Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей
Кафедра информатики
Дисциплина: Операционные среды и системное программирование

ОТЧЕТ
к лабораторной работе №2
на тему

ОБРАБОТКА ТЕКСТОВОЙ ИНФОРМАЦИИ. РЕГУЛЯРНЫЕ ВЫРАЖЕНИЯ

Студент О. Л. Дайнович
Преподаватель Н. Ю. Гриценко

Минск 2024

СОДЕРЖАНИЕ

1 ЦЕЛЬ РАБОТЫ

Изучить методы и средства обработки текстовой информации, включая регулярные выражения, и использующих их утилит. Написать скрипт для sed, awk и т.д., либо скрипт shell, обращающийся к необходимым программам, для обработки и автокоррекции входных данных (файлов). В частности, разрабатываемая программа должна заменять строчные буквы на заглавные в начале предложения. Необходимо также предусмотреть поведение скрипта (скриптов) при ошибочных или «неочищенных» входных данных.

2 ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Sed, сокращение от «stream editor», является мощным инструментом командной строки для обработки текстовых данных. Он предназначен для преобразования потокового ввода или текстовых файлов в соответствии с заданными правилами.

Основными операциями, которые sed может выполнять, являются поиск и замена, удаление строк, вставка текста, а также обработка текстового вывода других программ.

Sed работает построчно, применяя указанные шаблоны и действия к каждой строке текста. Это делает его полезным инструментом для автоматизации обработки текстовых данных, например, в скриптах оболочки или командных файлов.

Помимо базовых операций поиска и замены, sed также поддерживает использование регулярных выражений, что делает его еще более мощным инструментом для манипуляции текстом. Регулярные выражения позволяют точно определить шаблоны для поиска и замены в тексте.

Кроме того, sed поддерживает флаги и опции, которые позволяют настраивать его поведение, такие как флаги глобального поиска и замены, флаги, управляющие выводом, и многое другое. [1]

AWK – это скриптовый язык, который полезен при работе в командной строке и широко применяется для обработки текста. При использовании AWK можно выбирать данные – один или более отдельных фрагментов текста – на основе заданного критерия. Например, с помощью awk можно выполнять поиск конкретного слова или шаблона во фрагменте текста, а также выбирать определённую строку/столбец в файле, выполнять подстановку, замену или вывод по определенному шаблону. [2]

Регулярные выражения (Regular Expressions, или regex) – это мощный инструмент для работы с текстом, который позволяет искать и сопоставлять строки, основываясь на шаблонах. Они широко используются для поиска, замены, валидации или извлечения информации из текстовых данных. [3]

3 ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ

В результате лабораторной работы был написан скрипт, реализующий замену строчных букв на заглавные в начале предложений, т.е. в начале документа и после точки, не находящейся внутри, например, числа, а также после знаков «!», «?».

Скрипт считывает исходный текст из файла «input» расширения .txt и редактирует его, заменяя первые буквы каждого предложения на заглавные (рисунок 1).

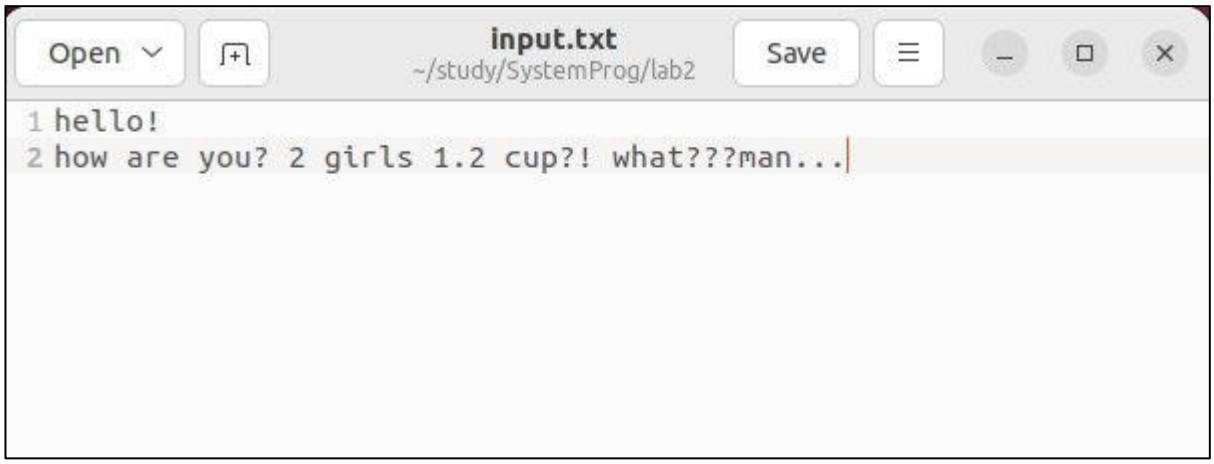

Рисунок 1 – Исходный текст

После изменения регистра букв, скрипт записывает готовый текст в файл «output» расширения .txt (рисунок 2).

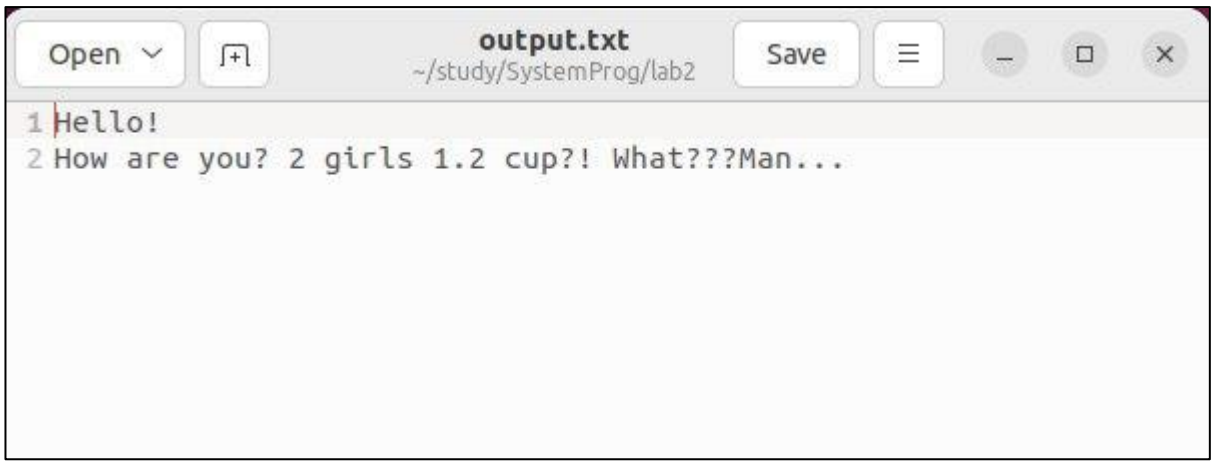

Рисунок 2 – Итоговый текст

ЗАКЛЮЧЕНИЕ

В ходе данной лабораторной работы были изучены методы и средства обработки текстовой информации, включая регулярные выражения, и использующих их утилит. В ходе работы был написан bash скрипт, для обработки входных данных, реализующий замену строчных букв на заглавные в начале предложений.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] Sed a stream editor [Электронный ресурс]. – Режим доступа: https://www.gnu.org/software/sed/manual/sed.html – Дата доступа: 19.02.2024.

[2] Команда awk [Электронный ресурс]. – Режим доступа: https://habr.com/ru/companies/ruvds/articles/665084/ – Дата доступа: 19.02.2024.

[3] Регулярные выражения [Электронный ресурс]. – Режим доступа: https://aidalinux.ru/w/regex – Дата доступа: 19.02.2024.

ПРИЛОЖЕНИЕ А
(обязательное)
Листинг кода

Листинг 1 – Файл lab2.sh

```
#!/bin/bash

input_file="input.txt"
output_file="output.txt"

text=$(<"$input_file")
regex="[\.|!|\?]\s*([a-z])"

text=${text^}

is_correct=0
while [[ $is_correct == 0 ]]; do
  is_correct=1
  if [[ $text =~ $regex ]]; then
    is_correct=0

    wrong_part="${BASH_REMATCH[0]}"
    echo "$wrong_part"

    new_part="${BASH_REMATCH[0]}"
    new_part=${new_part::-1}
    new_part+="${BASH_REMATCH[1]^}"
    echo "$new_part"

    text=${text/$wrong_part/$new_part}
  fi
done

echo "$text" > $output_file
echo "$text"
```