

**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. 3344

Коршунов П.И.

Преподаватель

Глазунов С.А.

Санкт-Петербург

2024

## **Цель работы**

Освоение работы с регулярными выражениями на языке Си на примере использующей их программы.

### **Задание.**

Вариант 2. На вход программе подается текст, представляющий собой набор предложений с новой строки. Текст заканчивается предложением "Fin." В тексте могут встречаться примеры запуска программ в командной строке Linux. Требуется, используя регулярные выражения, найти только примеры команд в оболочке суперпользователя и вывести на экран пары <имя пользователя> - <имя\_команды>. Если предложение содержит какой-то пример команды, то гарантируется, что после нее будет символ переноса строки.

Примеры имеют следующий вид:

Сначала идет имя пользователя, состоящее из букв, цифр и символа \_

Символ @

Имя компьютера, состоящее из букв, цифр, символов \_ и -

Символ : и ~

Символ \$, если команда запущена в оболочке пользователя и #, если в оболочке суперпользователя. При этом между двоеточием, тильдой и \$ или # могут быть пробелы.

Пробел

Сама команда и символ переноса строки.

## Выполнение работы

Была подключена стандартная библиотека для ввода и вывода `<stdio.h>`, стандартная библиотека `<stdlib.h>`, библиотека для работы со строками `<string.h>` и библиотека для работы с регулярными выражениями `<regex.h>`. Была проинициализирована строка, отвечающая за регулярное выражение `char *regexString = "(\\w+)@([A-Za-z0-9_-]+)(:\\|s?~\\|s?#\\|s)(.+) "` и создана переменная, отвечающая за максимальное число групп для поиска в регулярном выражении `size_t maxGroups = 5`. Была объявлена структура для хранения информации о скомпилированном регулярном выражении `regex_t regexCompiled`. Был объявлен массив `regmatch_t groupArray[maxGroups]` размером `maxGroups`, который будет использоваться для хранения информации о совпадениях групп в регулярном выражении. Была вызвана функция для компиляции регулярного выражения с проверкой на успех `if (regcomp(&regexCompiled, regexString, REG_EXTENDED))`. Была выделена память под строку для входных данных `char* s`. Был запущен цикл, который считывал входные данные, пока строка на вход не равнялась "Fin\n" `while (fgets(s, 1000, stdin) && strcmp(s, "Fin.\n") != 0)`. В `if` была вызвана функция `if (regexexec(&regexCompiled, s, maxGroups, groupArray, 0) == 0)`, которая проверяет, соответствует ли строка `s` регулярному выражению. Если соответствие найдено, функция возвращает 0, иначе - ненулевое значение. Далее были запущены циклы, которые выводят символы из первой и четвертой группы регулярного выражения. После цикла `while` высвобождалась память под строку для входных данных и структуру для скомпилированного регулярного выражения. После чего программа завершалась.

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>Run docker container: kot@kot-ThinkPad:~\$ docker run -d --name stepik stepik/challenge-avr:latest You can get into running /bin/bash command in interactive mode: kot@kot-ThinkPad:~\$ docker exec -it stepik "/bin/bash" Switch user: su : root@84628200cd19: ~ # su box box@84628200cd19: ~ \$ ^C Exit from box: box@5718c87efaa7: ~ \$ exit exit from container: root@5718c87efaa7: ~ # exit kot@kot-ThinkPad:~\$ ^C Fin.</p>	<p>root - su box root - exit</p>	-
2.	<p>Switch user: su : roofsafast@8dasd4628200cd19 : ~ # d dadad ad a box@84628200cd19: ~ \$ ^C Exit from box: box@5718c87efaa7: ~ \$ exit exit from container: root@5718c87efaa7: ~ # exit kot@kot-ThinkPad:~\$ ^C Fin. root@da__da:~# su bsax Fin.</p>	<p>roofsafast - d dadad ad a root - exit root - su bsax</p>	-

## **Выводы**

Была освоена работа с регулярными выражениями на языке Си на примере использующей их программы.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: Korshunov\_Petr\_lb1.c

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

int main ()
{
    char * regexString = "(\\w+)@([A-Za-z0-9_-]+)(:\\s?~\\s?#\\s) (.+)";
    size_t maxGroups = 5;

    regex_t regexCompiled;
    regmatch_t groupArray[maxGroups];

    if (regcomp(&regexCompiled, regexString, REG_EXTENDED))
    {
        printf("Can't compile regular expression\n");
        return 0;
    };

    char *s = (char*)malloc(1000 * sizeof(char));

    while (fgets(s, 1000, stdin) && strcmp(s, "Fin.\n") != 0){
        if (regexexec(&regexCompiled, s, maxGroups, groupArray, 0) == 0)
        {
            for(int j=groupArray[1].rm_so;j<groupArray[1].rm_eo;j++)
                printf("%c", s[j]);
            printf(" - ");
            for(int j=groupArray[4].rm_so;j<groupArray[4].rm_eo;j++)
                printf("%c", s[j]);
        }
    }

    free(s);
    regfree(&regexCompiled);

    return 0;
}
```