два файла с одинаковыми числовыми последовательностями | Пройден |
| Последовательности одинаковы, некоторые числа во втором файле находятся в начале строк. | Пройден |
| Последовательности одинаковы, некоторые числа во втором файле находятся в конце строк. | Пройден |
| Разные последовательности. | Пройден |
| Второй файл пуст | Пройден |
| Во втором файле нет чисел | Пройден |

# Четвертый вариант сравнения

Необходимо найти все ЧПТ, записанные в том числе в экспоненциальной форме, в каждом из файлов (каждое отделено пробельным символом), а затем сравнить полученные последовательности.

## 1.Ввод имен файлов и их проверка

#!/bin/bash

file1=$1

file2=$2

verb=$3

**if** [ -z "$file1" ] || [ -z "$file2" ] || [ **!** -f "$file1" ] || [ **!** -f "$file2" ]; **then**

**if** [[ $verb == "-v" ]]; **then**

**echo** "Please enter correct file names"

**fi**

**exit** 2

**elif** [ **!** -r "$file1" ] || [ **!** -r "$file2" ]; **then**

**if** [[ $verb == "-v" ]]; **then**

**echo** "Permission error"

**fi**

**exit** 3

**fi**

Значения позиционных параметров с именами сравниваемых файлов присваиваются переменным file1 и file2, а затем проверяется, были ли введены

имена и существуют ли данные файлы.

## 2.Создание последовательностей целых чисел для каждого из файлов

nums1=$(grep -owE "[-]?[0-9]+(\.[0-9]+)?([eE][+-]?[0-9]+)?" "$file1")

nums2=$(grep -owE "[-]?[0-9]+(\.[0-9]+)?([eE][+-]?[0-9]+)?" "$file2")

Запись последовательностей в переменные с помощью регулярных выражений.

## 3.Вывод результатов

**if** [[ "$nums1" == "$nums2" ]]; **then**

**if** [[ $verb == "-v" ]]; **then**

**echo** "files match"

**fi**

**exit** 0

**else**

**if** [[ $verb == "-v" ]]; **then**

**echo** "files do not match"

**fi**

**exit** 1

**fi**

Если последовательности равны, код возврата нулевой, иначе ненулевой.

## 4.Тесты

|  |  |
| --- | --- |
| **Тест** | **Результат** |
| Два одинаковых файла | Пройден |
| Два файла с одинаковыми числовыми последовательностями | Пройден |
| Последовательности одинаковы, некоторые числа во втором файле находятся в начале строк. | Пройден |
| Последовательности одинаковы, некоторые числа во втором файле находятся в конце строк. | Пройден |
| Разные последовательности. | Пройден |
| Второй файл пуст | Пройден |
| Во втором файле нет чисел | Пройден |

# Заключение

Реализован и протестирован скрипт, сравнивающий два файла по четырем вариантам сравнения:

1. Сравнивание последовательностей целых чисел в файлах.
2. Сравнивание текста в файлах после первого вхождения подстроки «string:».
3. Сравнивание последовательностей чисел с плавающей точкой, записанных не в экспоненциальной форме, в файлах.
4. Сравнивание последовательностей чисел с плавающей точкой, в том числе записанных в экспоненциальной форме, в файлах.