**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 |  | Пушко К.Д. |
| Преподаватель |  | Глазунов С.А. |

Санкт-Петербург

2024

## Цель работы

Целью работы является освоение работы с регулярными выражениями и группами, а также использование их в языке программирования 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:

* 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

Сначала создаются регулярное выражение, группы, а также паттерн регулярного выражения. Далее регулярное выражение компилируется. После этого начинается цикл, который длится до тех пор, пока на вход не подадут слово, останавливающее ввод.

В каждой итерации цикла идет считывание строки, которая далее проверяется на соответствие регулярному выражению. Если строка подходит, то посимвольно выводится с помощью групп сначала имя пользователя, совершившего команду, а затем саму команду.

Данна программа выводит имена пользователей и команды, которые они совершили, если они соответствуют регулярным выражениям.

Разработанный программный код см. в приложении А.

## 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: ~ $ ^C  Exit from box: box@5718c87efaa7:  ~ $ exit  exit from container:  root@5718c87efaa7: ~ # exit  kot@kot-ThinkPad:~$ ^C  Fin. | root - su box  root - exit |

## Выводы

Были изучены правила написания регулярных выражений, а также работа с группами. Так же было реализовано применение регулярных выражений и групп в языке программирования Си.

# Приложение А Исходный код программы

Название файла: main.c

#include <stdio.h>

#include <stdlib.h>

#include <regex.h>

#include <string.h>

int main()

{

regex\_t regex;

regmatch\_t groups[4];

char\* pattern = "([A-Za-z0-9\_]+)@([A-Za-z0-9\_-]+):\\s?~\\s?# (.+)";

regcomp(&regex, pattern, REG\_EXTENDED);

char sentence[100];

char breakWord[10] = "Fin.";

while (strcmp(sentence,breakWord))

{

fgets(sentence,100,stdin);

if(regexec(&regex,sentence,4,groups,0)==0)

{

for (int i = groups[1].rm\_so; i < groups[1].rm\_eo; ++i)

{

printf("%c",sentence[i]);

}

printf(" - ");

for (int i = groups[3].rm\_so; i < groups[3].rm\_eo; ++i)

{

printf("%c",sentence[i]);

}

}

}

return 0;

}