

**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. 3342

Легалов В. В.

Преподаватель

Глазунов С.А.

Санкт-Петербург

2024

### **Цель работы**

Изучить работу и применение регулярных выражений при работе со строками в языке Си.

## 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 идёт 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

В начале выполнения алгоритма инициализируется буффер для хранения строк считанных из входного потока, структура, содержащая регулярное выражение, используемое в решении задачи, в пригодном для применения формате и массив для хранения местоположения совпадений строки с регулярным выражением.

В функции *print\_name\_and\_command* выполняется построчное считывание входных данных с помощью *fgets* и сравнение полученных строк с регулярным выражением с помощью *regexes*. В случае обнаружения совпадения выводятся имя пользователя и введённая команда, указанные в рассматриваемой строке с помощью функции *print\_group*.

Функция *print\_group* используется для вывода групп символов совпадений строки с регулярным выражением.

Разработанный программный код см. в приложении А.

## **Выводы**

Были изучены и применены на практике регулярные выражения в языке программирования Си. Была написана программа, определяющая среди полученных строк примеров команд, команды в оболочке суперпользователя и выводящая имя пользователя и введённую команду.

## Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные
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: ~ \$ Exit from box: box@5718c87efaa7: ~ \$ exit exit from container: root@5718c87efaa7: ~ # exit kot@kot-ThinkPad:~\$ Fin.</p>	<p>root – su box  root - exit</p>

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.c

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

#define REGEX_STRING "([A-Za-z0-9_]+)@[A-Za-z0-9_-]+: ?~ ?\\# (.+\\n)"
#define LAST_STRING "Fin."
#define SIZE_BUFFER 100
#define COUNT_GROUPS 3

void print_group(char *str, regmatch_t *groups, int group_number){
    int position_begin = groups[group_number].rm_so;
    int position_end = groups[group_number].rm_eo;
    for(int i = position_begin; i < position_end; ++i)
        printf("%c", str[i]);
}

void print_name_and_command(regex_t *regex, char *buffer,
regmatch_t *groups){
    while(fgets(buffer, SIZE_BUFFER, stdin) != NULL &&
    strstr(buffer, LAST_STRING) == NULL)
        if(regexec(regex, buffer, COUNT_GROUPS, groups, 0) == 0)
        {
            print_group(buffer, groups, 1);
            printf(" - ");
            print_group(buffer, groups, 2);
        }
}

int main(){
    char buffer[SIZE_BUFFER];
    regmatch_t groups[COUNT_GROUPS];
    regex_t regex;
    regcomp(&regex, REGEX_STRING, REG_EXTENDED);

    print_name_and_command(&regex, buffer, groups);

    regfree(&regex);
    return 0;
}
```