**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №1**

**по дисциплине «Программирование»**

Тема: Регулярные выражения

|  |  |  |
| --- | --- | --- |
| Студент гр. 3341 |  | Анисимов Д.А. |
| Преподаватель |  | Глазунов С.А. |

Санкт-Петербург

2024

## Цель работы

Цель работы заключается в овладении навыками работы с регулярными выражениями и их применением через разработку программы на языке программирования Си. Для достижения этой цели предполагается выполнение следующих задач:

- Изучение основных структур и функций регулярных выражений;

- Формирование регулярного выражения, способного решить поставленную задачу;

- Написание программы, которая будет использовать созданное регулярное выражение для решения задачи.

## Задание

На вход программе подается текст, представляющий собой набор предложений с новой строки. Текст заканчивается предложением "***Fin.***" В тексте могут встречаться ссылки на различные файлы в сети интернет. Требуется, используя регулярные выражения, найти все эти ссылки в тексте и вывести на экран пары *<название\_сайта>* - *<имя\_файла>*. Гарантируется, что если предложение содержит какой-то пример ссылки, то после ссылки будет символ переноса строки.

Ссылки могут иметь следующий вид:

* Могут начинаться с названия протокола, состоящего из букв и **://** после
* Перед доменным именем сайта может быть ***www***
* Далее доменное имя сайта и один или несколько доменов более верхнего уровня
* Далее возможно путь к файлу на сервере
* И, наконец, имя файла с расширением.

## Выполнение работы

Перед написанием программы было создано регулярное выражение, которое ищет в тексте все ссылки:

*([a-z]+\\:\\/\\/)?(www\\.)?(([a-z]+\\.)+[a-z]+)\\/([a-z]+\\/)\*([a-z]+\\.[a-z0-9]+)*

Далее была написана программа, использующая представленное выше регулярное выражение.

Для работы с регулярными выражениями подключается библиотека *<regex.h>.*

Инициализируются следующая константа:

*const char\* regexPattern*– хранит в себе написанное регулярное выражение;

Была написана функция *void printMatchedLinks (char\* currentStr, regmatch\_t\* currentGroup)*, которая принимает на вход строку *buffer*, в которой была найдена ссылка, и массив групп *groups*. В функции с помощью двух циклов for, пробегающих значения между границами группы, указанными в *currentGroup*, посимвольно выводятся название сайта и имя файла.

Далее в функции *int main()* создаются переменная *regex\_* *compiledRegex* для компиляции регулярного выражения, массив *groups[1000]* для хранения индексов начала и конца групп, строка-буфер *groups[1000]*. С помощью функции *regcomp()* регулярное выражение компилируется.

Затем в цикле *do {…} while(),* выполняющемся до тех пор, пока не встретится маркер конца текста, считываются предложения текста с помощью функции *fgets()*, и, если в предложении будет найдена ссылка с помощью функции *regexec()*, ссылка будет выведена на экран с помощью *printMatchedLinks*.

В конце память от скомпилированного регулярного выражения очищается с помощью функции *regfree()*.

Разработанный программный код см. в приложении А.

## Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Входные данные | Выходные данные | Комментарии |
|  | Fin. |  |  |
|  | ftp://skype.com/qqwe/qweqw/qwe.avi  Fin. | skype.com - qwe.avi |  |
|  | This is simple url:  Fin. |  |  |
|  | This is simple url:  http://www.google.com/track.mp3  May be more than one upper level  domain http://www.google.com.edu/hello.avi  Many of them.  Rly. Look at this!  http://www.qwe.edu.etu.yahooo.org.net.ru/qwe.q  Some other protocols  ftp://skype.com/qqwe/qweqw/qwe.avi  Fin. | google.com - track.mp3  google.com.edu - hello.avi  qwe.edu.etu.yahooo.org.net.ru - qwe.q  skype.com - qwe.avi |  |

## Выводы

В ходе работы достигнуты следующие результаты: усовершенствованы навыки работы с регулярными выражениями, выполнены поставленные задачи, включая изучение основных конструкций регулярных выражений, написание шаблона для поиска ссылок в тексте и разработку программы на языке C, которая использует данное выражение для извлечения названия сайта и имени файла из ссылок, содержащихся в тексте.

# Приложение А Исходный код программы

Название файла: main.c

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<regex.h>

const char\* regexPattern = "([a-z]+\\:\\/\\/)?(www\\.)?(([a-z]+\\.)+[a-z]+)\\/([a-z]+\\/)\*([a-z]+\\.[a-z0-9]+)";

void printMatchedLinks(char\* currentStr, regmatch\_t\* currentGroups)

{

for(int i=currentGroups[3].rm\_so;i<currentGroups[3].rm\_eo;i++)

printf("%c", currentStr[i]);

printf(" - ");

for(int i=currentGroups[6].rm\_so;i<currentGroups[6].rm\_eo;i++)

printf("%c", currentStr[i]);

printf("\n");

}

int main()

{

regex\_t compiledRegex;

regmatch\_t groups[1000];

regcomp(&compiledRegex, regexPattern, REG\_EXTENDED);

char buffer[1000];

do

{

fgets(buffer, 1000, stdin);

if(regexec(&compiledRegex, buffer, 7, groups, 0)==0)

{

printMatchedLinks(buffer, groups);

}

}

while(strncmp(buffer, "Fin.", 1000));

regfree(&compiledRegex);

return 0;

}