

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

Преподаватель

Пчелкин Н.И,

Глазунов С.А.

Санкт-Петербург

2024

Цель работы

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

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, 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* – переменная хранит скомпилированное регулярное выражение
- *group_array* – массив групп захвата

Регулярное выражение "`\"([a-z]+:\\\\V)?(www\\\\.)?((\\[a-z]+\\\\.)+[a-z]+)\\\\V(\\[a-z]+\\\\V)*\\([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
1.	<p>This is simple url:</p> <p>http://www.google.com/trac</p> <p>k.mp3</p> <p>May be more than one upper level domain</p> <p>http://www.google.com.edu/hello.avi</p> <p>Many of them.</p> <p>Rly. Look at this!</p> <p>http://</p> <p>www.qwe.edu.etu.yahooo.org.net.ru/qwe.q</p> <p>Some other protocols</p> <p>ftp://skype.com/qqwe/qweqw/qwe.avi</p> <p>Fin.</p>	<p>google.com - track.mp3</p> <p>google.com.edu</p> <p>hello.avi</p> <p>qwe.edu.etu.yahooo.org.net.ru - qwe.q</p> <p>skype.com - qwe.avi</p>	Предложенный авторами задачи тест
2.	<p>google.com/bububu.bebebe</p> <p>www.google.com/ooomaagaa/bebe.bubu</p> <p>Fin.</p>	<p>google.com</p> <p>bububu.bebebe</p> <p>google.com - bebe.bubu</p>	- В стандартном тесте упустили URL без протокола. Поэтому делаем это в отдельном тесте

Выводы

Была освоена работа с регулярными выражениями на языке С.

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 (regexexec(&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;
} }
```