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.3343 _____ Волох И.О.

Преподаватель _____ Государкин Я.С.

Санкт-Петербург
2024

Цель работы.

Изучение и применение на практике регулярных выражений в 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.

Написанная программа (см. приложение А) выполняет поиск ссылок на различные файлы в сети интернет. Она проверяет введённые с клавиатуры строки на соответствие регулярному выражению, если строка соответствует то она сохраняет данные в структуру Bigger_Info, после чего происходит вывод в формате <название_сайта> - <имя_файла>.

Переменные:

- MAX_RAZ – длина строки
- char gotten[MAX_RAZ] – строка введённая с клавиатуры
- regex_t regor – структура в которой хранится скомпилированное регулярное выражение
- regmatch_t groups[] – структура в которой хранятся адреса групп

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	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	

Выводы.

В данной лабораторной работе было изучено и применено на практике написание регулярных выражений в языке Си. И были изучены функции из библиотеки regex.h требуемые для выполнения задания.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.c

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

#define MAX_RAZ 250

struct Bigger_Info {
   char *adress;
   char *file;
};

int main() {
   regex_t regor;
   int rep;
   char gotten[MAX_RAZ];
   struct Bigger_Info *whole_data = NULL;
   int count = 0;
   regmatch_t groups[7];
   rep = regcomp(&regor, "(http://)?(www\\.)?(([A-Za-z0-9\\-]+\\.)+[A-Za-z0-9\\-]+)\\/([A-Za-z0-9\\/\\-_]+\\/)*([A-Za-z0-9_\\-]+\\.[A-Za-z0-9]+)", REG_EXTENDED);
   while (1) {
      fgets(gotten, MAX_RAZ - 1, stdin);
      if (strstr(gotten, "Fin.") != NULL) {
         break;
      }
      rep = regexec(&regor, gotten, 7, groups, 0);
      if (rep == 0) {
         whole_data = realloc(whole_data, (count + 1) * sizeof(struct Bigger_Info));
         whole_data[count].adress = strndup(gotten + groups[3].rm_so, groups[3].rm_eo - groups[3].rm_so);
         whole_data[count].file = strndup(gotten + groups[6].rm_so, groups[6].rm_eo - groups[6].rm_so);
         count++;
      }
   }
   for (int i = 0; i < count; i++) {
      printf("%s - %s\n", whole_data[i].adress,  whole_data[i].file);
      free(whole_data[i].adress);
      free(whole_data[i].file);
   }
   free(whole_data);
   regfree(&regor);
   return 0;
}
```