

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 языка Си

Студент гр. 0382

Гудов Н.Р.

Преподаватель

Берленко Т.А.

Санкт-Петербург

2021

Цель работы.

Знакомство со стандартной библиотекой языка Си. Приобретение навыков применения функций, и прочего содержимого стандартной библиотеки языка.

Задание.

Вариант1

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.

Основные теоретические положения.

Стандартной библиотекой языка Си (также известная как `libc`, `crt`) называется часть стандарта ANSI C, посвященная заголовочным файлам и библиотечным подпрограммам. Является описанием реализации общих операций, таких как обработка ввода-вывода и строк, в языке программирования Си. Стандартная библиотека языка Си — это описание программного интерфейса, а не настоящая библиотека, пригодная для использования в процессе компиляции.

Так в работе были использованы следующие заголовочные файлы:

string.h — заголовочный файл стандартной библиотеки языка Си, содержащий функции для работы с нуль-терминированными строками и различными функциями работы с памятью. Кроме этого, строковые функции работают только с набором символов ASCII или его совместимыми расширениями.

stdlib.h — заголовочный файл стандартной библиотеки языка Си, который содержит в себе функции, занимающиеся выделением памяти, контролем процесса выполнения программы, преобразованием типов и другие.

stdio.h — заголовочный файл стандартной библиотеки языка Си, содержащий определения макросов, константы и объявления функций и типов, используемых для различных операций стандартного ввода и вывода.

Выполнение работы.

В начале работы выделяется память *text*, для хранения введенного пользователем текста и буфер *word* для слова. Обе символьной строки считываются функцией стандартной библиотеки *fgets()* из стандартного потока ввода. После чего из строк удаляются полученные при вводе символы новой строки.

Далее выделяется память под отдельное хранение каждого из слов, содержащегося в полученном предложении. Слова выявляются при помощи функции *strtok()*, после чего последовательно направляются в выделенную память, где подвергаются сортировке с использованием *qsort()*. Для сравнения используется функция стандартной библиотеки языка *strcmp()*.

На завершающем этапе, перед получением ответа о наличии слова в строке, отсортированные слова подвергаются проверки при помощи алгоритма двоичного поиска, реализованного в функции *bsearch()*. Очищается используемая память.

Тестирование

Результаты тестирования представлены в таблице 1.

Таблица 1-Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	Abcd qwerty Linux. Zaq ghj. Linus	doesn't exist	Работает верно
2.	Abcd qwerty Linux. Zaq ghj. Linux	exist	Работает верно

Вывод

Были изучены основные положения стандартной библиотеки языка Си.

Успешно разработана программа, использующая функции стандартной библиотеки языка.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.c

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

int qsf(const void *a, const void *b)
{
    char** x=(char**)a;
    char** y=(char**)b;
    return strcmp(*x, *y);
}

int main(){
    char *text=(char*)malloc(1001);
    char *word=(char*)malloc(31);

    fgets( text,1001,stdin);
    text[strcspn(text, "\n")] = 0;
    /**
    fgets(word,31,stdin);
    word[strcspn(word, "\n")] = 0;

    char **words=(char**)malloc(300*sizeof(char*));
    int i=0;
    char* cutword=strtok(text, ". ");

    while (cutword != NULL) {
        words[i++] = cutword;
        cutword = strtok(NULL, ". ");
    }

    qsort(words, i, sizeof(char**), qsf);

    char ** t = bsearch(&word, words ,i, sizeof(char**), qsf);

    free(text);
    free(word);
    free(words);

    if(t)
    {printf("exists");}
    else
    {printf("doesn't exist");}
    return 0;
}
```