

**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 строки. 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.	<p>This is simple url:</p> <p><a href="http://www.google.com/track.mp3">http://www.google.com/track.mp3</a></p> <p>May be more than one upper level domain</p> <p><a href="http://www.google.com.edu/hello.avi">http://www.google.com.edu/hello.avi</a></p> <p>Many of them.</p> <p>Rly. Look at this!</p> <p><a href="http://www.qwe.edu.etu.yahooo.org.net.ru/qwe.q">http://www.qwe.edu.etu.yahooo.org.net.ru/qwe.q</a></p> <p>Some other protocols</p> <p><a href="ftp://skype.com/qqwe/qweqw/qwe.avi">ftp://skype.com/qqwe/qweqw/qwe.avi</a></p> <p>i</p> <p>Fin.</p>	<p>google.com - track.mp3</p> <p>google.com.edu - hello.avi</p> <p>qwe.edu.etu.yahooo.org.net.ru - qwe.q</p> <p>skype.com - qwe.avi</p>	<p>Верный вывод</p>

## **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;
```

}