**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. 3341 |  | Пчелкин Н.И, |
| Преподаватель |  | Глазунов С.А. |

Санкт-Петербург

2024

## Цель работы

Целью работы является освоение работы с регулярными выражениями на языке C.

Для достижения поставленной цели требуется решить следующие задачи:

* ознакомиться с регулярными выражениями;
* научиться их использовать;
* написать программу, решающую задачу в соответствии с индивидуальным условием с использованием регулярных выражений.

## Задание

1 вариант.

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](https://e.moevm.info/mod/lesson/view.php?id=509), 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

Используемые переменные:

* макрос *BLOCK -* максимальная длина строки
* *text* – текущая строка
* *regexString -* строка с регулярным выражением
* *regex\_compiled –* переменнаяхранит скомпилированное регулярное выражение
* *group\_array –* массив групп захвата

Регулярное выражение "([a-z]+:\\/\\/)?(www\\.)?(([a-z]+\\.)+[a-z]+)\\/([a-z]+\\/)\*([a-z]+\\.[a-z0-9]+)" начинается с протокола, который может быть, а может и не быть (поэтому после группы стоит знак вопроса). Затем может стоять “www.”, за ним группа – доменное имя (которое нас интересует по условию), состоящее из одного домена и одного или нескольких доменов более верхнего уровня, выделяем их в одну группу. Следующая группа – некоторый путь к файлу, и последняя группа – файл с некоторым расширением.

Функции:

* *void search* принимает указатель на текущую строку, проверяет, подходит ли строка по заданной маске и после чего выводит необходимые нам группы через дефис, в конце каждой строки – ставит символ переноса строки.

Обработка всего текста происходит по строкам. Обрабатываются они в цикле с предусловием, где условием выхода из цикла является последнее предложение "Fin.". Каждая строка обрабатывается вышеописанной функцией search.

Обработка происходит благодаря библиотеке regex.h, с помощью которой мы и можем создать маску (регулярное выражение).

Разработанный программный код см. в приложении А.

## Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Входные данные | Выходные данные | Комментарии |
|  | 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 | Предложенный авторами задачи тест |
|  | google.com/bububu.bebebe  www.google.com/ooomaagaa/bebe.bubu  Fin. | google.com - bububu.bebebe  google.com - bebe.bubu | В стандартном тесте упустили URL без протокола. Поэтому делаем это в отдельном тесте |

## Выводы

Была освоена работа с регулярными выражениями на языке C.

Для достижения поставленной цели были решены следующие задачи:

* ознакомление с регулярными выражениями;
* их использование;
* написана программа, которая, используя регулярные выражения, находит только примеры команд в оболочке суперпользователя и выводит на экран пары <имя пользователя> - <имя команды>.

# Приложение А Исходный код программы

Название файла: main.c

#include <stdio.h>

#include <stdlib.h>

#include <regex.h>

#include <string.h>

#define BLOCK 100

void search(char\* text, regex\_t regexCompiled, size\_t maxGroups, regmatch\_t groupArray[]){

if (regexec(&regexCompiled, text, maxGroups, groupArray, 0) == 0){

for(int k = groupArray[3].rm\_so; k < groupArray[3].rm\_eo; k++) printf("%c", text[k]);

printf(" - ");

for(int k = groupArray[6].rm\_so; k < groupArray[6].rm\_eo; k++) printf("%c", text[k]);

printf("\n");

}

}

int main(){

char text[BLOCK];

char\* regexString = "([a-z]+:\\/\\/)?(www\\.)?(([a-z]+\\.)+[a-z]+)\\/([a-z]+\\/)\*([a-z]+\\.[a-z0-9]+)";

regex\_t regexCompiled;

size\_t maxGroups = 7;

regcomp(&regexCompiled, regexString, REG\_EXTENDED);

regmatch\_t groupArray[maxGroups];

while(fgets(text, BLOCK, stdin)){

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

search(text, regexCompiled, maxGroups, groupArray);

}

return 0;

} }