**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. 3344 |  | Валиев Р.А. |
| Преподаватель |  | Глазунов С.А. |

Санкт-Петербург

2024

## Цель работы

Изучить принцип работы с регулярными выражениями и их применение в языке программирования С.

## 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](https://e.moevm.info/mod/lesson/view.php?id=509), 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

Подключаются заголовочные файлы *<stdio.h>*, *<stdlib.h>*, *<string.h>*, *<regex.h>.*

**Основные шаги выполнения работы:**

1. Компиляция регулярного выражения:

Программа начинается с компиляции регулярного выражения ((w\+\\.)+\\w+)(.\*)(\\/\\w+\\.\\w+), которое используется для извлечения доменного имени и конечного пути из URL.

2. Чтение ввода:

Программа затем читает строки из стандартного ввода до тех пор, пока не встретит строку, содержащую "Fin.".

3. Поиск и вывод результата:

Для каждой введенной строки программа проверяет, есть ли в ней совпадения с регулярным выражением.

**Вывод результатов:** Извлеченное доменное имя и конечный путь затем выводятся на экран.

**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  domain google.com.edu/hello.q  ftp://skype.com/qqwe/qweqw/qwe.avi  Fin. | google.com - track.mp3  google.com.edu - hello.q  skype.com - qwe.avi | Верный ответ. |
| 2. | test google..com/hello.mp23  pr http://www.google.com//qwe.exe  fw wf fds://google.com/test.exe | google.com - test.exe | Верный ответ (первые два не подходят) |

## Выводы

Было изучено и практически применено использование регулярных выражений в программировании.

Была разработана программа для поиска и выделения определенных частей URL – адресов с использованием регулярных выражений. Это показало, что регулярные выражения могут использоваться для решения широкого спектра задач, связанных с обработкой текста и данных.

# Приложение А Исходный код программы

Название файла: Valiev\_Ruslan\_lb\_1.c

#include <stdio.h>  
#include <regex.h>  
#include <string.h>  
int main (){  
 char \* regexString = "((\\w+\\.)+\\w+)(.\*)(\\/\\w+\\.\\w+)";  
 size\_t maxGroups = 5;  
 regex\_t regexCompiled;  
 regmatch\_t groupArray[maxGroups];  
 if (regcomp(&regexCompiled, regexString, REG\_EXTENDED)){return 0;};  
 char s[1000];  
 while (fgets(s,1000,stdin)){  
 if (strstr(s, "Fin.") != NULL){break; printf("\n");}  
 if (regexec(&regexCompiled, s, maxGroups, groupArray, 0) == 0) {  
 char url[1000];  
 char filename[1000];  
 strncpy(url, s + groupArray[1].rm\_so, groupArray[1].rm\_eo - groupArray[1].rm\_so);  
 url[groupArray[1].rm\_eo - groupArray[1].rm\_so] = '\0';  
 char \*wwwSubstr = "www.";  
 char \*wwwPos = strstr(url, wwwSubstr);  
 if (wwwPos == url) {  
 memmove(url, url + strlen(wwwSubstr), strlen(url) - strlen(wwwSubstr) + 1);  
 }  
 strncpy(filename, s + groupArray[4].rm\_so + 1, groupArray[4].rm\_eo - groupArray[4].rm\_so - 1);  
 filename[groupArray[4].rm\_eo - groupArray[4].rm\_so - 1] = '\0';  
 printf("\n%s - %s", url, filename);  
 }  
 }  
 regfree(&regexCompiled);  
 return 0;  
}