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 И. В. Бобко

Преподаватель Н. Ю. Гриценко

Минск 2024

СОДЕРЖАНИЕ

[1 Цель работы 3](#_Toc146752068)

[2 Теоретические сведения 4](#_Toc146752069)

[3 Результат выполнения 6](#_Toc146752070)

[Заключение 6](#_Toc146752071)

[Список использованных источников 7](#_Toc146752072)

[Приложение А (обязательное) Листинг кода 8](#_Toc146752073)

1. **ЦЕЛЬ РАБОТЫ**

Целью данной работы является изучение методов и средств обработки текстовой информации, включая регулярные выражения, и использующих их утилит. В ходе работы будет написан скрипт для sed, awk и т.д., либо скрипт shell, обращающийся к необходимым программам, для обработки входных данных, реализующий замену строчных букв на заглавные в начале предложений.

1. **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 [1].

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.

Благодаря своей простоте в использовании и мощным возможностям, sed является популярным инструментом для обработки текстовых данных в различных сценариях, от автоматизации задач до фильтрации и обработки вывода программ.

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 [2].

1. **РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ**

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 «!», «?» (рисунок 1).

Предложение может начинаться с новой строки (т.е. предыдущая точка может находиться в одной строке, а заменяемая строчная буква – в следующей).

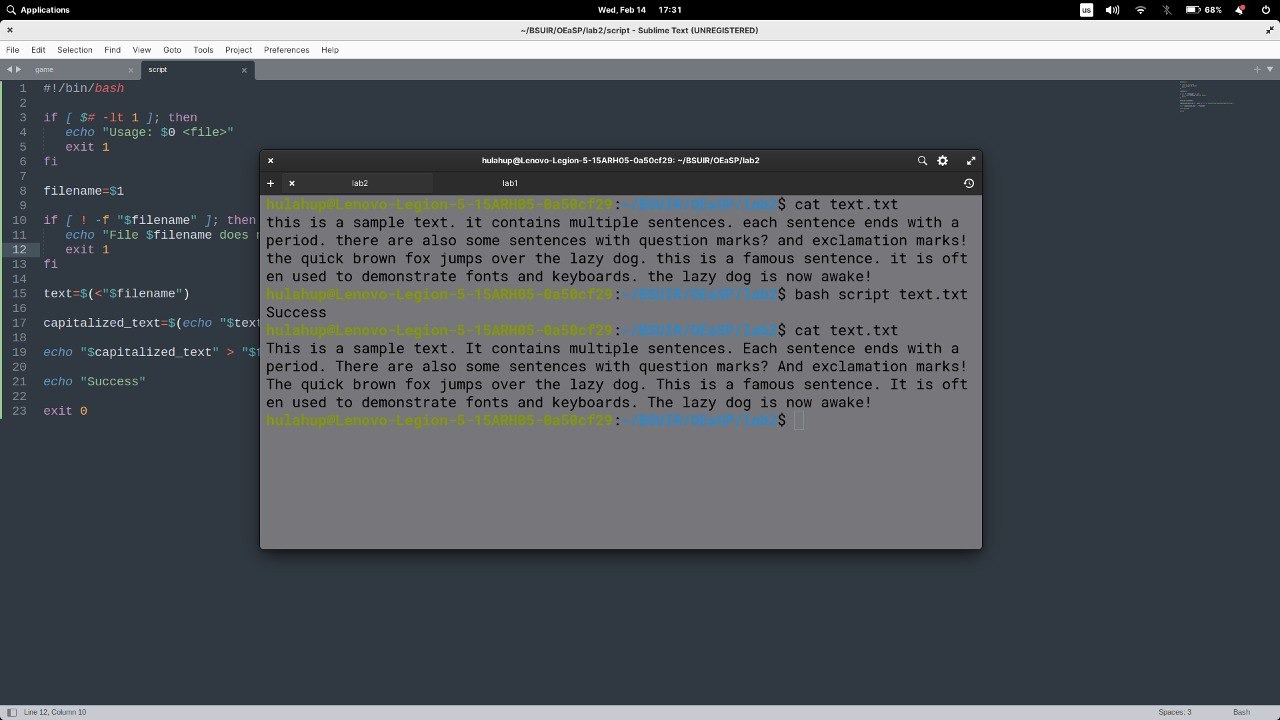


Рисунок 1 – Результаты выполнения скрипта

# 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 скрипт для sed, 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: 12.02.2024.
2. Регулярные выражения [Электронный ресурс]. – Режим доступа: https://aidalinux.ru/w/regex – Дата доступа: 12.02.2024.

# ПРИЛОЖЕНИЕ А (обязательное) Листинг кода

Листинг 1 – Файл script

#!/bin/bash

if [ $# -lt 1 ]; then

echo "Usage: $0 <file>"

exit 1

fi

filename=$1

if [ ! -f "$filename" ]; then

echo "File $filename does not exist."

exit 1

fi

text=$(<"$filename")

capitalized\_text=$(echo "$text" | sed -E 's/([.!?]\s\*|^)([[:alpha:]])/\1\U\2/g')

echo "$capitalized\_text" > "$filename"

echo "Success"

exit 0