**Министерство образования и науки Российской Федерации**

**Федеральное государственное автономное образовательное учреждение**

**высшего образования**

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ**

**ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

****

Факультет безопасности информационных технологий

Дисциплина: «Системное программирование»

**ОТЧЕТ**

**Лабораторная работа № 1**

**«Файлы и потоки ввода-вывода»**

**Вариант 21**

|  |  |  |  |
| --- | --- | --- | --- |
| Выполнил студент группы N3351: | | Яресько С.А. | \_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Проверил: | Гирик А.В. | \_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Дата: | 25.10.19 |  |

Санкт-Петербург, 2019

Цель работы:

Написать программу для POSIX-совместимой ОС на языке C, выполняющую ввод текстовой информации из файла или стандартного потока ввода, затем осуществляющую

преобразование информации в соответствии с вариантом задания и выводящую

преобразованную информацию в файл или в стандартный поток вывода.

Вариант 21.

Задание:

Найти в исходном тексте все слова, соответствующие заданному шаблону. В

шаблоне могут присутствовать символы “.” (точка), означающий любую букву,

“@” (коммерческое эт), означающий гласную, “$” (знак доллара), означающий

согласную, и “?” (знак вопроса), означающий любую букву или её отсутствие.

Пример: шаблону «.@$@??» соответствуют слова «mama», «pony», «topic», «aero»

и не соответствуют «dad», «port», «aeroplan», «beam».

Шаблон передается программе в командной строке с помощью обязательной

опции -совместимой ОС на языке C, выполняющую вводt шаблон. Найденные слова выводятся по одному на строку с указанием

номера строки исходного файла, в которой было обнаружено совпадение.

#include <stdio.h>

#include <string.h>

#include <unistd.h>

#include <stdlib.h>

int main (int argc, char \*argv[])

{

*// Для getopt()*

int option=0; *// для получения ключа*

int key\_t=0; *// для проверки, был ли указан ключ -t*

char \*\*input\_file = NULL; *// хранит адрес исходного файла, если указан*

char \*\*output\_file = NULL; *// хранит адрес конечного файла, если указан*

*// Для getline()*

FILE \* inFile = NULL; *// ссылка на исходных файл*

size\_t len = 0; *// для работы getline*

*// Для main()*

int c = 0; *// для ввода строки*

FILE \* outFile = NULL; *// ссылка на конечный файл*

char \* mask = NULL; *// шаблон для поиска слова*

char \* str = NULL; *// отдельные строки из файла*

char \* word = NULL; *// отдельное слово из строки*

int condition = 0; *// для переменного условия главного цикла while*

int yes = 0; *// кол-во успешно найденных символов при . @ $*

int del = 0; *// кол-во успешно найденных символов при ?*

int m = 0; *// счетчик маски*

int mask\_count = 0; *// кол-во элементов в маске без ?*

int str\_count = 0; *// подсчет номера строки*

char any [] = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"; *// любая буква*

char vowel [] = "eaiouyEAIOUY"; *// гласные*

char consonant [] = "bcdfghjklmnpqrstvwxzBCDFGHJKLMNPQRSTVWXZ"; *// согласные*

*// Принятие ключей через getopt()*

**while** ( (option = getopt(argc,argv,"t:h")) != -1)

{

**switch** (option)

{

**case** 't': **for** (int i=0; i<strlen(optarg); i++)

**if** (strchr(".@$?",optarg[i]) == NULL) { *// поиск посторонних символов в введенном шаблоне*

printf("**\n**Not allowed symbol(s) in the template**\n**"

"**\n**For more information use option: -h**\n\n**");

**return** 0; }

key\_t=1;

mask = optarg;

**break**;

**case** 'h': printf("h INSERTED**\n**");

**break**;

**case** '?': printf("**\n**Invalid option: -%c**\n**Use option: -t **\'**<template>**\'** [source\_file]**\n**For example: -t

**\'**.@$@?**\'\n**"

"**\n**For more information use option: -h**\n\n**",optopt);

**return** 0;

}

}

*// Если нет ключа '-t' для задания шаблона, то выдается ошибка*

**if** (key\_t==0) {

printf("**\n**Option is required**\n**Use option: -t **\'**<template>**\'\n**For example: -t **\'**.@$@?**\'\n**"

"**\n**For more information use option: -h**\n\n**");

**return** 0; }

*// Считывание кол-ва символов в маске, которые не являются '?'*

**for** (int a=0; a < strlen(mask); a++)

**if** ( (mask[a]=='.') || (mask[a]=='@') || (mask[a]=='$') )

mask\_count++;

*// Считывание пути исходного файла*

input\_file = argv + optind;

*// Открытие указанного исходного файла для чтения*

**if** (\*input\_file!=NULL) {

inFile = fopen(\*input\_file,"r");

**if** (inFile == NULL) {

printf("Error accessing the IN specified file.**\n\n**"); *// ошибка доступа к файлу*

**return** 0; } }

*// Если исходный файл не был указан, создается и заполняется временный файл*

**else** {

inFile = tmpfile(); *// создание временного файла и получение указателя на файл*

**if** (inFile==NULL) {

printf("Unable to create temprorary file.");

**return** 0; }

printf("**\n**Insert your text for scanning:**\n**");

c = getchar();

**while** (c != EOF) {

fputc(c, inFile);

c = getchar(); }

fclose(inFile);

inFile = fopen(inFile, "w+r"); *// открытие временного файла*

printf("**\n\n**"); }

*// задание условия "проверка на конец файла" для исходного файла*

**if** (inFile!=NULL)

condition = !feof(inFile);

*// Считывание пути конечного файла*

output\_file = argv + optind + 1;

*// Открытие конечного файла для записи и перенаправление в него потока вывода*

**if** ( \*output\_file!=NULL && strcmp(\*output\_file,"SHELL=/bin/bash") )

outFile = freopen(\*output\_file, "w+", stdout);

**while** ( condition )

{

**if** (inFile!=NULL) {

getline(&str, &len, inFile); } *// Выделяется строка из исходного файла и записывается в str*

str\_count++; *// Подсчет строк*

word = strtok(str, " **\n\r\t**"); *// Отделяется первое слово в строке str*

**while** (word) *// Пока не конец строки, читаем слово*

{

**if** ( (strlen(word) <= strlen(mask)) && (strlen(word) >= mask\_count) ) *// Отсекаются заведомо ложные*

{

yes = del = m = 0;

*// Перебор отделённого слова word[t]*

**for** (int t=0; t < strlen(mask); t++)

{

**switch** (mask[m])

{

**case** '.':

**if** (strchr(any,word[t]) && word[t]!='\0') {

yes++; m++; }

**else** **if** (mask[m-1]=='?') {

t=t-2; del--; }

**else**

yes = del = m = 0;

**break**;

**case** '@':

**if** (strchr(vowel,word[t]) && word[t]!='\0') {

yes++; m++; }

**else** **if** (mask[m-1]=='?') {

t=t-2; del--; }

**else**

yes = del = m = 0;

**break**;

**case** '$':

**if** (strchr(consonant,word[t]) && word[t]!='\0') {

yes++; m++; }

**else** **if** (mask[m-1]=='?') {

t=t-2; del--; }

**else**

yes = del = m = 0;

**break**;

**case** '?':

**if** (strchr(any,word[t]) && word[t]!='\0') {

del++; m++; }

**else** **if** (mask[m-1]=='?') {

t=t-2; del--; }

**else** **if** ((word[t]=='\0') && ((yes+del)==strlen(word)) && (yes==mask\_count)) {

printf("%d**\t**%s**\n**", str\_count, word);

**break**; }

**break**;

}

*// Проверка на найденное слово, когда закончилась маска (шаблон)*

**if** ( ((yes+del)==strlen(word)) && (mask[m]!='.') && (mask[m]!='@') && (mask[m]!='$') &&

(mask[m]!='?') ) {

printf("%d**\t**%s**\n**", str\_count, word);

**break**; }

}

}

*// Отделяем следующее слово в строке*

word = strtok(NULL," **\n\r\t**");

}

**if** (inFile!=NULL)

condition = !feof(inFile); *// обновление условия "проверка на конец файла" для исходного файла*

}

**if** (inFile!=NULL)

fclose(inFile); *// закрытие исходного файла*

**return** 0;

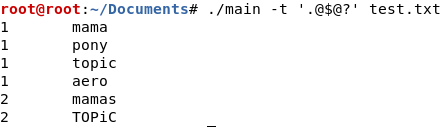
}

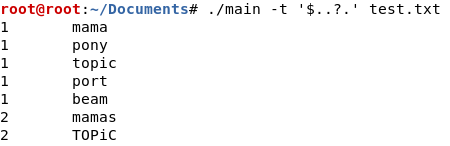
Примеры работы программы:

1. Случай, если задан исходный файл с текстом.
   1. Содержимое исходного файла.



* 1. Запуск. Через ключ ‘-t’ указывается шаблон для поиска слова. Затем путь к файлу.





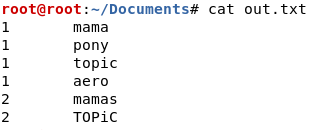
1. Случай, если задан конечный файл.
   1. Содержимое исходного файла.



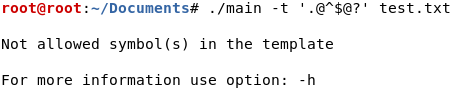
* 1. Запуск. Через ключ ‘-t’ указывается шаблон для поиска слова. Затем путь к файлу. Затем путь к конечному файлу.



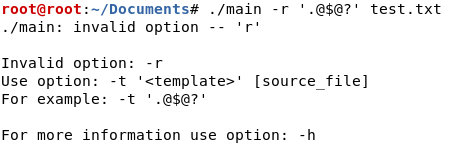
* 1. Содержимое конечного файла.



1. Пример ошибки. Некорректный символ в шаблоне.



1. Пример ошибки. Некорректный ключ.



1. Пример ошибки. Отсутствие необходимого ключа ‘-t’.

