**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 |  | Сербиновский Ю.М. |
| Преподаватель |  | Глазунов С.А. |

Санкт-Петербург

2024

## Цель работы

Научиться использовать регулярные выражения в языке программирования C.

## 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:

1. Сначала идет имя пользователя, состоящее из букв, цифр и символа **\_**
2. Символ @
3. Имя компьютера, состоящее из букв, цифр, символов **\_** и **-**
4. Символ **:** и **~**
5. Символ **$**, если команда запущена в оболочке пользователя и **#**, если в оболочке суперпользователя. При этом между двоеточием, тильдой и **$** или **#** могут быть пробелы.
6. Пробел
7. Сама команда и символ переноса строки.

## Выполнение работы

Регулярное выражение записано в reg\_str, данная строка компилириется, и результат записывается в reg\_compiled. Далее в цикле из stdin считывается по одному предложению (каждое предложение — с новой строки). При помощи regexec каждое предложение проверяется на соответствие регулярному выражению. Если проверка пройдена, то при помощи двойного цикла for и массива структур regmatch\_t match\_groups, хранящего информацию о группах регулярного выражения относительно текущего предложения, в stdout выводится комбинация <имя пользователя> - <имя\_команды>. Считывание завершается, когда пользователь вводит «Fin.».

## 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

Был получен опыт работы с регулярными выражениями на языке C, была написана программа, определяющая по строке в консоли Linux, каким пользователем и какая была выполнена команда.

# Приложение А Исходный код программы

Название файла: main.c

#include <stdio.h>

#include <regex.h>

#include <string.h>

int main() {

char\* reg\_str = "([a-zA-Z0-9\_]+)@[a-zA-Z0-9\_-]+: ?~ ?# (.+\n)";

regex\_t reg\_compiled;

regcomp(&reg\_compiled, reg\_str, REG\_EXTENDED);

size\_t max\_groups = 3;

regmatch\_t match\_groups[max\_groups];

char cur\_str[1000];

fgets(cur\_str, 1000, stdin);

while(strstr(cur\_str, "Fin.") == 0) {

if (regexec(&reg\_compiled, cur\_str, max\_groups, match\_groups, 0) == 0) {

for (int i = 1; i < max\_groups; i++) {

for (int j = match\_groups[i].rm\_so; j < match\_groups[i].rm\_eo; j++) {

printf("%c", cur\_str[j]);

}

if (i == 1)

printf(" - ");

}

}

fgets(cur\_str, 1000, stdin);

}

return 0;

}