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 _____ Кузнецов Р.А.

Преподаватель _____ Глазунов С.А.

Санкт-Петербург
2024

Цель работы

Изучение работы регулярных выражений и получение навыков работы с регулярными выражениями в языке программирования Си.

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

Подключаются библиотеки stdio, string, stdlib и regex. Инициализируется как переменная для хранения текста, так и переменная для хранения регулярного выражения. Выражение разбивается на группы для облегчения дальнейшей работы. Создаются переменные для представления скомпилированного регулярного выражения, максимального количества групп в выражении и массив для хранения информации о каждой группе в регулярном выражении. Компилируется шаблон выражения в regex. Считывается при помощи бесконечного цикла весь текст функцией fgets и сохраняется в переменную text. newline используется для нахождения переноса строки, чтобы потом заменить его на конец строки. Следом идет проверка на конец текста при помощи функции strcmp. Функция regexec определяет есть ли подходящее выражение в тексте под шаблон регулярного выражения, если есть то функция возвращает 0. На экран выводится домен и файл при помощи групп. Используется специальная строка формата „%.*s“, что обозначает необходимость двух аргументов. Первый аргумент ширина необходимой строки, которая обозначается индексами начала и конца найденной строки rm_eo и rm_so, второй аргумент же это указатель на найденную строку. Освобождается память выделенная для регулярного выражения и программа завершается.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Коммен тарии
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	-

Выводы

Была изучена работа регулярных выражения в языке программирования Си, получены навыки работы с ними.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main_lb1.c

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

int main(){
    char* text = (char*)malloc(sizeof(char) * 10000);
    char* pattern = "((http|https|ftp):\\/\\/)?"  // 1-2
              "(www\\.)?" //  3
              "([a-zA-Z0-9]+(\\.[a-zA-Z0-9]+)+)" //  4-5
              "(\\/[-a-zA-Z0-9:%_\\+.~#?&//=]*)?" // 6
              "\\/([^\\/]+\\.[a-zA-Z0-9]+)"; // 7

    regex_t regex;
    size_t max_group = 8;
    regmatch_t groupArray[max_group];
    regcomp(&regex, pattern, REG_EXTENDED);

    while(1) {
      fgets(text, 10000, stdin);
      char *newline = strchr(text, '\n');

      if (newline) *newline = '\0';

      if (strcmp(text, "Fin.") == 0) break;

      if(regexec(&regex, text, max_group, groupArray, 0) == 0){
           printf("%.*s - %.*s\n",
                (int)(groupArray[4].rm_eo     -     groupArray[4].rm_so),
&text[groupArray[4].rm_so],
                (int)(groupArray[7].rm_eo     -     groupArray[7].rm_so),
&text[groupArray[7].rm_so]);
      }
      }
    regfree(&regex);
    return 0;
}
```