Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ»

КАФЕДРА «ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ»

ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ

Студент **Поляков Андрей Игоревич**

Группа **ИУ7-12Б**

Тип практики **Проектно-технологическая практика**

Название предприятия **НУК ИУ МГТУ им. Н. Э. Баумана**

Студент _____ **Поляков А И**

Руководитель практики _____ **Ломовской И. В.**

Руководитель практики _____ **Кострицкий А. С.**

Оценка _____

2022 г.

Оглавление

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 в последовательность.

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
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
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
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.

Источники

- Электронная образовательная система Кафедры "Программное обеспечение ЭВМ и информационные технологии
- Регулярные выражения (regexp) – основы (habr)
- Работа со строками в bash – webhamster - https://webhamster.ru/mytetrashare/index/mtb0/1516
- Перенаправление ввода-вывода (younglinux)
- Что такое grep и с чем его едят (habr) - https://habr.com/p/229501/