



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» (ИУ)

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии
«(ИУ7)»

ОТЧЕТ ПО ПРОЕКТНО-ТЕХНОЛОГИЧЕСКОЙ ПРАТИКЕ

Студент:
Нгуен Ань Зунг

Группа:
ИУ7И-11Б

Руководителя:
Ломовской Игорь Владимирович
Кострицкий Александр Сергеевич

Москва 2022

ОГЛАВЛЕНИЕ

В	
Р	
ЗАКЛЮЧЕНИЕ.....	6
Л	

И
С
Х
О
Д
Н
О
Г
О

К
О
Д
А
..8
..5

·
·
·
·
·
·

ВВЕДЕНИИ

Разработать скрипт командной оболочки для сравнения содержимого двух текстовых файлов по определённым правилам.

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), за-

писанных **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

вычлe- нять и анализировать нужно именно их. Сравниваются числа как текст — например, числа 1.0 и 1.00 следует считать различными, числа 1.0 и 1.0e0 – тоже.

Название скрипта: comparator4.sh.

РАЗЛИЧНЫЕ ТЕСТОВЫЕ СЛУЧАИ

1. Некоторые примеры «регулярных регулярные выражение», использованные в статье:

Образцы шаблонов	
([A-Za-z0-9-]+)	Буквы, числа и знаки переноса
(\d{1,2}\d{1,2}\d{4})	Дата (напр., 21/3/2006)
([^\s]+(?:\.(jpg gif png))\.\2)	Имя файла jpg, gif или png
(^[1-9]{1}\$ ^[1-4]{1}[0-9]{1}\$ ^[50]\$)	Любое число от 1 до 50 включительно
(#?([A-Fa-f0-9]){3}([A-Fa-f0-9]){3})?)	Шестнадцатичный код цвета
((?=[*\d])(?=[*a-z])(?=[*A-Z]).{8,15})	От 8 до 15 символов с минимум одной цифрой, одной заглавной и одной строчной буквой (полезно для паролей).
(\w+@[a-zA-Z_]+?\.[a-zA-Z]{2,6})	Адрес email
(\<\/?[^\>]+\>)	HTML теги

2. Как пользоваться "Право доступа" использованным в статье

- **r** — read (чтение) — право просматривать содержимое файла;
- **w** — write (запись) — право изменять содержимое файла;
- **x** — execute (выполнение) — право запускать файл, если это программа или скрипт.

3. Некоторые способы использования `cmp` в Linux

1. Сравнение содержимого двух файлов
 - `$ cmp [имя-файла-1] [имя-файла-2]`
2. Вывод отличающихся значений байтов
 - `$ cmp -b [имя-файла-1] [имя-файла-2]`
3. Пропуск начальных байтов двух файлов
 - `$ cmp -i [количество-пропускаемых-байтов] [имя-файла-1] [имя-файла-2]`
4. Вывод информации о позициях (и значениях) всех отличающихся байтов
 - `$ cmp -l [имя-файла-1] [имя-файла-2]`
5. Сокращение вывода
 - `$ cmp -s [имя-файла-1] [имя-файла-2]`

6. Ограничение количества байтов для сравнения
- `$ cmp -n [количество-байтов-для-сравнения] [имя-файла-1] [имя-файла-2]`

ЗАКЛЮЧЕНИЕ

1. Comparator1.sh

Чтобы сравнить целые числа в двух файлах, мы сначала проверим, существует ли файл или нет. Если файл не существует, будет напечатан экран «Не удастся найти файл». Если файл существует, мы продолжим проверять права доступа к файлу. Далее мы будем запускать каждую строку в файле и использовать «регулярные выражения», чтобы найти целое число. Затем мы будем использовать «`cmp -s`» для сравнения двух файлов. Когда 2 файла одинаковы, он будет печатать «файлы совпадают» и, наоборот, печатать «файлы не совпадают».

3. Comparator3.sh

Чтобы сравнить числа с плавающей точкой в диапазон чисел двойной точности в двух файлах, мы сначала проверим, существует ли файл или нет. Если файл не существует, будет напечатан экран «Не удастся найти файл». Если файл существует, мы продолжим проверять права доступа к файлу. Далее мы будем запускать каждую строку в файле и использовать «регулярные выражения», чтобы найти целое число. Затем мы будем использовать «`cmp -s`» для сравнения двух файлов. Когда 2 файла одинаковы, он будет печатать «файлы совпадают» и, наоборот, печатать «файлы не совпадают».

4. Comparator4.sh

Чтобы сравнить последовательности чисел с плавающей точкой (ЧПТ), записанных в том числе в экспоненциальной форме в двух файлах, мы сначала проверим, существует ли файл или нет. Если файл не существует, будет напечатан экран «Не удастся найти файл». Если файл существует, мы продолжим проверять права доступа к файлу. Далее мы будем запускать каждую строку в файле и использовать

«регулярные выражения», чтобы найти целое число. Затем мы будем использовать «`str -s`» для сравнения двух файлов. Когда 2 файла одинаковы, он будет печатать «файлы совпадают» и, наоборот, печатать «файлы не совпадают».

ЛИСТИНГ ИСХОДНОГО КОДА

```
N1.  
  
#!/bin/bash  
  
if [ ! -f $1 ]; then  
    echo Ошибка! Файл 1 не найден!  
    exit 2  
fi  
  
if [ ! -f $2 ]; then  
    echo Ошибка! Файл 2 не найден!  
    exit 2  
fi  
  
if [ ! -r "$1" ]; then  
    echo Ошибка! У вас нет прав доступа к файлу 1!  
    exit 2  
fi  
  
if [ ! -r "$2" ]; then  
    echo Ошибка! У вас нет прав доступа к файлу 2!  
    exit 2  
fi  
  
file1=$(cat $1)  
file2=$(cat $2)
```



```
file1_nums=''

for word in $file1; do

    echo $word

    if [[ "$word" =~ ^[-+]?[0-9]+$ ]]; then

        file1_nums="$file1_nums $word"

        echo file1_nums

    fi

done
```

```
file2_nums=''

for word in $file2; do

    echo $word

    if [[ "$word" =~ ^[-+]?[0-9]+$ ]]; then

        file2_nums="$file2_nums $word"

        echo file2_nums

    fi

done
```

```
if cmp -s "$file1_nums" "$file2_nums"; then

    echo Файлы совпадают

    exit 0

else

    echo Файлы не совпадают

    exit 1

fi
```

```
fi

N3.

#!/bin/bash

if [ ! -f $1 ]; then

    echo Ошибка! Файл 1 не найден!

    exit 2

fi

if [ ! -f $2 ]; then

    echo Ошибка! Файл 2 не найден!

    exit 2

fi

if [ ! -r "$1" ]; then

    echo Ошибка! У вас нет прав доступа к файлу 1!

    exit 2

fi

if [ ! -r "$2" ]; then

    echo Ошибка! У вас нет прав доступа к файлу 2!

    exit 2

fi

file1=$(cat $1)

file2=$(cat $2)
```

```
file1_nums=''

for word in $file1; do

    echo $word

    if [[ "$word" =~ ^[+-]?[0-9]*[.]?[0-9]{1,2}$ ]] ; then

        file1_nums="$file1_nums $word"

        echo file1_nums

    fi

done

file2_nums=''

for word in $file2; do

    echo $word

    if [[ "$word" =~ ^[+-]?[0-9]*[.]?[0-9]{1,2}$ ]] ; then

        file2_nums="$file2_nums "$word"

        echo file2_nums

    fi

done

if cmp -s "$file1_nums" "$file2_nums"; then

    echo Файлы совпадают

    exit 0

else

    echo Файлы не совпадают

    exit 1

fi
```

N4.

```
#!/bin/bash
```

```
if [ ! -f $1 ]; then
```

```
    echo Ошибка! Файл 1 не найден!
```

```
    exit 2
```

```
fi
```

```
if [ ! -f $2 ]; then
```

```
    echo Ошибка! Файл 2 не найден!
```

```
    exit 2
```

```
fi
```

```
if [ ! -r "$1" ]; then
```

```
    echo Ошибка! У вас нет прав доступа к файлу 1!
```

```
    exit 2
```

```
fi
```

```
if [ ! -r "$2" ]; then
```

```
    echo Ошибка! У вас нет прав доступа к файлу 2!
```

```
    exit 2
```

```
fi
```

```
file1=$(cat $1)
```

```
file2=$(cat $2)
```

```
file1_nums=''
```

```
for word in $file1; do

    echo $word

    if [[ "$word" =~ ^[+-]?[0-9]*[. ,]?[0-9]{1,2}([eE][+-]?[0-9]+)?$ ]]; then

        file1_nums="$file1_nums $word"

        echo file1_nums

    fi

done

file2_nums=''

for word in $file2; do

    echo $word

    if [[ "$word" =~ ^[+-]?[0-9]*[. ,]?[0-9]{1,2}([eE][+-]?[0-9]+)?$ ]]; then

        file2_nums="$file2_nums $word"

        echo file2_nums

    fi

done

if cmp -s "$file1_nums" "$file2_nums"; then

    echo Файлы совпадают

    exit 0

else

    echo Файлы не совпадают

    exit 1

fi
```