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»

Выполнил             Е. А. Киселёва

Проверил                          Н. Ю. Гриценко

Минск 2024

**СОДЕРЖАНИЕ**

[1 Постановка задачи 3](#_Toc157722973)

[2 Краткие теоретические сведения 4](#_Toc157722974)

[3 Результаты выполнения лабораторной работы 5](#_Toc157722975)

[Выводы 6](#_Toc157722976)

[Список использованных источников 7](#_Toc157722977)

[Приложение А (обязательное) Листинг исходного кода 8](#_Toc157722978)

# **1 ПОСТАНОВКА ЗАДАЧИ**

Целью выполнения данной лабораторной работы является изучение методов и средств обработки текстовой информации, включая регулярные выражения, и использующих их утилит. Кроме того, необходимо написать скрипт, обращающийся к необходимым программам, для обработки входных данных, а также предусмотреть поведение скрипта при ошибочных или «неочищенных» входных данных.

# **2 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

Обработка текстовой информации в среде оболочки часто выполняется с использованием различных утилит и команд, таких как grep, sed, awk.

Утилита grep используется для поиска строк в текстовых файлах, которые соответствуют определенному шаблону или регулярному выражению. Эффективна для быстрого поиска строк по паттернам, но не имеет возможности выполнения сложных манипуляций с данными.

Утилита sed используется для выполнения текстовых замен и преобразований в потоковых данных или в файлах. Удобна для выполнения простых текстовых замен, но может быть неэффективна для сложных обработок данных.

Утилита awk является языком программирования для обработки и анализа текстовых данных, разбивки строк на поля и выполнения операций над ними. Мощный инструмент для обработки и анализа текстовых данных.

Регулярные выражения – мощный инструмент для работы с текстовой информацией. Они используются для поиска, сопоставления и манипулирования строками текста с использованием определенных шаблонов. [1]

Для выполнения данной лабораторной работы были использованы следующие сведения и концепции:

1 Скрипты bash: скрипты bash являются подмножеством скриптов shell, предоставляющим большие возможности для автоматизации задач.

2 Управление потоком выполнения: используются условные операторы [ -z "$1" ], [ ! -f "$1" ] для проверки аргументов командной строки и существования файлов. Выполняется проверка условий, и в зависимости от результата выполняются различные действия.

3 Работа с переменными: используются переменные script\_dir, input, result для хранения различных данных.

4 Использование командной строки и параметров: используются встроенные переменные, такие как $0, для доступа к имени скрипта.

5 Работа с файлами: используются команды для работы с файлами: dirname для получения имени директории, cat для чтения содержимого файла, echo для вывода данных в файл.

6 Использование регулярных выражений: используются регулярные выражения для поиска определенных шаблонов в тексте файла.

7 Преобразование текста: применяются команды sed и echo для обработки и вывода текстовых данных.

# **3 РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ             РАБОТЫ**

В ходе выполнения лабораторной работы был разработан небольшой bash-скрипт, который заменяет строчные буквы на заглавные в начале предложений, то есть в начале документа и после точки, не являющейся частью числа, а также после знаков «!» и «?», предложение может начинаться с новой строки. Результат работы bash-скрипта представлен на рисунке 3.1.



Рисунок 3.1 – Результат работы bash-скрипта

Если при вызове скрипта указать несуществующий файл или не указать файла вообще, программа выдаст информацию о том, что файл не найден. Результат обработки отсутствия файла представлен на рисунке 3.2.



Рисунок 3.2 – Результат обработки отсутствия файла

Если при вызове скрипта указать файл, в котором нет букв, которые нужно исправить на заглавные, то программа выдаст информацию о том, что в файле и так все в порядке, то есть его не нужно корректировать. Результат обработки некорректных данных представлен на рисунке 3.3.



Рисунок 3.3 – Результат обработки некорректных данных

Результат работы скрипт выводит текстом в терминал, а также в файл результата, находящийся в той же директории, что и исходный файл.

# **ВЫВОДЫ**

В ходе лабораторной работы были изучены методы и средства обработки текстовой информации, включая регулярные выражения, и использующих их утилит. Кроме того, в ходе лабораторной работы был разработан скрипт, обращающийся к необходимым программам, для обработки входных данных, а также было предусмотрено поведение скрипта при ошибочных или «неочищенных данных».

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

[1] Инструменты для обработки текста в командной оболочке [Электронный ресурс]. – Режим доступа: https://habr.com/ru/companies/ itsumma/articles/492932/. – Дата доступа: 11.02.2024.

## **ПРИЛОЖЕНИЕ А**

## **(обязательное)**

## **Листинг исходного кода**

Листинг 1 – Программный код разработанного приложения

#!/bin/bash

script\_dir="$(dirname "$0")"

if [ -z "$1" ]; then

echo "Usage: $0 <filename>"

exit 1

fi

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

echo "File not found: $1"

exit 1

fi

input=$(cat "$1")

if ! echo "$input" | grep -q -E '(\.\s+[a-z])|(!\s+[a-z])|(\?\s+[a-z])|^[a-z]'; then

echo "In that file all good"

exit 0

fi

result=$(echo "$input" | sed -e 's/\.\s\{1,\}[a-z]/\U&/g' \

-e 's/!\s\{1,\}[a-z]/\U&/g' \

-e 's/\?\s\{1,\}[a-z]/\U&/g' \

-e 's/^[a-z]/\U&/')

echo "$result"

echo "$result" > "$script\_dir/result2"