



**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 Смирнов Иван Владимирович

Группа ИУ7-12Б

Тип практики Проектно-технологическая практика

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

Студент _____ Смирнов И В

Руководитель практики _____ Ломовской И. В.

Руководитель практики _____ Кострицкий А. С.

Оценка _____

2022 г.

Оглавление

1. Введение	3
2. Сценарий командной оболочки	4
3. Заключение	8
4. Список использованных источников	9

1. Введение

Задание: разработать скрипт командной оболочки для сравнения содержимого двух текстовых файлов по определенным правилам (как сказано в варианте).

Вариант 1

Сравниваются последовательности целых чисел в файлах. Каждое целое число заведомо входит в диапазон знакового целого в 4 байта. Целые числа отделяются от других символов в файле пробельными символами. Обращаем Ваше внимание, что в файле могут находиться не только числа, но вычленять и анализировать нужно именно их. Сравниваются числа как текст — числа 100 и 0100 следует считать различными.

Название скрипта: `comparator1.sh`.

Для решения задания предлагается решить ряд подзадач, что позволит составить цельное представление о задании.

- 1) Написать проверку файлов на их существование.
- 2) Написать функцию, которая проверяет, является ли переданная ей строка числом, так как в передаваемом файле хранятся не только числа.
- 3) Ввести переменные, которые определяют, какое число (по номеру и по значению) из одного и из другого файла было взято.
- 4) Считывать файлы по словам, найти среди них числа.
- 5) Сравнить переданные числа (их номер в файле, их значения, а также длина должны быть равны).
- 6) Результат выполнения программы вернуть в виде кода возврата.

Вариант 2

Сравнивается текст в файлах после первого вхождения подстроки «string:». Подразумевается строгое сравнение с учётом разницы в пробельных символах и символах окончания строки

Название скрипта: `comparator2.sh`.

Для решения задания предлагается решить ряд подзадач, что позволит составить цельное представление о задании.

- 1) Написать проверку файлов на их существование.
- 2) Считывать файлы построчно. Найти первое вхождение подстроки с помощью соответствующего регулярного выражения

- 3) Сравнить оставшуюся часть строк с первым вхождением подстроки и следующие строки файлов.
- 4) Результат выполнения программы вернуть в виде кода возврата.

Вариант 3

Сравниваются последовательности чисел с плавающей точкой (ЧПТ), записанных **не** в экспоненциальной форме. Каждое найденное ЧПТ заведомо входит в диапазон чисел двойной точности. Обращаем Ваше внимание, что в файле могут находиться не только числа, но вычленять и анализировать нужно именно их. Сравниваются числа как текст — например, числа 1.0 и 1.00 следует считать различными.

Название скрипта: comparator3.sh.

Для решения задания предлагается решить ряд подзадач, что позволит составить цельное представление о задании.

- 1) Написать проверку файлов на их существование.
- 2) Написать функцию, которая проверяет, является ли переданная ей строка ЧПТ, так как в передаваемом файле хранятся не только числа.
- 3) Ввести переменные, которые определяют, какое число (по номеру и по значению) из одного и из другого файла было взято.
- 4) Считывать файлы по словам, найти среди них числа.
- 5) Сравнить переданные числа (их номер в файле, их значения, а также длина должны быть равны).
- 6) Результат выполнения программы вернуть в виде кода возврата.

2. Сценарий командной оболочки

Вариант 1

```
#!/bin/bash

# Функция для проверки на число
function num
{
    if [[ $1 =~ ^[+-]?[0-9]+$ ]]; then
        echo 1
    else
        echo 0
    fi
}
```

```

f1=$1
f2=$2
if [ ! -f $f1 ]; then
    echo "$f1 (файл не существует)."
    exit 1
elif [ ! -f $f2 ]; then
    echo "$f2 (файл не существует)."
    exit 1
fi
f1_num=0
f2_num=0
f1_num_count=0
f2_num_count=0

# Сравнение чисел из обоих файлов
for word1 in $(cat $f1); do
    if [[ "$( num $word1 )" -eq 1 ]]; then
        f1_num=$((f1_num + 1))
        f1_num_count=$((f1_num_count + 1))
        f2_num=0
        f2_num_count=0
        for word2 in $(cat $f2); do
            if [[ "$( num $word2 )" -eq 1 ]]; then
                f2_num_count=$((f2_num_count + 1))
                f2_num=$((f2_num + 1))
                if [[ f2_num -eq f1_num ]]; then
                    if [[ word1 -ne word2 ]] || [[ ${#word1} != ${#word2} ]]; then
                        echo "Файлы не совпадают."
                        exit 1
                    fi
                fi
            fi
        done
    fi
done

# Проверка второго файла на наличие чисел
if [ $f1_num_count == 0 -a $f2_num_count == 0 ]; then
    for word2 in $(cat $f2); do
        if [[ "$( num $word2 )" -eq 1 ]]; then
            f2_num_count=$((f2_num_count + 1))
        fi
    done
fi

if [[ f1_num_count -ne f2_num_count ]]; then
    echo "Файлы не совпадают."
    exit 1
else

```

```
    echo "Файлы совпадают."
    exit 0
fi
```

Вариант 2

```
#!/bin/bash

lines_found=0
f1=$1
f2=$2
if [ ! -f $f1 ]; then
    echo "$f1 (файл не существует)."
    exit 1
elif [ ! -f $f2 ]; then
    echo "$f2 (файл не существует)."
    exit 1
fi
string=$3
l1_num=0
l2_num=0
r=0
while read word1; do
    l1_num=$((l1_num+1))
    if [[ $word1 =~ .*$string.* ]] || [ $lines_found == 1 ]; then
        l2_num=0
        while read word2; do
            l2_num=$((l2_num+1))
            if [[ $word2 =~ .*$string.* ]] && [ $lines_found == 0 ]; then
                r=$((l1_num - l2_num))
                lines_found=1
                sent1_end=$(echo "$word1" | sed "s/.*${string}/"/")
                sent2_end=$(echo "$word2" | sed "s/.*${string}/"/")
                if [[ ! $sent1_end == $sent2_end ]]; then
                    echo "Файлы не совпадают"
                    exit 1
                fi
            elif [ $lines_found == 1 ] && [[ $((l1_num - l2_num)) -eq $r ]]; then
                if [[ ! $word1 == $word2 ]]; then
                    echo "Файлы не совпадают"
                    exit 1
                fi
            elif [[ $((l1_num - l2_num)) -lt $r ]]; then
                break
            fi
        done < $f2
        if [[ $l2_num == 0 ]]; then
            echo "Файлы не совпадают"
            exit 1
        fi
    fi
done < $f1
```

```

        fi
    fi
done < $f1

if [[ $l1_num == 0 ]]; then
    while read word2; do
        if [[ $word2 =~ .*$string.* ]]; then
            echo "Файлы не совпадают"
            exit 1
        fi
    done < $f2
fi
echo "Файлы совпадают"
exit 0

```

Вариант 3

```

#!/bin/bash

# Функция для проверки на число двойной точности
function num
{
    if [[ $1 =~ ^[+-]?[0-9]+([.][0-9]+)?$ ]]; then
        echo 1
    else
        echo 0
    fi
}

f1=$1
f2=$2

if [ ! -f $f1 ]; then
    echo "$f1 (файл не существует)."
    exit 1
elif [ ! -f $f2 ]; then
    echo "$f2 (файл не существует)."
    exit 1
fi

f1_num=0
f2_num=0
f1_num_count=0
f2_num_count=0

# Сравнение чисел из обоих файлов
for word1 in $(cat $f1); do
    if [[ "$( num $word1 )" -eq 1 ]]; then
        f1_num=$((f1_num + 1))
        f1_num_count=$((f1_num_count + 1))
        f2_num=0
    fi
done

```

```

f2_num_count=0
for word2 in $(cat $f2); do
    if [[ "$( num $word2 )" -eq 1 ]]; then
        f2_num_count=$((f2_num_count + 1))
        f2_num=$((f2_num + 1))
        if [[ f2_num -eq f1_num ]]; then
            if [[ ${#word1} != ${#word2} ]] || [[ ! $word1 == $word2 ]]; then
                echo "Файлы не совпадают."
                exit 1
            fi
        fi
    fi
done

# Проверка второго файла на наличие чисел
if [ $f1_num_count == 0 -a $f2_num_count == 0 ]; then
    for word2 in $(cat $f2); do
        if [[ "$( num $word2 )" -eq 1 ]]; then
            f2_num_count=$((f2_num_count + 1))
        fi
    done
fi

if [[ f1_num_count -ne f2_num_count ]]; then
    echo "Файлы не совпадают."
    exit 1
else
    echo "Файлы совпадают."
    exit 0
fi

```

3. Заключение

Вариант 1

- 1) В скрипте написана проверка файлов на их существование.
- 2) В скрипте написана функция проверки строки на число.
- 3) В скрипте введены переменные, которые определяют номер числа в файле
- 4) В скрипте реализовано считывание файлов по словам, а также нахождение чисел среди них.
- 5) В скрипте написан алгоритм сравнения переданных чисел (их номер в файле, их значения, а также длина сравниваются).
- 6) Результат выполнения программы получен в виде кода возврата.

Вариант 2

- 1) В скрипте написана проверка файлов на их существование.
- 2) В скрипте реализовано считывание файлов построчно. Был написан способ нахождения первого вхождения подстроки с помощью соответствующего регулярного выражения
- 3) В скрипте реализовано сравнение оставшихся частей строк с первым вхождением подстроки, а также сравнение следующих строк файлов.
- 4) Результат выполнения программы получен в виде кода возврата.

Вариант 3

- 1) В скрипте написана проверка файлов на их существование.
- 2) В скрипте написана функция проверки строки на ЧПТ.
- 3) В скрипте введены переменные, которые определяют номер числа в файле
- 4) В скрипте реализовано считывание файлов по словам, а также нахождение чисел среди них.
- 5) В скрипте написан алгоритм сравнения переданных чисел (их номер в файле, их значения, а также длина сравниваются).
- 6) Результат выполнения программы получен в виде кода возврата.

В данном отчёте все поставленные задачи и подзадачи были выполнены.

4. Список использованных источников

1. Курс “Проектно-технологическая практика (знакомство с Linux)”:
<https://e-learning.bmstu.ru/iu7/course/view.php?id=76>
2. Тестировщик регулярных выражений:
<https://regex101.com/>
3. Тестировщик команды Sed:
<https://sed.js.org/index.html>
4. Статья на тему “Команда Sed для Linux/Unix с примерами”:
<https://www.geeksforgeeks.org/sed-command-in-linux-unix-with-examples/>
5. Статья на тему “if в Bash: от новичка до профессионала”:
<https://www.assertnotmagic.com/2019/01/16/bash-if-statements/>