

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

ОТЧЁТ  
к лабораторной работе №2  
на тему

**ОБРАБОТКА ТЕКСТОВОЙ ИНФОРМАЦИИ. РЕГУЛЯРНЫЕ  
ВЫРАЖЕНИЯ**

Выполнил: студент гр. 253503  
Тимошевич К.С.

Проверил: ассистент кафедры  
информатики Гриценко Н.Ю.

Минск 2025

## СОДЕРЖАНИЕ

1 Постановка задачи.....	3
2 Описание работы программы.....	4
2.1 Обработка входных данных.....	4
2.2 Обработка текста и исправление регистра.....	4
2.3 Запись результата.....	4
3 Ход выполнения программы.....	5
3.1 Примеры выполнения задания.....	5
Вывод.....	6
Список использованных источников.....	7
Приложение А (справочное) Исходный код.....	8

# 1 ПОСТАНОВКА ЗАДАЧИ

Целью данной лабораторной работы является изучение методов обработки текстовой информации в среде *Unix/Linux* с использованием регулярных выражений и специализированных утилит, таких как *sed*, *awk*, а также возможностей командного интерпретатора *shell* [1]

В рамках работы необходимо разработать скрипт, который выполняет автоматическую коррекцию заглавных букв в тексте. Скрипт должен обрабатывать входные данные (текстовый файл) и изменять регистр первых букв предложений, приводя их к заглавному виду.

Коррекция должна выполняться в следующих случаях: в начале документа после точки, за исключением случаев, когда она является десятичным разделителем числа а также после знаков ! и ?.

Если предложение начинается с новой строки (т.е. точка, восклицательный или вопросительный знак находятся в предыдущей строке).

Скрипт должен учитывать возможные ошибки входных данных, такие как пробельные символы перед предложением, и корректно обрабатывать их. В ходе выполнения лабораторной работы необходимо продемонстрировать навыки использования регулярных выражений и текстовых утилит (*sed*, *awk*), а также основы работы с командной оболочкой *shell*.

## 2 ОПИСАНИЕ РАБОТЫ ПРОГРАММЫ

Данный раздел описывает основные функции скрипта, выполняющего автоматическую коррекцию заглавных букв в тексте. Рассмотрены этапы обработки входных данных, применения регулярных выражений и сохранения результата.

### 2.1 Обработка входных данных

Скрипт принимает один аргумент — путь к файлу, который необходимо обработать. На первом этапе проверяется корректность переданного аргумента. Если аргумент отсутствует или их больше одного, программа выводит сообщение об использовании и завершает выполнение [2]. Если указанный файл не существует, выводится сообщение об ошибке.

### 2.2 Обработка текста и исправление регистра

После проверки файла выполняется его обработка с помощью утилиты *sed*. Применяется регулярное выражение, которое приводит к заглавному регистру первую букву предложения в следующих случаях: в начале текста после символов *.*, *!*, *?*, за которыми следует пробел или новая строка и если предложение начинается с новой строки.

### 2.3 Запись результата

После обработки исправленный текст временно записывается в файл *input\_file.tmp*, затем этот временный файл заменяет оригинальный файл. Такой подход предотвращает потерю данных в случае ошибки обработки. Если *sed* выполняется напрямую с перезаписью (*sed -i*), а во время работы скрипта произойдет ошибка (например, внезапное завершение программы), исходный файл может повредиться или очиститься. [3] Использование временного файла снижает риск потери данных.

## 3 ХОД ВЫПОЛНЕНИЯ ПРОГРАММЫ

### 3.1 Примеры выполнения задания

На рисунке 3.1 показано состояние файла до выполнения скрипта.

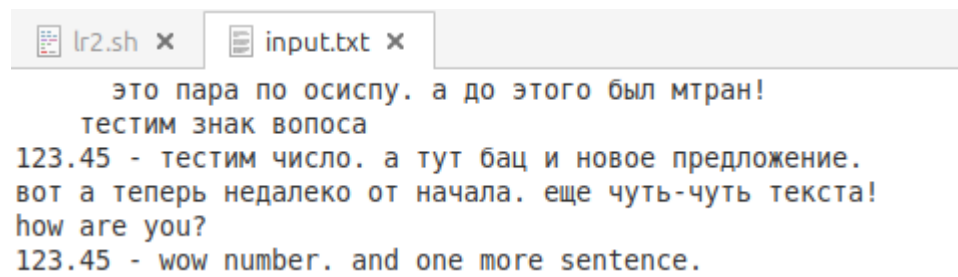


Рисунок 3.1 - Состояние файла до работы скрипта

На рисунке 3.2 продемонстрирован результат работы скрипта после его выполнения.

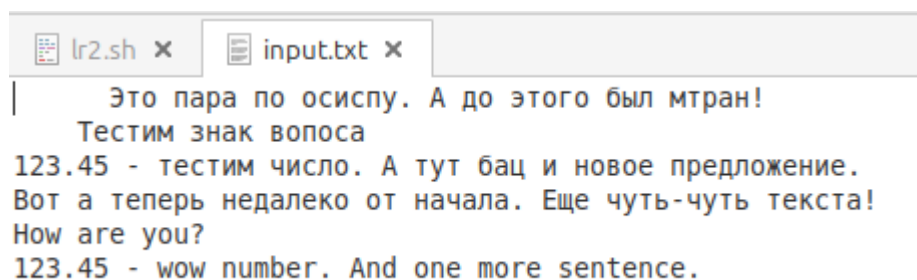


Рисунок 3.2 - Результат обработки файла

## ВЫВОД

В ходе выполнения лабораторной работы был разработан и реализован скрипт на языке *shell*, который выполняет автоматическую коррекцию заглавных букв в тексте. Скрипт предназначен для обработки входных данных (текстового файла) и исправления регистра первых букв предложений в соответствии с установленными правилами.

Во-первых, в скрипте реализована проверка входных данных, что позволяет избежать ошибок при передаче аргументов командной строки. Если файл не найден или аргументы переданы некорректно, программа уведомляет пользователя и завершает выполнение.

Во-вторых, обработка текста выполняется с использованием регулярных выражений и утилиты *sed*. Скрипт корректно обрабатывает случаи, когда предложения начинаются с новой строки или следуют после знаков ., !, ?, игнорируя точки внутри чисел. Это позволяет достичь точности в исправлении регистра.

В-третьих, в целях безопасности данных используется механизм временного файла. Исправленный текст сначала записывается во временный файл, а затем заменяет оригинальный. Такой подход предотвращает потерю данных при возможных сбоях или ошибках во время выполнения программы.

В результате работы был создан скрипт, способный автоматически исправлять заглавные буквы в предложениях, используя встроенные инструменты *Unix/Linux*. Он демонстрирует применение регулярных выражений, работу с потоками данных и основы автоматизированной обработки текстовых файлов в командной оболочке.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] Bash-скрипты: параметры и ключи командной строки [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/companies/ruvds/articles/326328/>. – Дата доступа: 02.02.2024.

[2] Bash-скрипты: функции и разработка библиотек [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/companies/ruvds/articles/327248/>. – Дата доступа: 02.02.2024.

[3] Bash-скрипты, часть 9: регулярные выражения [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/companies/ruvds/articles/327896/>. – Дата доступа: 02.02.2024.

## ПРИЛОЖЕНИЕ А

### (справочное)

### Исходный код

```
#!/bin/bash

if [ $# -ne 1 ]; then
    echo "Использование: $0 <входной_файл>"
    exit 1
fi

input_file="$1"

if [ ! -f "$input_file" ]; then
    echo "Ошибка: файл '$input_file' не найден!"
    exit 1
fi

sed -E 's/^(^|[\.\!\?]\s+)([[:space:]]*)([a-za-яё])/\1\2\U\3/g' "$input_file" >
"$input_file.tmp" && mv "$input_file.tmp" "$input_file"
```