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

Студентка гр. 9381 _____ Калинина Е. Н.

Преподаватель _____ Берленко Т.А.

Санкт-Петербург

2020

Цель работы.
Обработка текста с помощью регулярных выражений и языка 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.

С помощью цикла do-while считывается текст. Регулярное выражение хранится в переменной regexString. Поиск соответствий в тексте с регулярным выражением происходит в помощью специальных функций. Сначала, с помощью regcomp регулярное выражение компилируется, далее, с помощью скомпилированного буферного шаблона регулярного выражения и функции regexec производится поиск совпадений в каждом предложении по отдельности. При их наличии — выводится нужная информация, а точнее пары <название_сайта> - <имя_файла>. Далее с помощью free и regfree выделенная ранее память очищается.

Тестирование.

Результаты тестирования представлены в таблице.

№	Входные данные	Выходные данные
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
2.	skype.com/qqwe/qweqw/qwe.avi skype.com/qwq.info s jsldkf js s sf www.skype.com/qwq.info dsfsd s sf sdf ftp://skype.com/qwq.info Fin.	skype.com - qwe.avi skype.com - qwq.info skype.com - qwq.info skype.com - qwq.info

Выводы.

Во время выполнения лабораторной работы была проведена работа с текстом и регулярными выражениями с помощью языка Си.

ПРИЛОЖЕНИЕ 1
ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.c

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


int main(){

        char * regexString = "([a-z]+\\:\\/{2})?(www\\.)?((\\w+\\.)+\\w+)\\/
(\\w*\\/)*(\\w+\\.\\w+)$";
    regex_t regexCompiled;
    size_t maxGroups = 7;
    regmatch_t groupArray[maxGroups];
    regcomp(&regexCompiled, regexString, REG_EXTENDED);

    char **arr = malloc(10*sizeof(char*));
    int Count_sentence = 0; //предложение №
    int size = 10; //память в arr
    char current_symbol;
    do{
       int Count_symbol = 0; //символ в предложении
       int size_sentence = 10; //память в предложении
       arr[Count_sentence] = malloc(size_sentence*sizeof(char));
       current_symbol = getchar();
       do{
          if (Count_symbol == size_sentence - 2 ){
             size_sentence += 10;
             arr[Count_sentence] = realloc(arr[Count_sentence],size_sentence);
          }
          arr[Count_sentence][Count_symbol] = current_symbol;
          Count_symbol++;
          if (Count_symbol == 4){
             arr[Count_sentence][Count_symbol] = '\0';
             if (!strcmp(arr[Count_sentence],"Fin."))
                break;
          }
          current_symbol = getchar();
       }while (current_symbol != '\n');
       arr[Count_sentence][Count_symbol] = '\0';
```

```
        if (Count_sentence == size - 2 ){
            size += 10;
            arr = realloc(arr, size * sizeof(char*));
        }
        Count_sentence++;
    }while (strcmp(arr[Count_sentence-1],"Fin."));
    int k = 0;
    while (k < Count_sentence){
        if (regexec(&regexCompiled, arr[k], maxGroups, groupArray, 0) == 0){
            for(int j = groupArray[3].rm_so; j < groupArray[3].rm_eo; j++)
                printf("%c", arr[k][j]);
            printf(" - ");
            for(int i = groupArray[6].rm_so; i < groupArray[6].rm_eo; i++)
                printf("%c", arr[k][i]);
        }
        printf("\n");
        k++;
    }
    for (int i = 0; i < Count_sentence; i++){
        free(arr[i]);
    }
    free(arr);
    regfree(&regexCompiled);

    return 0;
}
```