

**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. 3341

Игнатьев К.А.

Преподаватель

Глазунов С.А.

Санкт-Петербург

2024

## **Цель работы**

Цель работы - освоение работы с регулярными выражениями и их применение с помощью программы на языке Си

Для достижения цели требуется решить следующие задачи:

- Ознакомиться с регулярными выражениями;
- Написать программу, использующее регулярное выражение для решения задачи.

## 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

Перед написанием кода программы было создано регулярное выражение, которое ищет в тексте все ссылки по шаблону:

```
([a-z]+:\W)?(www\.)?((\w+\.)?(\w+\.)*\w+)\W((\w+\W)?(\w+\.\w+))
```

В данном регулярном выражении каждая часть ссылки выделена в отдельную группу, а 3-я и 8-я группы содержат в себе название сайта и имя файла соответственно.

Далее была написана программа, использующая указанное выше регулярное выражение.

Для работы с регулярными выражениями подключается библиотека `<regex.h>`.

В функции `main()` инициализируются:

- массив символов *line* для дальнейшего заполнения им строками
- *str\_pat*, содержащее в себе регулярное выражение
- *regex\_compiled* для компиляции регулярного выражения
- массив *groups* для хранения индексов начала и конца групп

Далее в цикле `do {} while()`, работающем до тех пор, пока не появится маркер конца текста, считываются символы строки с помощью функции `getline()`, и, если в предложении будет найдена ссылка, соответствующая шаблону, то она будет выведена на экран с помощью `print()`. Память скомпилированного регулярного выражения очищается с помощью `regfree()`.

Для получения строки была написана функция `getline()`, которая, с помощью цикла `for()` будет считывать символы строки через `getchar()`. Так же было написано условие выхода из цикла, в случае встречи символа переноса строки. В конце ставится символ конца строки.

Для вывода результата на экран была реализована функция `print()`, которая принимает на вход строку с ссылкой *link* и массив групп *groups*. С помощью циклов `for` посимвольно выводятся части ссылок, содержащие название сайта и название файла.

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.	Fin.		Ничего не выведено, как и должно быть.
2.	This is simple url: http://www.google.com/ track.mp3 May be more than one upper level domain http://www.google.com.e du/hello.avi Many of them. Rly. Look at this! http:// www.qwe.edu.etu.yahoo o.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	Ссылки выведены
3.	<a href="https://se.moevm.info/doku.php">https://se.moevm.info/ doku.php</a> Fin.	se.moevm.info doku.php	- Ссылки выведены
4.	Url Fin.		Ничего не выведено, как и должно быть

## **Выводы**

В результате выполнения работы была освоена работа с регулярными выражениями, выполнены поставленные задачи, а именно: были изучены основные конструкции регулярных выражений, написано регулярное выражение, позволяющее искать ссылку в тексте, а также написана программа на языке си, использующее заранее подготовленное регулярное выражение для вывода на экран названия сайта и имени файла из ссылок, содержащихся в тексте.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <regex.h>

void getLine(char* line) {
    int i=0;
    for (i = 0; i < 100; ++i) {
        line[i]=getchar();
        if (line[i]==10){
            break;
        }
    }
    line[i]='\0';
    i=0;
}

void print(char* link, regmatch_t* groups){
    for (int i = groups[3].rm_so; i < groups[3].rm_eo; ++i) {
        printf("%c", link[i]);
    }
    printf(" - ");
    for (int i = groups[8].rm_so; i < groups[8].rm_eo; ++i) {
        printf("%c", link[i]);
    }
    puts("");
}

int main(){
    char* line= malloc(sizeof(char)*100);
    char* str_pat="([a-z]+:\\\\\/\\\/)?(www\\.)?((\\w+\\.)(\\w+\\.)*\\w+)\\\/((\\w+\\\/)*)?(\\w+\\.\\w+)";
    regex_t regex_compiled;
    regcomp(&regex_compiled, str_pat, REG_EXTENDED);
    regmatch_t groups[9];
    do{
        getLine(line);
        //fgets(line, 100, stdin);
        //printf("----\nline - %s\n----\n", line);
        if (regexexec(&regex_compiled, line, 9, groups, 0) == 0)
            print(line, groups);

    }while (strncmp(line, "Fin.", strlen("Fin."))!=0);
    regfree(&regex_compiled);
    free(line);

    return 0;
}
```