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 П. Н. Носкович

Преподаватель Н. Ю. Гриценко

Минск 2025

**СОДЕРЖАНИЕ**

[1 Цель работы 3](#_Toc159347008)

[2 Краткие теоретические сведения 4](#_Toc159347009)

[3 Результаты выполнения лабораторной работы 7](#_Toc159347010)

[Выводы 8](#_Toc159347011)

[Список использованных источников 9](#_Toc159347012)

[Приложение А (обязательное) Листинг кода 10](#_Toc159347013)

## 1 ЦЕЛЬ РАБОТЫ

Целью выполнения данной лабораторной работы является изучение методов и средств обработки текстовой информации, включая регулярные выражения, и использующих их утилит. Разработать и реализовать программу, осуществляющую замену строчных букв на заглавные в начале предложений, т.е. в начале документа и после точки, не находящейся внутри.

## 2 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Исторически первый стандартный текстовый редактор в *Unix*-системах – однострочный командный редактор *ed (EDitor*). Важная особенность: постоянное отображение редактируемого текста не предполагается, все действия над текстом (в т. ч. отображение) выполняются по командам. Расширенная версия *ed – ex* (*EXtended editor*). Тоже однострочный, но существенно расширен набор команд для сложного поиска и модификации в тексте. Это множестве команд поддерживается и в редакторе *vi*. [1]

Редактор *ed* нельзя считать в полной мере фильтром: его поток ввода – это команды и их данные. Однако *sed* и *grep* – типичные фильтры, имеющие собственный командный язык.

Редакторы *ed, vi* и *sed* загружают текстовый документ в буфер и выполняют над ним операции согласно командам, поступающим из потока ввода (*stdin*) или файла. Текст в течение всего сеанса работы остается в буфере, запись в файл только по команде, вывод – тоже (для *ed* и *sed*). В зависимости от конкретной утилиты и способа ее выполнения, источником документа может служить как файл, так и входной поток (*stdin*).

«Потоковый редактор» *sed* (*Stream EDitor*) – неинтерактивное средство обработки текстов. Можно рассматривать как сложный универсальный фильтр для разнообразной обработки текстовых данных, имеющий собственный специализированный командный язык с широким использованием регулярных выражений. Основной сценарий использования – применение к входному потоку текстовых данных одной или нескольких команд, но может работать и с входным файлом или множеством файлов (аналогично большинству программ-фильтров). Результаты всегда передаются в поток и при необходимости могут быть перенаправлены. [2]

Утилита (точнее, семейство утилит) *grep* – программа-фильтр для поиска по образцу строк в файлах и/или потоках. Одна из наиболее часто используемых программ. По умолчанию *grep* выводит строки, в которых обнаружено соответствие с заданным образцом поиска. Образец рассматривается как регулярное выражение. Таким образом, *grep* выполняет более узкий по сравнению с *sed* набор функций и поэтому не нуждается в отдельном командном языке, но одновременно он имеет дополнительные возможности именно поиска (фильтрации). [3]

Регулярное выражение (*regular expression* или *regexp*) – последовательность символов и метасимволов (специальных символов), которую может интерпретировать (разобрать) конечный автомат. И наоборот, конечный автомат может быть определен через возможность обработки им регулярных выражений. В более узком смысле регулярное выражение – способ описать множество текстовых строк через соответствие данному регулярному выражению. Аналогично, регулярное выражение может быть использовано для порождения множества соответствующих ему текстовых строк.

## 3 РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

В ходе выполнения лабораторной работы была реализована программа для автокоррекции предложений. Эта программа предназначена для автокоррекции предложений. Она помогает улучшить качество текста и исправить ошибки.

Реализованный скрипт позволяет обрабатывать текст, находящийся в файле, и сохранять его в нужной форме. Используемый инструментарий включает *sed* и регулярные выражения, которые обеспечивают гибкость и точность выполнения.

Дополнительно были изучены и применены методы поиска в текстах. Например, использовались команды *grep* и *awk* для извлечения шаблонов текста, таких как палиндромы длиной пять символов. Также была реализована возможность извлечения текста между заданными символами, независимо от длины промежуточного содержания.

Во время работы подробно исследована команда *chmod*, которая позволила наделить скрипт правами на выполнение. Установленные права доступа обеспечивают возможность запуска программы как самостоятельного исполняемого файла.

Результаты работы свидетельствуют о корректной реализации алгоритмов, применяемых для анализа и обработки текстовых данных.

# ВЫВОДЫ

В ходе выполнения данной лабораторной работы была разработана и реализована программа, которая выполняет автокоррекцию текста, а именно – производит замену строчных букв на заглавные в начале предложений, т. е. в начале документа и после точки, не находящейся внутри.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] [Электронный ресурс]. – Режим доступа: https://parallel.uran.ru/book/export/html/493 – Дата доступа 06.02.2025

[2] [Электронный ресурс]. – Режим доступа: https://losst.pro/komanda-sed-linux – Дата доступа: 06.02.2025

[3] [Электронный ресурс]. – Режим доступа: https://highload.today/grep-regex/ – Дата доступа: 06.02.2025

# ПРИЛОЖЕНИЕ А (обязательное) Листинг кода

lab2.sh

#!/bin/bash # Чтение из файла и обработка текста

sed -e 's/\([.!?]\)\([[:space:]]\*\)\([a-zа-я]\)/\1\2\U\3/g' Text.txt |

# Обработка первой строки текста

sed '1s/^[a-zа-я]/\U&/'