**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №1**

**по дисциплине «Программирование»**

**Тема: Лабораторная работа № 1. Регулярные выражения**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 3343 |  | Стрижков И. А. |
| Преподаватель |  | Государкин Я. С. |

Санкт-Петербург

2024

## Цель работы

Изучить и научиться применять функции библиотеки regex.h языка Си для поиска совпадений в строках при помощи регулярных выражений. Освоить навыки для написания регулярных выражений на языке Си.

## 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

Описание функций:

* printMatch(char\* s, regmatch\_t groupArray): Функция для вывода совпадения, найденного регулярным выражением.
* main(): Основная функция программы, выполняющая следующие задачи:

- Инициализация переменных и компиляция регулярного выражения: Создает строку для хранения регулярного выражения и использует regcomp для его компиляции в скомпилированную структуру regex\_t.

- Считывание входных данных: Читает строки из стандартного ввода в массив s до тех пор, пока не будет введена строка "Fin.".

- Обработка ввода: Для каждой считанной строки использует regexec для выполнения регулярного выражения и поиска совпадений. Если совпадение найдено, outputs specific matches используя функцию printMatch - сначала домен, затем имя файла.

- Освобождение ресурсов: Вызывает regfree для освобождения ресурсов, связанных с скомпилированным регулярным выражением.

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 |
| 1. | 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 | Выходные данные соответствуют ожиданиям. |
| 2. | This is simple url: http://www.google.-aaaaaa.com/track.mp3  May be more than one upper level  domain http://www.google.com.edu/hello.avi  Many of them. youtube.en/file.f  Rly. Look at this!  Fin. | google.-aaaaaa.com - track.mp3  google.com.edu - hello.avi  youtube.en - file.f | Выходные данные соответствуют ожиданиям. |
| 3. | This is simple url: http://www.google.aaaaaa.com//track.mp3  May be more than one upper level  domain http://www.google.com.edu/hello.avi  Many of them. youtube.en/file.f  Rly. Look at this! This is simple url: aaa://googleaaaaaacom/a.a  May be more than one upper level  domain http://www.google\_google.com.edu/hello.avi  Fin. | google.com.edu - hello.avi  youtube.en - file.f  google\_google.com.edu - hello.avi | Выходные данные соответствуют ожиданиям. |

## Выводы

В ходе выполнения лабораторной работы были освоены необходимые навыки для использования регулярных выражений на языке Си с помощью библиотеки regex.h, а также для составления регулярных выражений согласно требованиям. Были изучены необходимые языковые конструкции и особенности записи регулярных выражений на языке Си.

# Приложение А Исходный код программы

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <regex.h>

#define INPUT\_SIZE 100

#define NUM\_INPUTS 100

void printMatch(char\* s, regmatch\_t groupArray) {

for (int i = groupArray.rm\_so; i < groupArray.rm\_eo; i++)

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

}

int main() {

size\_t maxGroups = 100;

char\* regexString = "([A-z]+:\\/\\/)?(w{3}\\.)?([A-z\_-]+(\\.[A-z\_-]+){1,})((\\/[A-z]+){1,})?(\\/([A-z]+\\.[A-z0-9]+\n))";

regex\_t regexCompiled;

char s[NUM\_INPUTS][INPUT\_SIZE];

int numInputs = 0;

if (regcomp(&regexCompiled, regexString, REG\_EXTENDED)) {

printf("can't compile regular expression\n");

return 0;

}

while (fgets(s[numInputs], INPUT\_SIZE, stdin) && numInputs < NUM\_INPUTS) {

if (strstr(s[numInputs], "Fin.") != NULL){

break;

}

numInputs++;

}

regmatch\_t groupArray[maxGroups];

for (int i = 0; i < numInputs; i++) {

if (regexec(&regexCompiled, s[i], maxGroups, groupArray, 0) == 0) {

printMatch(s[i], groupArray[3]);

printf(" - ");

if (groupArray[8].rm\_so != -1) {

printMatch(s[i], groupArray[8]);

}

}

}

regfree(&regexCompiled);

return 0;

}