САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ОТЧЕТ

по лабораторной работе №1

по дисциплине «Программирование»

Тема: Лабораторная работа № 1. Регулярные выражения

Студент гр. 3343 _____ Кербель Д. А.

Преподаватель _____ Государкин Я. С.

Санкт-Петербург

2024

Цель работы

Обучиться использованию регулярных выражений, заложенных в библиотеке regex.h, написав программу на языке программирования 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.

Выполнение работы

Программа получает на вход текст, заканчивающийся строчкой ‘Fin.’. Из текста она выводит только команды в оболочке суперпользователя.

Функция GroupPrint принимает строку string и структуру regmatch_t group, которая содержит информацию о группе найденной в регулярном выражении. Функция выводит символы из строки string в диапазоне от group.rm_so до group.rm_eo.

В функции main программа компилирует регулярное выражение, считывает строки из стандартного ввода, и если строка соответствует регулярному выражению, то вызывает GroupPrint для вывода найденных групп. Поиск прекращается при обнаружении строки "Fin.". После завершения работы программа освобождает ресурсы, выделенные для компиляции регулярного выражения.

Разработанный программный код см. в приложении А.

Тестирование

Таблица 1.

№	Входные данные	Выходные данные	Комментарии
1.	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	Ожидаемый вывод.

Выводы

В ходе выполнения лабораторной работы, мною был освоен синтаксис, необходимый для написания регулярных выражений на языке Си, а также написана программа с их использованием.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

```
#include <stdio.h>

#include <string.h>

#include <regex.h>

void GroupPrint(char* string, regmatch_t group) {

    for (int i=group.rm_so; i<group.rm_eo; i++) {

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

    }

}


int main()

{

    char* rStr = "([a-zA-Z0-9_]+)@[a-zA-Z0-9_-]+: *~ *# (.*)";

    size_t maxGroups = 3; regmatch_t groupArray[maxGroups]; regex_t
regexCompiled;

    if (regcomp(&regexCompiled,rStr,REG_EXTENDED))

    {

        printf("Cant Compile\n");

        return 0;

    }
```

```
    char s[101];
    while(1)
    {
        fgets(s,100,stdin);
        if (strstr(s,"Fin.") != NULL){
            break;
        }
        if(regexec(&regexCompiled, s,maxGroups,groupArray,0) == 0){
            GroupPrint(s,groupArray[1]);
            printf(" - ");
            GroupPrint(s,groupArray[2]);
        }
    }
    regfree(&regexCompiled);
    return 0;
}
```