Федеральное агентство связи

ФГОБУ ВО «СибГУТИ»

Кафедра вычислительных систем

Технология разработки программного обеспечения

Курсовой проект

Алфавит

Выполнили:

Студенты 1-го курса

Группы РИ-57

Кудратов Ф.Ш.

Цебров А.В.

Сат Т.А.

Овчинникова С.Д.

Проверил

Ассистент кафедры ВС

Пименов Е.С.

Новосибирск

2016

Содержание:

1.Задание……………………………………………………..

2.Техническое задание…………………………………………………………………

3.Описание алгоритма ……………..

4.Код………………………………………………………….

5.Описание теста……………..

6.Тестовые данные …………………..

**Задание.**

Программа “Алфавит” ­ выводит слова из входного текста по одному в алфавитном порядке.

Распределение ролей:

Кудратов – тимлид, организация.

Цебров – код программы, тесты.

Сат – интеграция и автоматизация.

Овчинникова –тесты.

**Техническое задание.**

"Алфавит" - программа, рассчитанная для сортировки входного текста в алфавитном порядке

**Интерфейс пользователя.**

Приложение работает в интерактивном консольном режиме: будет спрошен язык, на котором будет отображаться интерфейс. Далее требуется набор названия файла, ввод которого можно отследить на экране, из соответственного файла будет совершен ввод текста в программу и его дальнейшая обработка.

## Формат входных данных.

* Программа поддерживает ввод английских названий текстовых документов, которые должны принадлежать к типу ".txt".
* Внутри самого файла должен содержаться текст на русском языке или английском, который может иметь в себе знаки препинания и символы. Разрешено одновременное использование и русских, и английских слов в тексте, поскольку предполагается, что пользователю может понадобиться такая функция.
* Длина строки, как и название файла, не должны превышать 260 символов. Текст в вводном файле также должен заканчиваться символом "\n" - переводом на другую строку(Enter<-).

## Результат работы программы.

Исходом работы программы будет являться построковый вывод слов, взятых из файла, в алфавитном порядке в рабочую консоль.

Замечания:

* Если в исходном тексте стояло два слова, начинающихся на одну букву, то они остаются на своих местах.
* Приоритета заглавных букв над строчными в расстановке не существует.

**Краткое описание функций и алгоритма.**

Из файла, название которого вводится пользователем, производится познаковый ввод символов, после чего происходит обработка текста: удаляются все символы, не являющиеся буквами. Потом происходит разделение текста на отдельные слова, у которых проверяется первая буква, которая и проверяется функцией и по которой идет расстановка. Далее происходит вывод по одному слов.

**Exchange**

Меняет слова-указатели местами, если они не в алфавитном порядке, информацию о котором получает из функции sort.

**Stok**

Делит строку на слова для удобной работы с ними.

**Schr**

По введенному массиву происходит поиск символа, который также передается в функцию, и возвращает значение индекса первой встречи символа в случае успеха.

**Crop**

Удаляет все символы, не являющиеся буквами поддерживаемых алфавитов из входного массива символов.

**Sort**

Сортирует по алфавитному порядку, используя функции, приведенные выше, слова из массива.

**Interact**

Осуществляет пользовательский ввод названия файла, а также ввод слов, которые требуется расставить по алфавиту.

**Out**

Функция, выводящая слова по одному в алфавитном порядке.

**Mes**

Функция, выводящая сообщения на выбранном языке.

**In**

Спрашивает пользователя о том, какой язык интерфейса использовать.

**Код программы.**

**Файл Main.c**

#include <stdio.h>

#include <stdlib.h>

#include "functions.h"

int main() {

int b = 0, a = 0;

char string[260];

char fstring[260];

a = in();

if (a == -1)

return -1;

b = interact(string, a) + 1;

if (b == 0)

return 0;

char \*ptr[b];

crop(string, fstring);

stok(fstring, ptr);

sort(ptr, b);

out(ptr, b, a);

getchar();

getchar();

return 0;

}

**Файл Interface.c**

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

void mes(int a, int b) {

setlocale(LC\_ALL, "Rus");

if (a == 1){

if (b == 0)

printf("\t\t---Welcome to program Alphabet--\n");

if (b == 1)

printf("Input name of file to take string from:");

if (b == 2)

printf("Error: file is incorrect.\n");

if (b == 3)

printf("Error: string is incorrect.\n");

if (b == 4)

printf("The original string is:");

if (b == 5)

printf("\nNumber of words is:");

if (b == 6)

printf("Output of words one by one:");

}

if (a == 2){

if (b == 0)

printf("\t\t---Добро пожаловать в программу Алфавит---\n");

if (b == 1)

printf("Введите название файла для взятия строки:");

if (b == 2)

printf("Ошибка: некорректный файл.\n");

if (b == 3)

printf("Ошибка: некорректная строка.\n");

if (b == 4)

printf("Исходная строка:");

if (b == 5)

printf("\nКоличество слов:");

if (b == 6)

printf("Вывод слов по одному:");

}

}

int interact (char \*string, int a) {

int j = 0, size = 0;

char namein[260];

mes(a, 0);

mes(a, 1);

getchar();

do{

if (j > 260)

return j;

scanf("%c", &namein[j]);

j++;

} while (namein[j - 1]!='\n');

namein[j-1] = '\0';

FILE \*fp = fopen (namein, "r");

if(fp == NULL){

mes(a, 2);

return -1;

}

fscanf (fp, "%c", &string[0]);

if ((string[0] == '\n') || (string[0] == ' ')){

mes(a, 3);

return -1;

}

j = 1;

do{

if (j >= 260){

mes(a, 3);

return -1;

}

fscanf (fp, "%c", &string[j]);

if (string[j] == ' ')

size++;

j++;

} while ((string[j - 1] != '\n') && (j < 260));

string[j - 1] = '\0';

mes(a, 4);

printf("%s", string);

fclose(fp);

return size;

}

void out (char \*ptr[], int size, int a) {

setlocale(LC\_ALL, "Rus");

int j = 0;

mes(a, 5);

printf("%d\n", size);

mes(a, 6);

for (j = 0; j < size; j++)

printf("\n%s", ptr[j]);

}

int in() {

int a;

printf("Choose your interface language:\nType 1 for English.\nType 2 for Russian.\n");

scanf("%d", &a);

if ((a != 1) && (a != 2)){

printf("Wrong language input.");

return -1;

}

return a;

}

**Файл Functions.h**

#ifndef funcs

#define funcs

void exchange (char \*ptr[], int j);

int schr(char str[],char ch);

int stok(char str[], char \*ptr[]);

void crop(char string[], char \*fstring);

void sort(char \*ptr[], int size);

int interact (char \*string, int a);

void out (char \*ptr[], int size, int a);

void mes(int a, int b);

int in();

#endif

**Файл Functions.c**

#include <stdio.h>

#include <stdlib.h>

void exchange (char \*ptr[], int j) {

char \*chtmp;

char \*chtmp2;

chtmp2 = ptr[j + 1];

chtmp = ptr[j];

ptr[j + 1] = chtmp;

ptr[j] = chtmp2;

}

int schr(char str[],char ch) {

int i;

int idx = -1;

for(i = 0;(str[i] != '\0') && (str[i] != ch);i++);

if(str[i] == ch)

idx = i;

return idx;

}

int stok(char str[], char \*ptr[]) {

char \*suf = str;

ptr[0] = str;

int i, j = 1;

char delim = ' ';

while((i = schr(suf, delim)) >= 0){

suf[i] = '\0';

suf = suf + i + 1;

ptr[j] = suf;

j++;

}

return j;

}

void crop(char string[], char \*fstring) {

int j = 0, i = 0;

while (string[j] != '\0'){

if ((string[j] >= 'А' && string[j] <= 'Б') || (string[j] >= 'а' && string[j] <= 'п') || (string[j] >= 'р' && string[j] <= 'я') || string[j]==' '){

fstring[i] = string[j];

i++;

}

if ((string[j] >= 'A' && string[j] <= 'Z') || (string[j] >= 'a' && string[j] <= 'z')){

fstring[i] = string[j];

i++;

}

j++;

}

fstring[i] = '\0';

}

void sort(char \*ptr[], int size) {

char \*mstmp;

char \*mstmp2;

char A[] = "АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ";

char a[] = "абвгдеёжзийклмнопрстуфхцчшщъыьэюя";

char B[] = "ABCDEFGHIJKLMNOPQRSTUVWXYZ ";

char b[] = "abcdefghijklmnopqrstuvwxyz ";

int i = 0, j = 0, k = 0;

int x = -1, y = -1;

for (i = 0; i < (size - 1); ++i){

for(j = 0; j < (size - 1); ++j){

mstmp = ptr[j];

mstmp2 = ptr[j + 1];

for(k = 0; k < 33; k++){

if ((mstmp[0] == A[k]) || (mstmp[0] == a[k]))

x = k;

if ((mstmp2[0] == A[k]) || (mstmp2[0] == a[k]))

y = k;

if ((mstmp[0] == B[k]) || (mstmp[0] == b[k]))

x = k;

if ((mstmp2[0] == B[k]) || (mstmp2[0] == b[k]))

y = k;

}

if (y < x)

exchange(ptr, j);

}

}

}

**Описание теста**

Главной задачей теста было нахождение результатов, отличных от ожидаемых, не предусмотренных кодом программы.

**Код теста**

**Файл Main.с**

#define CTEST\_MAIN

#include <ctest.h>

int main(int argc, const char\*\* argv) {

return ctest\_main(argc, argv);

}

**Файл functest.c**

#include <ctest.h>

#include <functions.h>

#include <locale.h>

CTEST(equation\_suite, crop\_of\_symbols)

{

//Given

char string[] = "...sad";

//When

char fstring[260];

crop(string, fstring);

//Then

const int exp = fstring[0];

const int real = 's';

ASSERT\_EQUAL(exp, real);

}

CTEST(equation\_suite, crop\_of\_symbols\_rus)

{

setlocale(LC\_ALL, "Rus");

//Given

char string[] = "дид";

//When

char fstring[260];

crop(string, fstring);

const int exp = fstring[2];

//Then

const int real = 'д';

ASSERT\_EQUAL(exp, real);

}

CTEST(strings\_process, search\_of\_rus\_word)

{

setlocale(LC\_ALL, "Rus");

//Given

char string[] = "...ячсмир";

char ch = 'я';

//When

const int real = schr(string, ch);

//Then

const int exp = 3;

ASSERT\_EQUAL(real, exp);

}

CTEST(strings\_process, search\_of\_rus\_cap\_word)

{

setlocale(LC\_ALL, "Rus");

//Given

char string[8] = "АаааААа";

char ch = 'А';

//When

int real = schr(string, ch);

//Then

const int exp = 0;

ASSERT\_EQUAL(real, exp);

}

CTEST(string\_process, search\_of\_eng\_word)

{

//Given

char string[] = "twitter";

char ch = 'w';

//When

int real = schr(string, ch);

//Then

const int exp = 1;

ASSERT\_EQUAL(real, exp);

}

CTEST(equation\_suite, search\_of\_empty)

{

//Given

char string[] = "123456789";

char ch = 'Z';

//When

int real = schr(string, ch);

//Then

const int exp = -1;

ASSERT\_EQUAL(real, exp);

}

CTEST(equation\_suite, search\_of\_eng\_word2)

{

//Given

char string[] = "TTT";

char ch = 'T';

//When

int real = schr(string, ch);

const int exp = 0;

//Then

ASSERT\_EQUAL(real, exp);

}

CTEST(equation\_suite, stok\_multiple)

{

//Given

char string[]= "aaaay lmaaao";

char \*ptr[2];

int size = stok(string, ptr);

char \*tmp1 = ptr[0];

char \*tmp2 = ptr[1];

//When

const int real\_1 = tmp2[0];

const int exp\_1 = 'l';

const int exp\_2 = 'a';

const int real\_2 = tmp1[0];

const int exp\_size = 2;

//Then

ASSERT\_EQUAL(exp\_1, real\_1);

ASSERT\_EQUAL(real\_2, exp\_2);

ASSERT\_EQUAL(exp\_size, size);

}

CTEST(equation\_suite, stok\_single)

{

//Given

char string[]= "haallo";

char \*ptr[1];

stok(string, ptr);

char \*tmp1 = ptr[0];

//When

const int real = tmp1[0];

//Then

const int expct = 'h';

ASSERT\_EQUAL(real, expct);

}

CTEST(equation\_suite, sort\_eng)

{

//Given

char \*ptr[3];

ptr[0]="b";

ptr[1]="a";

ptr[2]="c";

sort(ptr, 3);

char \*tmp\_a = ptr[0];

char \*tmp\_b = ptr[1];

char \*tmp\_c = ptr[2];

//When

const int real\_a = tmp\_a[0];

const int real\_b = tmp\_b[0];

const int real\_c = tmp\_c[0];

//Then

const int expct\_a = 'a';

const int expct\_b = 'b';

const int expct\_c = 'c';

ASSERT\_EQUAL(expct\_a, real\_a);

ASSERT\_EQUAL(expct\_b, real\_b);

ASSERT\_EQUAL(expct\_c, real\_c);

}

CTEST(equation\_suite, sort\_rus)

{

setlocale(LC\_ALL, "Rus");

//Given

char \*ptr[4];

ptr[0]="Б";

ptr[1]="а";

ptr[2]="Д";

ptr[3]="Г";

sort(ptr, 4);

char \*tmp1 = ptr[0];

char \*tmp2 = ptr[1];

char \*tmp3 = ptr[2];

char \*tmp4 = ptr[3];

//When

const int real\_1 = tmp1[0];

const int real\_2 = tmp2[0];

const int real\_3 = tmp3[0];

const int real\_4 = tmp4[0];

//Then

const int expct\_1 = 'а';

const int expct\_2 = 'Б';

const int expct\_3 = 'Г';

const int expct\_4 = 'Д';

ASSERT\_EQUAL(expct\_1, real\_1);

ASSERT\_EQUAL(expct\_2, real\_2);

ASSERT\_EQUAL(expct\_3, real\_3);

ASSERT\_EQUAL(expct\_4, real\_4);

}

CTEST(equation\_suite, exchange\_of\_eng\_words)

{

//Given

char \*ptr[1];

ptr[0]="lmao";

ptr[1]="ay";

exchange(ptr, 0);

char \*tmp1 = ptr[0];

char \*tmp2 = ptr[1];

//When

const int real\_1 = tmp1[0];

const int real\_2 = tmp2[0];

//Then

const int expct\_1 = 'a';

const int expct\_2 = 'l';

ASSERT\_EQUAL(expct\_1, real\_1);

ASSERT\_EQUAL(expct\_2, real\_2);

}

CTEST(equation\_suite, exchange\_of\_rus\_words)

{

setlocale(LC\_ALL, "Rus");

//Given

char \*ptr[2];

ptr[0]="бо";

ptr[1]="язь";

ptr[2]="ш";

//When

exchange(ptr, 0);

char \*tmp1 = ptr[1];

char \*tmp = ptr[0];

const int real = tmp[0];

const int real\_1 = tmp1[0]

//Then

const int expct\_1 = 'б';

const int expct = 'я';

ASSERT\_EQUAL(real\_1, expct\_1);

ASSERT\_EQUAL(real, expct);

}

Тестовые данные