**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: Обзор стандартной библиотеки

|  |  |  |
| --- | --- | --- |
| Студент гр. 6304 |  | Запевалов А.И. |
| Преподаватель |  | Берленко Т. А. |

Санкт-Петербург

2017

**Цель работы**

Научиться использовать функции стандартной библиотеки при работе со строками.

**Задача.**

Напишите программу, на вход которой подается текст на английском языке (длина текста не превышает 1000 символов) и слово str (длина слова не превышает 30 знаков). Слова в тексте разделены пробелами или точкой. Программа должна вывести строку "exists", если str в тексте есть и "doesn't exist" в противном случае.

Программа должна реализовать следующий алгоритм:

разбить текст на слова, используя функции стандартной библиотеки

отсортировать слова, используя алгоритм быстрой сортировки (см. функции стандартной библиотеки)

определить, присутствует ли в тексте str, используя алгоритм двоичного поиска (для реализации алгоритма двоичного поиска используйте функцию стандартной библиотеки)

вывести строку "exists", если str в тексте есть и "doesn't exist" в противном случае.

Исходный код программы

1. Код скопирован в среду разработки и скомпилирован

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int compare(const void\* a, const void\* b) //Compares two strings using strcmp function

{

return strcmp((char\*) a, (char\*) b);

}

int main()

{

char del[]=". "; //Division tags

char strn[1000];

fgets(strn, 1000, stdin);

char str[30];

fgets(str, 30, stdin);// Reads text and key word

int i=0;

char mas[100][1000];

char \*p=strtok(strn, del); // Divides words from text into strings

while (p != NULL)

{

strcpy(mas[i], p);

p = strtok (NULL, del);

i++;

}

qsort(mas, i, 1000, compare); // Sorts divided words by ascending

char\* point=(char\*)bsearch(str, mas, i, 1000, compare); // Seeks for key word

if (point==NULL) // Prints result

{

printf("doesn't exist");

}

else

{

printf("exists");

}

return 0;

}

# **Ход работы**

## **Функция быстрой сортировки**

Функция **qsort** предназначена упорядочивать множества блоков байтов равной длины. Второй аргумент функции — это число таких блоков, третий аргумент — длина каждого блока. Первый аргумент — это адрес, где находится начало первого блока (предполагается, что блоки в памяти расположены друг за другом подряд).

Четвёртый аргумент функции **qsort** — это имя функции, которая умеет сравнивать два элемента массива. В нашем случае это:

int compare(const void\* a, const void\* b) //Compares two strings using strcmp function

{

return strcmp((char\*) a, (char\*) b);

}

В силу указанной универсальности функции сортировки, функция сравнения получает в качества аргумента адреса двух блоков, которые нужно сравнить и возвращает 1, 0 или -1:

* положительное значение, если x1 > x2
* 0, если x1 == x2
* отрицательное значение, если x1 < x2

Функция сортировки выглядит так:

qsort(mas, i, 1000, compare); // Sorts divided words by ascending

## **Функция двоичного поиска и вывод результата**

char\* point=(char\*)bsearch(str, mas, i, 1000, compare); // Seeks for key word

if (point==NULL) // Prints result

{

printf("doesn't exist");

}

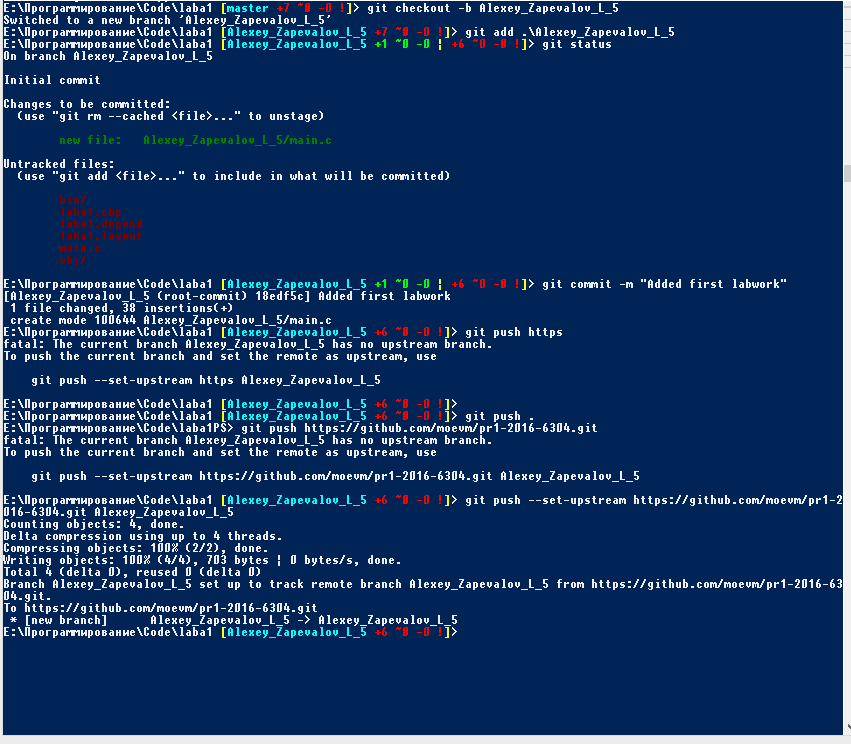
else

{

printf("exists");

}

Файлы запушены на git через командную строку



**Вывод.**

Благодаря функциям стандартной библиотеки языка С мы смогли работать с массивом строк, заметно сократив время и количество строк кода. Используя одну функцию сравнения мы смогли выполнить несколько задач.