**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 |  | Синицкая Д.В. |
| Преподаватель |  | Государкин Я. С. |

Санкт-Петербург

2024

## Цель работы

Изучение и применение функций из библиотеки regex.h языка программирования Cи для обнаружения совпадений в строках при помощи использования регулярных выражений. Получение умений создания регулярных выражений в Cи и успешное их применение для поиска определенных шаблонов в тексте.

## Задание

Вариант 1. 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

В программе использованны метод Print\_Correct\_URL(char\* s, regmatch\_t groupArray) и функция main().

Метод Print\_Correct\_URL(char\* s, regmatch\_t groupArray) предназначен для печати соответствующего URL.

В функции main() создается переменная size\_t maxGroups отвечающая за максимальное количество групп в регулярном выражении, создается строка char\* regexString, содержащая регулярное выражение, происходит компиляция регулярного выражения, далее в цикле while (fgets(s, 100, stdin)) чтение ввода с клавиатуры пока есть ввод происходит проверка соответствие строки регулярному выражению, если строка соответсвует - печать соответствующего URL. После цикла освобождается память, выделенная под компилированное регулярное выражение

Разработанный программный код см. в приложении А.

## Выводы

В процессе выполнения лабораторной работы мной были освоены навыки необходимые для использования регулярных выражений на языке Cи при помощи библиотеки regex.h, а также умения создавать регулярные выражения в соответствии с требованиями. Были изучены необходимые языковые конструкции и особенности написания регулярных выражений на языке Cи.

# Приложение А Исходный код программы

#include <stdio.h>

#include <string.h>

#include <regex.h>

//метод печати соответствующего URL

void Print\_Correct\_URL(char\* s, regmatch\_t groupArray) {

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

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

}

int main() {

//максимальное количество групп в регулярном выражении

size\_t maxGroups = 10;

//регулярное выражение для проверки URL

char\* regexString = "([a-zA-Z0-9]+:\\/\\/)?(www\\.)?([a-zA-Z\_0-9\\-]+(\\.[a-zA-Z0-9\_\\-]+)+)((\\/[a-zA-Z0-9\_\\-]+)+)?(\\/([a-zA-Z0-9\_\\-]+\\.[a-zA-Z0-9\_\\-]+\n))";

//компилиция регулярного выражения

regex\_t regexCompiled;

regmatch\_t groupArray[maxGroups];

regcomp(&regexCompiled, regexString, REG\_EXTENDED);

char s[100];

//чтение ввода с клавиатуры пока есть ввод

while (fgets(s, 100, stdin)) {

//проверка соответствие строки регулярному выражению

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

//печать соответствующего URL

Print\_Correct\_URL(s, groupArray[3]);

printf(" - ");

Print\_Correct\_URL(s, groupArray[8]);

}

}

//освобождение памяти, выделенной под компилированное регулярное выражение

regfree(&regexCompiled);

return 0;

}