

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ**

**ОТЧЕТ
по лабораторной работе №1
по дисциплине «Программирование»
Тема: Лабораторная работа № 1. Регулярные выражения**

Студентка гр. 3343

Синицкая Д.В.

Преподаватель

Государкин Я. С.

Санкт-Петербург

2024

Цель работы

Изучение и применение функций из библиотеки `regex.h` языка программирования Си для обнаружения совпадений в строках при помощи использования регулярных выражений. Получение умений создания регулярных выражений в Си и успешное их применение для поиска определенных шаблонов в тексте.

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

В программе использованы метод `Print_Correct_URL(char* s, regmatch_t groupArray)` и функция `main()`.

Метод `Print_Correct_URL(char* s, regmatch_t groupArray)` предназначен для печати соответствующего URL.

В функции `main()` создается переменная `size_t maxGroups` отвечающая за максимальное количество групп в регулярном выражении, создается строка `char* regexString`, содержащая регулярное выражение, происходит компиляция регулярного выражения, далее в цикле `while (fgets(s, 100, stdin))` чтение ввода с клавиатуры пока есть ввод происходит проверка соответствие строки регулярному выражению, если строка соответствует - печать соответствующего URL. После цикла освобождается память, выделенная под компилированное регулярное выражение

Разработанный программный код см. в приложении А.

Выводы

В процессе выполнения лабораторной работы мной были освоены навыки необходимые для использования регулярных выражений на языке Си при помощи библиотеки `regex.h`, а также умения создавать регулярные выражения в соответствии с требованиями. Были изучены необходимые языковые конструкции и особенности написания регулярных выражений на языке Си.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

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

//метод печати соответствующего URL
void Print_Correct_URL(char* s, regmatch_t groupArray) {
    for(int i = groupArray.rm_so; i < groupArray.rm_eo; i++)
        printf("%c", s[i]);
}

int main() {
    //максимальное количество групп в регулярном выражении
    size_t maxGroups = 10;
    //регулярное выражение для проверки URL
    char* regexString = "([a-zA-Z0-9]+:\\\\/\\\\/)?(www\\\\.)?([a-zA-Z0-9\\\\-]+(\\\\. [a-zA-Z0-9\\\\-]+)+)((\\\\/[a-zA-Z0-9\\\\-]+)+)?(\\\\/([a-zA-Z0-9\\\\-]+\\\\. [a-zA-Z0-9\\\\-]+\\\\n))";
    //компиляция регулярного выражения
    regex_t regexCompiled;
    regmatch_t groupArray[maxGroups];
    regcomp(&regexCompiled, regexString, REG_EXTENDED);

    char s[100];
    //чтение ввода с клавиатуры пока есть ввод
    while (fgets(s, 100, stdin)) {
        //проверка соответствие строки регулярному выражению
        if (regexec(&regexCompiled, s, maxGroups, groupArray, 0) ==
0) {
            //печать соответствующего URL
            Print_Correct_URL(s, groupArray[3]);
            printf(" - ");
            Print_Correct_URL(s, groupArray[8]);
        }
    }
}
```

```
        //освобождение памяти, выделенной под скомпилированное
регулярное выражение
    regfree(&regexCompiled);
    return 0;
}
```