



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 **Паламарчук Андрей Николаевич**

Группа **ИУ7-13Б**

Тип практики **Проектно-технологическая практика**

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

Студент _____ **Паламарчук А Н**

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

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

Оценка _____

2022 г.

Оглавление

Введение.....	3
Сценарий командной оболочки	4
Тестирование сценариев командной оболочки.....	11
Заключение	12
Список использованных источников	13

Введение

Цель работы - создание скриптов командной оболочки для сравнения содержимого двух текстовых файлов по вариантам приведенным ниже.

Варианты сравнения:

1. Сравняются 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. Сравняются числа как текст — числа 100 и 0100 следует считать различными. Название скрипта: `comparator1.sh`.
2. Сравняется 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`.
3. Сравняются 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. Сравняются числа как текст — например, числа 1.0 и 1.00 следует считать различными. Название скрипта: `comparator3.sh`.
4. Сравняются последовательности чисел с плавающей точкой (ЧПТ), записанных в том числе 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. Сравняются числа как текст — например, числа 1.0 и 1.00 следует считать различными, числа 1.0 и 1.0e0 – тоже.

Сценарий командной оболочки

Вариант 1:

```
#!/bin/bash
# Проверка корректности количества переданных аргументов
[[ "$#" = 2 || "$#" = 3 ]] || exit 1
# Проверка существования обоих файлов
[[ -f "$1" && -f "$2" ]] || exit 1
# Проверка возможности чтения обоих файлов
[[ -r "$1" && -r "$2" ]] || exit 1
# Корректность ключа verbose
[[ "$#" = 3 && "$3" != "-v" ]] && exit 1
[[ "$#" = 3 && "$3" = "-v" ]] && echo "Дополнительная информация: - "
# Собираем числа из первого файла в первый массив
arrres1=()
i1=0
while read text1; do
    for cell1 in $text1; do
        if [[ "$cell1" =~ ^[+-]?[0-9]+$ ]]; then
            arrres1[$i1]="${cell1}"
            i1=$((i1+1))
        fi
    done
done < $1
# Собираем числа из второго файла во второй массив
arrres2=()
i2=0
while read text2; do
    for cell2 in $text2; do
        if [[ "$cell2" =~ ^[+-]?[0-9]+$ ]]; then
            arrres2[$i2]="${cell2}"
            i2=$((i2+1))
        fi
    done
done < $2
# Проверяем длины массивов
if [[ "${#arrres1[@]}" = 0 && "${#arrres2[@]}" = 0 ]]; then
    echo "В обоих текстах нет ни одного числа"
    exit 1
else
    if [[ "${#arrres1[@]}" = 0 ]]; then
        echo "В первом тексте нет ни одного числа"
        exit 1
    elif [[ "${#arrres2[@]}" = 0 ]]; then
        echo "Во втором тексте нет ни одного числа"
        exit 1
    fi
fi
# Длины массивов корректны, проверяем циклом совпадение элементов
```

```

if [[ ${#arrres1[@]} != ${#arrres2[@]} ]]; then
    echo "Последовательности чисел в файлах $1 и $2 не совпадают"
    exit 1
else
    for i in ${!arrres1[@]}; do
        if [[ "$arrres1[$i]" != "$arrres2[$i]" ]]; then
            echo "Последовательности чисел в файлах $1 и $2 не
совпадают"
            exit 1
        fi
    done
fi
echo "Последовательности чисел в файлах $1 и $2 совпадают"
exit 0

```

Вариант 2:

```

#!/bin/bash
# Проверка корректности количества переданных аргументов
[[ "$#" = 2 || "$#" = 3 ]] || exit 1
# Проверка существования обоих файлов
[[ -f "$1" && -f "$2" ]] || exit 1
# Проверка возможности чтения обоих файлов
[[ -r "$1" && -r "$2" ]] || exit 1
# Корректность ключа verbose
[[ "$#" = 3 && "$3" != "-v" ]] && exit 1
[[ "$#" = 3 && "$3" = "-v" ]] && echo "Дополнительная информация: - "
# Поиск маркера начала сравнения
coincidence1=0
while read text1; do
    for cell1 in $text1; do
        if [[ "$cell1" =~ "string:" ]]; then
            # Флаг существования маркера в первом тексте
            coincidence1=1
        fi
    done
done < $1
coincidence2=0
while read text2; do
    for cell2 in $text2; do
        if [[ "$cell2" =~ "string:" ]]; then
            # Флаг существования маркера во втором тексте
            coincidence2=1
        fi
    done
done < $2
# Проверка наличия маркера начала сравнения
if [[ "$coincidence1" != 1 && "$coincidence2" != 1 ]]; then
    echo "В обоих текстах нет маркера начала сравнения (string:)"

```

```

        exit 1
else
    if [[ "$coincidence1" != 1 ]]; then
        echo "В первом тексте нет маркера начала сравнения
(string:)"
        exit 1
    elif [[ "$coincidence2" != 1 ]]; then
        echo "Во втором тексте нет маркера начала сравнения
(string:)"
        exit 1
    fi
fi
# Сохраняем старый разделитель и изменяем его
ifs="$IFS"
IFS=" "
# Собираем из первого текста всё, что после маркера в первый массив
arrstr1=()
# Счётчик для индекса первого массива
i1=0
# Флаг для первой строки
first_str1=1
while read text1; do
    if [[ "$first_str1" = 1 ]]; then
        if [[ "$text1" =~ "string:" ]]; then
            arrstr1[$i1]={"$(echo "$(echo "$text1" | grep -
Eo "string:.*")")"}
            i1=$((i1+1))
            first_str1=0
        fi
    elif [[ "$first_str1" = 0 ]]; then
        arrstr1[$i1]={"$text1"}
        i1=$((i1+1))
    fi
done < $1
# Собираем из второго текста всё, что после маркера во второй массив
arrstr2=()
# Счётчик для индекса второго массива
i2=0
# Флаг для первой строки
first_str2=1
while read text2; do
    if [[ "$first_str2" = 1 ]]; then
        if [[ "$text2" =~ "string:" ]]; then
            arrstr2[$i2]={"$(echo "$(echo "$text2" | grep -
Eo "string:.*")")"}
            i2=$((i2+1))
            first_str2=0
        fi
    elif [[ "$first_str2" = 0 ]]; then
        arrstr2[$i2]={"$text2"}

```

```

        i2=$((i2+1))
    fi
done < $2
# Проверяем длины массивов
if [[ "${#arrstr1[@]}" != "${#arrstr2[@]}" ]]; then
    echo "Тексты в файлах $1 и $2 не совпадают"
    exit 1
else
    # Проверяем совпадение элементов массива
    for i in ${!arrstr1[@]}; do
        if [[ "${arrstr1[$i]}" != "${arrstr2[$i]}" ]]; then
            echo "Тексты в файлах $1 и $2 не совпадают"
            exit 1
        fi
    done
fi
echo "Тексты в файлах $1 и $2 совпадают"
IFS="$ifs"
exit 0

```

Вариант 3:

```

#!/bin/bash
# Проверка корректности количества переданных аргументов
[[ "$#" = 2 || "$#" = 3 ]] || exit 1
# Проверка существования обоих файлов
[[ -f "$1" && -f "$2" ]] || exit 1
# Проверка возможности чтения обоих файлов
[[ -r "$1" && -r "$2" ]] || exit 1
# Корректность ключа verbose
[[ "$#" = 3 && "$3" != "-v" ]] && exit 1
[[ "$#" = 3 && "$3" = "-v" ]] && echo "Дополнительная информация: - "
# Собираем числа из первого файла в первый массив
arrres1=()
i1=0
while read text1; do
    for cell1 in $text1; do
        if [[ "$cell1" =~ ^[+-]?[0-9]+$ ]]; then
            arrres1[i1]="${cell1}"
            i1=$((i1+1))
        elif [[ "$cell1" =~ ^[+-]?[0-9]+\.[0-9]+$ ]]; then
            arrres1[i1]="${cell1}"
            i1=$((i1+1))
        fi
    done
done < $1
# Собираем числа из второго файла во второй массив
arrres2=()
i2=0

```

```

while read text2; do
    for cell2 in $text2; do
        if [[ "$cell2" =~ ^[+-]?[0-9]+$ ]]; then
            arrres2[$i2]="$cell2"
            i2=$((i2+1))
        elif [[ "$cell2" =~ ^[+-]?[0-9]+\.[0-9]+$ ]]; then
            arrres2[$i2]="$cell2"
            i2=$((i2+1))
        fi
    done
done < $2
# Проверяем длины массивов
if [[ "${#arrres1[@]}" = 0 && "${#arrres2[@]}" = 0 ]]; then
    echo "В обоих текстах нет ни одного числа"
    exit 1
else
    if [[ "${#arrres1[@]}" = 0 ]]; then
        echo "В первом тексте нет ни одного числа"
        exit 1
    elif [[ "${#arrres2[@]}" = 0 ]]; then
        echo "Во втором тексте нет ни одного числа"
        exit 1
    fi
fi
# Длины массивов корректны, проверяем циклом совпадение элементов
if [[ ${#arrres1[@]} != ${#arrres2[@]} ]]; then
    echo "Последовательности чисел в файлах $1 и $2 не совпадают"
    exit 1
else
    for i in ${!arrres1[@]}; do
        if [[ "$arrres1[$i]" != "$arrres2[$i]" ]]; then
            echo "Последовательности чисел в файлах $1 и $2 не
совпадают"
            exit 1
        fi
    done
fi
echo "Последовательности чисел в файлах $1 и $2 совпадают"
exit 0

```

Вариант 4:

```

#!/bin/bash
# Проверка корректности количества переданных аргументов
[[ "$#" = 2 || "$#" = 3 ]] || exit 1
# Проверка существования обоих файлов
[[ -f "$1" && -f "$2" ]] || exit 1
# Проверка возможности чтения обоих файлов
[[ -r "$1" && -r "$2" ]] || exit 1

```



```

# Корректность ключа verbose
[[ "$#" = 3 && "$3" != "-v" ]] && exit 1
[[ "$#" = 3 && "$3" = "-v" ]] && echo "Дополнительная информация: - "
# Собираем числа из первого файла в первый массив
arrres1=()
i1=0
while read text1; do
    for cell1 in $text1; do
        if [[ "$cell1" =~ ^[+-]?[0-9]+$ ]]; then
            arrres1[$i1]="{ $cell1}"
            i1=$((i1+1))
        elif [[ "$cell1" =~ ^[+-]?[0-9]+\.[0-9]+$ ]]; then
            arrres1[$i1]="{ $cell1}"
            i1=$((i1+1))
        elif [[ "$cell1" =~ ^[+-]?[0-9]+\.[0-9]*[eE]?[-]?[0-9]+$ ]]; then
            arrres1[$i1]="{ $cell1}"
            i1=$((i1+1))
        fi
    done
done < $1
# Собираем числа из второго файла во второй массив
arrres2=()
i2=0
while read text2; do
    for cell2 in $text2; do
        if [[ "$cell2" =~ ^[+-]?[0-9]+$ ]]; then
            arrres2[$i2]="{ $cell2}"
            i2=$((i2+1))
        elif [[ "$cell2" =~ ^[+-]?[0-9]+\.[0-9]+$ ]]; then
            arrres2[$i2]="{ $cell2}"
            i2=$((i2+1))
        elif [[ "$cell2" =~ ^[+-]?[0-9]+\.[0-9]*[eE]?[-]?[0-9]+$ ]]; then
            arrres2[$i2]="{ $cell2}"
            i2=$((i2+1))
        fi
    done
done < $2
# Проверяем длины массивов
if [[ "${#arrres1[@]}" = 0 && "${#arrres2[@]}" = 0 ]]; then
    echo "В обоих текстах нет ни одного числа"
    exit 1
else
    if [[ "${#arrres1[@]}" = 0 ]]; then
        echo "В первом тексте нет ни одного числа"
        exit 1
    elif [[ "${#arrres2[@]}" = 0 ]]; then
        echo "Во втором тексте нет ни одного числа"
        exit 1
    fi
fi

```

```

fi
fi
# Длины массивов корректны, проверяем циклом совпадение элементов
if [[ ${#arrres1[@]} != ${#arrres2[@]} ]]; then
    echo "Последовательности чисел в файлах $1 и $2 не совпадают"
    exit 1
else
    for i in ${!arrres1[@]}; do
        if [[ "$arrres1[$i]" != "$arrres2[$i]" ]]; then
            echo "Последовательности чисел в файлах $1 и $2 не
совпадают"
            exit 1
        fi
    done
fi
echo "Последовательности чисел в файлах $1 и $2 совпадают"
exit 0

```

Тестирование сценариев командной оболочки

Тестирование	Вариант 1	Вариант 2	Вариант 3	Вариант 4
Одинаковые файлы	Успех	Успех	Успех	Успех
Пустые файлы	Успех	Успех	Успех	Успех
Отсутствие файлов	Успех	Успех	Успех	Успех
Разные файлы	Успех	Успех	Успех	Успех
Только один файл заполнен	Успех	Успех	Успех	Успех
Передан только один аргумент	Успех	Успех	Успех	Успех

Заключение

В процессе работы над индивидуальным итоговым заданием были созданы сценарии командной оболочки, которые реализовывали сравнения двух файлов по вариантам 1-4, которые были указаны выше. Кроме того, были созданы сценарии, которые реализуют проверку. А также тесты необходимые для того, чтобы удостовериться в корректности работы сценариев.

В первом варианте создан сценарий, который сравнивает последовательности целых чисел в файлах. Целые числа отделяются от других символов в файле пробельными символами. Сравниваются числа как текст — числа 100 и 0100 следует считать различными.

Название скрипта: `comparator1.sh`.

Во втором варианте создан сценарий, который сравнивает 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`.

В третьем варианте создан сценарий, который сравнивает последовательности чисел с плавающей точкой (ЧПТ), записанных не в экспоненциальной форме. Сравниваются числа как текст — например, числа 1.0 и 1.00 следует считать различными.

Название скрипта: `comparator3.sh`.

В третьем варианте создан сценарий, который сравнивает последовательности чисел с плавающей точкой (ЧПТ), записанных в том числе в экспоненциальной форме. Сравниваются числа как текст — например, числа 1.0 и 1.00 следует считать различными, числа 1.0 и 1.0e0 – тоже.

Цель работы была достигнута, все задачи выполнены.

Список использованных источников

Регулярные выражения:

<https://serverspace.ru/support/help/regulyarnye-vyrazheniya/>

<https://habr.com/ru/company/ruvds/blog/327896/>

<https://linuxhint.com/equal-tilde-bash/>