

**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**

Студент гр. 3343

Поддубный В.А.

Преподаватель

Государкин Я. С.

Санкт-Петербург

2024

## **Цель работы**

Изучить и научиться применять функции библиотеки `regex.h` языка Си для поиска совпадений в строках при помощи регулярных выражений. Освоить навыки для написания регулярных выражений на языке Си.

## Задание

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.

### Описание функций

- Функция printGroup отвечает за вывод группы символов из строки в соответствии с заданным диапазоном.

### Основная часть программы

Программа представляет собой консольное приложение на языке программирования, которое осуществляет поиск и вывод определенных групп символов из введенной строки в соответствии с заданным регулярным выражением. В процессе работы программы используются библиотеки для работы с регулярными выражениями, строками и стандартным вводом-выводом.

## **Выполнение работы**

Описание функций:

- `int main()`: главная функция программы, в ней компилируется `regex_t` и проводится сравнение со строкой пришедшей из функции `char *getSentence`, при положительном результате печатает ответ
- `char *getSentence()`: считывает посимвольно текст с консоли.

Разработанный программный код см. в приложении А.

## **Выводы**

В ходе выполнения лабораторной работы были освоены необходимые навыки для использования регулярных выражений на языке Си с помощью библиотеки `regex.h`, а также для составления регулярных выражений согласно требованиям. Были изучены необходимые языковые конструкции и особенности записи регулярных выражений на языке Си.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

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

#define groupSize 3

char *getSentence() {
    char ch;
    int len = 0;
    char *sentence = calloc(len+1, sizeof(char));
    while ((ch = getchar()) != '\n') {
        sentence = realloc(sentence, sizeof(char) * len+2);
        sentence[len++] = ch;
        if(strcmp(sentence, "Fin.") == 0) return sentence;
    }
    return sentence;
}

int main() {
    regex_t regex;
    const char *expression = "([a-zA-Z0-9_]+)@[a-zA-Z0-9_-]+: *~ *#
(.+)";
    regcomp(&regex, expression, REG_EXTENDED);
    char *sentence;
    while (strcmp((sentence = getSentence()), "Fin.") != 0) {
        regmatch_t matches[groupSize];
        if (regexexec(&regex, sentence, groupSize, matches, 0) == 0) {
            printf("%.1s      -      %.1s\n", matches[1].rm_eo -
matches[1].rm_so, &(sentence[matches[1].rm_so]),
                    matches[2].rm_eo      -      matches[2].rm_so,
&(sentence[matches[2].rm_so]));
        }
        free(sentence);
    }
    return 0;
}
```