

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. 3342

Лучкин М.А.

Преподаватель

Глазунов С.А.

Санкт-Петербург

2024

Цель работы

Целью работы является изучение принципов работы регулярных выражений и использование их в программе на языке C.

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

Этот код на языке программирования C использует библиотеку `regex.h` для работы с регулярными выражениями. В нем компилируется регулярное выражение, которое ищет определенный шаблон вводимых строк (URL-адресов).

В цикле программа считывает строки из стандартного ввода до тех пор, пока введенная строка не равна "Fin.". Затем, она проверяет каждую строку на соответствие заданному регулярному выражению. Если строка соответствует шаблону, то она извлекает определенные группы из регулярного выражения и выводит их на экран.

В данном случае код извлекает и выводит домен и путь из URL-адресов, введенных пользователем.

Программа выводит подходящие ссылки в формате `<название_сайта> - <имя_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
1.	<p>This is simple url:</p> <p>http://www.google.com/track.mp3</p> <p>May be more than one upper level</p> <p>domain http://www.google.com.edu/hello.avi</p> <p>Many of them.</p> <p>Rly. Look at this!</p> <p>http://www.qwe.edu.etu.yahooo.org.net.ru/qwe.q</p> <p>Some other protocols</p> <p>ftp://skype.com/qqwe/qweqw/qwe.avi</p> <p>Fin.</p>	<p>google.com - track.mp3</p> <p>google.com.edu - hello.avi</p> <p>qwe.edu.etu.yahooo.org.net.</p> <p>ru - qwe.q</p> <p>skype.com - qwe.avi</p>

Выводы

Было написано регулярное выражение, изучены способы работы с ним и с группами захвата в языке программирования С.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.c

```
#include <stdio.h>
#include <string.h>
#include <regex.h>
#define STRING_SIZE 100

int main(){
    regex_t reg;
    regmatch_t groups[7];
    char string[STRING_SIZE];
    regcomp(&reg, "(.*:\\\\/*)*(w*\\.)*(([a-z]+\\.)+[a-z]+)\\\\/(.*\\\\/)*(.*\\.\\.*)\\n", REG_EXTENDED);

    while (strcmp(string, "Fin.") != 0){
        fgets(string, STRING_SIZE, stdin);

        if (!regexexec(&reg, string, 7, groups, 0)){
            for (int i = groups[3].rm_so; i < groups[3].rm_eo; i++){
                printf("%c", string[i]);
            }

            printf(" - ");
            for (int j = groups[6].rm_so; j < groups[6].rm_eo; j++){
                printf("%c", string[j]);
            }

            printf("%c", '\\n');
        }
    }
    regfree(&reg);

    return 0;
}
```