**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 слеующий вид:

Сначало 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 сторки 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 *regexec*. 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 полученнх 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 |
|  | 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. | root – su box  root - exit |

# 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;

}