**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

отчет

**по лабораторной работе №1**

**по дисциплине «Программирование»**

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

|  |  |  |
| --- | --- | --- |
| Студент гр. 3343 |  | Иванов П.Д. |
| Преподаватель |  | Государкин Я. С. |

Санкт-Петербург

2024

## Цель работы

Научиться использовать регулярные выражения в библиотеке regex.h, написав программу 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

Программа построчно считывает текст. Для каждого предложения она применяет шаблон регулярных предложений.

В *rattern* хранится регулярное выражение, по которому отбираются строчки. С помощью *regcomp()* в *regex\_compiled* сохраняется скомпилированное регулярное выражение, которое можно использовать далее.

В случае, если строка содержит регулярное выражение, то вызывается функция *print\_group()* которая принимает массив *matched\_group* и исходную строку. Далее эта функция печатает необходимые захватываемые группы.

Разработанный программный код см. в приложении А.

## Тестирование

Таблица 1.

|  |  |  |  |
| --- | --- | --- | --- |
| № | 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 | Верный вывод |

## Выводы

В ходе выполнения работы, была изучена библиотека regex.h, необходимая для написания регулярных выражений на языке Си.

# Приложение А Исходный код программы

#include <stdio.h>  
#include <regex.h>  
  
void print\_group(char\* str, regmatch\_t matched\_group){  
 for (int i = matched\_group.rm\_so; i < matched\_group.rm\_eo; ++i) {  
 printf("%c", str[i]);  
 }  
}  
  
int main(){  
 char pattern[] = "([A-z]+:\\/\\/)?(www\\.)?([A-z-]+\\.[A-z-]+(\\.[A-z\_-]+)\*)\\/([A-z]+\\/)\*([A-z]+\\.[A-z0-9]+\n?)"; //3 //6  
 int capturing\_groups = 7;  
  
 regex\_t regex\_compiled;  
 regmatch\_t matched\_groups[capturing\_groups];  
  
 if (regcomp(&regex\_compiled, pattern, REG\_EXTENDED)){  
 printf("pattern error");  
 return 0;  
 }  
  
 char str[100];  
 while (fgets(str, 100, stdin)){  
 if (regexec(&regex\_compiled, str, capturing\_groups, matched\_groups, 0) == 0){  
 print\_group(str, matched\_groups[3]);  
 printf(" - ");  
 print\_group(str, matched\_groups[6]);  
 }  
 }  
  
 regfree(&regex\_compiled);  
 return 0;  
}