**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. 6303 |  | Эвергрин П.С. |
| Преподаватель |  | Чайка К.В. |

Санкт-Петербург

2017

**Цель работы.**

Написать 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.

**Ход работы.**

1. В локальном репозитории git была создана новая ветка *evergreen\_\_sem2\_lab1* командой

git checkout -b *evergreen\_\_sem2\_lab1*

1. Создана папка *evergreen\_\_sem2\_lab1*

mkdir *evergreen\_\_sem2\_lab1*

и файл *main.c*

touch main.c

1. Далее приведено краткое описание работы программы.
2. Из входного потока считывается текст(input) и слово *str(key)*

char input[1001];

fgets(input, 1001, stdin);

char\* key = (char\*) malloc(31 \* sizeof(char));

scanf("%s", key);

1. С помощью функции стандартной библиотеки *strtok* выполняется разбиение текста на отдельные слова и запись их в массив

|  |
| --- |
| for(c=0;input[c];c++) |
|  | if((input[c]==' ') && (input[c+1] != ' ')) |
|  | i++; |
|  | char\* word, \*\* wordList; |
|  | wordList = (char\*\*) malloc(sizeof(char\*) \* i); |
|  | //starts assigning each array to each word |
|  | word = strtok(input, " ."); |
|  | c = 0; |
|  | while(word != NULL) { |
|  | wordList[c] = word; |
|  | word = strtok(NULL, " ."); |
|  | c++; |
|  | } |

1. С помощью функции стандартной библиотеки *qsort* выполняется сортировка слов по возрастанию

qsort(wordList, i, sizeof(char\*), cmpstr);

1. С помощью функции стандартной библиотеки *bsearch* выполняется бинарный поиск нужного слова в массиве

exist = (char\*) bsearch(&key, wordList, i, sizeof(char\*), cmpstr);

1. В качестве вспомогательной функции, передающейся в качестве аргумента в *qsort* иbsearch реализована функция-компаратор, срывающая две строки

int cmpstr(const void\* a, const void\* b)

**{**

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

**}**

1. После поиска слова в массиве программа возвращает результат поиска

if(exist != NULL)

printf("exists");

else

printf("doesn't exist");

1. Полный код программы
2. В локальный репозиторий git была добавлена папка с исходным кодом программы

git add *evergreen\_\_sem2\_lab1*

1. Сделан коммит в локальный репозитрий

git commit

1. Изменения отправлены в удаленный репозиторий

git push origin *evergreen\_\_sem2\_lab1*

1. Сделан *pull request* в ветку *master*

**Выводы.**

В ходе лабораторной работы изучены возможности стандартной библиотеки языка Си. Полученные знания закреплены в ходе разработки программы для анализа массива целых чисел на предмет наличия в нем указанного числа с использованием функций стандартной библиотеки.