Национальный исследовательский ядерный университет «МИФИ»

ОТЧЁТ

«Создание проекта и библиотеки пользовательских программ

обработки текстовых файлов»

Выполнил: студент 1 курса группы С19-103 Оленников В.Д.

Преподаватель: Пчелинцев А.В.

Москва 2020

**Текст** **задачи**

**Программа должна:**

- обрабатывать два входных текстовых файла;

- не изменять данные в файлах (для существующих файлов);

- давать возможность пользователю вводить исходные данные в файлы (для несуществующих файлов);

- распознавать аргументы командной строки и обрабатывать данные, в соответствии с ними; - контролироваться пользователем через меню, если отсутствует достаточный набор аргументов командной строки;

- предоставлять пользователю возможность выбора четырёх («смешивание», «выравнивание», «поиск», «статистика») вариантов обработки данных (по вариантам); - для пунктов «выравнивание», «поиск», «статистика» запрашивать номер обрабатываемого файла;

- для пункта «статистика» строить гистограмму распределения;

- обеспечивать возможность задать имена файлов;

- выводить результат на экран.

**Вариант 83**

**Смешивание (А)** - все чётные слова первого файла заменить на нечётные слова из второго файла.

**Выравнивание (Б)** - установить красную строку размером N пробелов, если строка начинается с пробела

**Поиск (В) -** найти в файле слово заданной длины, вывести слово и его позицию

**Статистика (Г) -** подсчитать в файле общее количество и число разных слов

**Алгоритм решения**

1. Разработать четыре алгоритма обработки данных (А, Б, В, Г). Каждый алгоритм реализовать в отдельной функции.

2. Подготовить алгоритм работы меню.

3. Подготовить алгоритм (блок-схему) работы программы.

4. Подготовить текст программы, провести отладку программы, используя возможности

встроенного отладчика интегрированной среды программирования.

5. Подготовить два варианта программы: в виде отдельных функций, расположенных в одном файле; в виде библиотеки функций, расположенных в разных файлах.

6. Подготовить отчёт о проделанной работе.

**Текст программы**

**Файл aligment.c**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "files.h"

#include "console.h"

#define STR\_SIZE ( 100 )

#define STR\_SCANF ( STR\_SIZE - 1 )

#define TEMPL\_SIZE ( 20 )

void alignment( FILE \* file, int spaceValue )//var1 alignment

{

int res;

char str[ STR\_SIZE ];

char templateAligment[ TEMPL\_SIZE ];//to read until new line

fseek( file, 0, SEEK\_SET );

sprintf( templateAligment, "%%%i[^\n]", STR\_SCANF );

if( check\_space( file ) )

put\_space( spaceValue );

res = fscanf( file, templateAligment, str );

for( ; res >= 1; )

{

printf( "%s\n", str );

fseek( file, 2, SEEK\_CUR );

if( check\_space( file ) )

put\_space( spaceValue);

res = fscanf( file, templateAligment, str );

}

}

void alignmentChoose( FILE \* file\_first, FILE \* file\_second, int argc,

char \*\*argv, int \*number )

{

int numFile, length;

char str[ STR\_SIZE ];

if( \*number + 1 > argc )

printf("\nEnter file number: ");

scanfCMD( str, argc, argv, number );//count the file number

numFile = atoi( str );

if( \*number + 1 > argc )

printf("\nEnter the length of the red string: ");

scanfCMD( str, argc, argv, number );//count the length of the red string

length = atoi( str );

if( numFile == 1 )

alignment( file\_first, length );

else

alignment( file\_second, length );

}

**Файл aligment.h**

#ifndef \_FILEALI\_H\_

#define \_FILEALI\_H\_

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define STR\_SIZE ( 100 )

#define STR\_SCANF ( STR\_SIZE - 1 )

#define TEMPL\_SIZE ( 20 )

void alignment( FILE \* file, int spaceValue );//var1 alignment

void alignmentChoose( FILE \* file\_first, FILE \* file\_second, int argc,

char \*\*argv, int \*number );

#endif

**Файл console.c**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define STR\_SIZE ( 100 )

#define STR\_SCANF ( STR\_SIZE - 1 )

#define TEMPL\_SIZE ( 20 )

#define TEMPL\_OUT ( 15 )

#define ERROR\_OPEN ( 1 )

#define STAR\_MAX ( 55 )

void put\_remaining( FILE \* file, int res, char \* str )//prints characters after the end of the second file

{

char templateSTR[ TEMPL\_SIZE ];//to read a given number of characters

sprintf( templateSTR, "%%%is", STR\_SCANF );

for( ; res >= 1; )

{

printf( "%s", str );

put\_unprintf( file );

res = fscanf( file, templateSTR, str );

}

}

void put\_unprintf( FILE \* file )//prints unprinted characters

{

char str[ STR\_SIZE ];

fscanf( file, "%[ ' ', '\n', '\t' ]", str );

printf( "%s", str );

}

int check\_space( FILE \* file )//checks for a space in a string

{

char symbol;

symbol = fgetc( file );

if( symbol == ' ' )

return 1;

fseek( file, -1, SEEK\_CUR );

return 0;

}

void put\_space( int value )

{

int i;

for( i = 0; i < value; i++ )

printf( " " );

}

void putWORD( char \*word, int value )

{

char templateOUT[ TEMPL\_SIZE ];//to put word

sprintf( templateOUT, "%%-%is", TEMPL\_OUT );

printf( templateOUT, word );

printf( "%-3i ", value );

put\_STAR( value );

}

void put\_STAR( int value )

{

int i;

for( i = 0; i < value; i++ )

{

if( i > STAR\_MAX )

break;

printf( "\*" );

}

printf( "\n" );

}

//returns a pointer to the read line from CMD or menu

void scanfCMD( char \*str, int argc, char \*\*argv, int \*number )

{

char templateSTR[ TEMPL\_SIZE ];

sprintf( templateSTR, "%%%is", STR\_SCANF );

if( \*number + 1 > argc )//reading a line from menu

{

scanf( templateSTR, str );

}

else

{

strcpy( str, argv[ \*number ] );

( \*number )++;

}

}

**Файл console.h**

#ifndef \_FILECMD\_H\_

#define \_FILECMD\_H\_

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define STR\_SIZE ( 100 )

#define STR\_SCANF ( STR\_SIZE - 1 )

#define TEMPL\_SIZE ( 20 )

#define TEMPL\_OUT ( 15 )

#define ERROR\_OPEN ( 1 )

void put\_remaining( FILE \* file, int res, char \* str );//prints characters after the end of the second file

void put\_unprintf( FILE \* file );//prints unprinted characters

int check\_space( FILE \* file );//checks for a space in a string

void put\_space( int value );

void putWORD( char \*word, int value );

void put\_STAR( int value );

void scanfCMD( char \*str, int argc, char \*\*argv, int \*number );//returns a pointer to the read line from CMD or menu

#endif

**Файл files.c**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define STR\_SIZE ( 100 )

#define STR\_SCANF ( STR\_SIZE - 1 )

#define ERROR\_OPEN ( 1 )

void check\_file( FILE \* file )

{

if( file == NULL )

{

printf( "File doesn't found\n" );

exit( ERROR\_OPEN );

}

}

void find\_files( FILE \*\* file, char \*str, int argc, char \*\*argv,

int \*number )

{

if( strcmp( str, "-EXIST\0" ) == 0 || \*str == 'E' )

{

file\_exist( file, argc, argv, number );

}

else if( strcmp( str, "-NEW\0" ) == 0 || \*str == 'N' )

{

file\_create( file, argc, argv, number );

}

else

\*file = NULL;

}

void file\_exist( FILE \*\* file, int argc, char \*\*argv, int \*number )

{

char name[ STR\_SIZE ];

if( \*number + 1 > argc )

printf("\nEnter the name of exist file: ");

scanfCMD( name, argc, argv, number );

\*file = fopen( name, "r" );

check\_file( \*file );

}

void file\_create( FILE \*\*file, int argc, char \*\*argv, int \*number )

{

char name[ STR\_SIZE ];

char str[ STR\_SIZE ];

if( \*number + 1 > argc )

printf("\nEnter the name of new file: ");

scanfCMD( name, argc, argv, number );

\*file = fopen( name, "w+" );

check\_file( \*file );

if( \*number + 1 > argc )

printf("\nFill out the file, the word STOP stops entering\n ");

scanfCMD( str, argc, argv, number );

for( ; strcmp( str, "STOP\0" ) != 0; )

{

fprintf( \*file, "%s ", str);

scanfCMD( str, argc, argv, number );

}

fseek( \*file, 0, SEEK\_SET);

}

void operation( FILE \* file\_first, FILE \* file\_second, char \*str,

int argc, char \*\*argv, int \*number )

{

int numFile;

if( strcmp( str, "-MIX\0" ) == 0 )

mixing( file\_first, file\_second );

if( strcmp( str, "-SEARCH\0" ) == 0 )

{

searchChoose( file\_first, file\_second, argc, argv, number );

}

if( strcmp( str, "-ALIGN\0" ) == 0)

{

alignmentChoose( file\_first, file\_second, argc, argv, number);

}

if( strcmp( str, "-STAT\0" ) == 0 )

{

if( \*number + 1 > argc )

printf("\nEnter file number: ");

scanfCMD( str, argc, argv, number );//count the file number

numFile = atoi( str );

if( numFile == 1 )

statistick( file\_first );

else

statistick( file\_second );

}

}

**Файл files.h**

#ifndef \_FILE\_H\_

#define \_FILE\_H\_

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define STR\_SIZE ( 100 )

#define STR\_SCANF ( STR\_SIZE - 1 )

#define ERROR\_OPEN ( 1 )

void check\_file( FILE \* file );//file open success

void find\_files( FILE \*\* file, char \*str, int argc, char \*\*argv, int \*number );//add file in programm

void file\_exist( FILE \*\* file, int argc, char \*\*argv, int \*number );//open exist file

void file\_create( FILE \*\*file, int argc, char \*\*argv, int \*number );//create new file

void operation( FILE \*file\_first, FILE \*file\_second, char \*str, int argc, char \*\*argv, int \*number );

#endif

**Файл mixing.c**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "files.h"

#include "console.h"

#define STR\_SIZE ( 100 )

#define STR\_SCANF ( STR\_SIZE - 1 )

#define TEMPL\_SIZE ( 20 )

void mixing( FILE \* src\_first, FILE \* src\_second )//var6 mixing

{

int i, res1, res2 = 1;

char str[ STR\_SIZE ], templateSTR[ TEMPL\_SIZE ];//to read a given number of characters

fseek( src\_first, 0, SEEK\_SET );

fseek( src\_second, 0, SEEK\_SET );

sprintf( templateSTR, "%%%is", STR\_SCANF );

printf( "\n" );

put\_unprintf( src\_first );

res1 = fscanf( src\_first, templateSTR, str );

for( i = 0; res1 >= 1 && res2 >= 1; i++ )

{

printf( "%s", str );

put\_unprintf( src\_first );

if( i % 2 == 0 )

{

res1 = fscanf( src\_first, templateSTR, str );

res2 = fscanf( src\_second, templateSTR, str );

}

else

{

res2 = fscanf( src\_second, templateSTR, str );

res1 = fscanf( src\_first, templateSTR, str );

}

}

put\_remaining( src\_first, res1, str );//prints characters after the end of the second file

printf( "\n" );

}

**Файл mixing.h**

#ifndef \_FILEMIX\_H\_

#define \_FILEMIX\_H\_

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define STR\_SIZE ( 100 )

#define STR\_SCANF ( STR\_SIZE - 1 )

#define TEMPL\_SIZE ( 20 )

void mixing( FILE \* src\_first, FILE \* src\_second );//var6 mixing

#endif

**Файл search.c**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "files.h"

#include "console.h"

#define STR\_SIZE ( 100 )

#define STR\_SCANF ( STR\_SIZE - 1 )

#define TEMPL\_SIZE ( 20 )

void search( FILE \* file, int length )//var5 search

{

char templateSTR[ TEMPL\_SIZE ];//to read a given number of characters

int res, position = 0;

char str[ STR\_SIZE ];

sprintf( templateSTR, "%%%is", STR\_SCANF );

fseek( file, 0, SEEK\_SET);

printf( "\n" );

res = fscanf( file, templateSTR, str );

for( ; res >= 1; )

{

if( strlen( str ) == length )

{

position = ftell( file ) - length + 1;

printf( "Position: %3i, word: %s\n", position, str );

}

res = fscanf( file, templateSTR, str );

}

if( position == 0 )

printf( "No words of a given length\n" );

}

void searchChoose( FILE \* file\_first, FILE \* file\_second, int argc, char \*\*argv, int \*number )

{

int numFile, length;

char str[ STR\_SIZE ];

char templateSTR[ TEMPL\_SIZE ];

sprintf( templateSTR, "%%%is", STR\_SCANF );

if( \*number + 1 > argc )

printf("\nEnter file number: ");

scanfCMD( str, argc, argv, number );//count the file number

numFile = atoi( str );

if( \*number + 1 > argc )

printf("\nEnter the word length: ");

scanfCMD( str, argc, argv, number );//count the word length

length = atoi( str );

if( numFile == 1 )

search( file\_first, length );

else

search( file\_second, length );

}

**Файл search.h**

#ifndef \_FILESEARCH\_H\_

#define \_FILESEARCH\_H\_

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define STR\_SIZE ( 100 )

#define STR\_SCANF ( STR\_SIZE - 1 )

#define TEMPL\_SIZE ( 20 )

void search( FILE \* file, int length );//var5 search

void searchChoose( FILE \* file\_first, FILE \* file\_second, int argc, char \*\*argv, int \*number );

#endif

**Файл statistick.c**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "files.h"

#include "console.h"

#define STR\_SIZE ( 100 )

#define STR\_SCANF ( STR\_SIZE - 1 )

#define TEMPL\_SIZE ( 20 )

#define TEMPL\_OUT ( 15 )

#define ERROR\_OPEN ( 1 )

void statistick( FILE \* file )

{

int all, different;

all = allWords( file );

different = diffWords( file );

printf( "All words: %i\n", all );

printf( "Different words: %i\n\n", different );

histogram( file );

}

int allWords( FILE \* file )//count all words

{

int count = 0, res;

char str[ STR\_SIZE ];

char templateSTR[ TEMPL\_SIZE ];//read a given number of characters

sprintf( templateSTR, "%%%is", STR\_SCANF );

fseek( file, 0, SEEK\_SET );

res = fscanf( file, templateSTR, str );

for( ; res >= 1; )

{

count++;

res = fscanf( file, templateSTR, str );

}

return count;

}

int diffWords( FILE \* file )//count different words

{

int count = 0, pos, res;

char word[ STR\_SIZE ];

char templateSTR[ TEMPL\_SIZE ];//read a given number of characters

sprintf( templateSTR, "%%%is", STR\_SCANF );

fseek( file, 0, SEEK\_SET );

res = fscanf( file, templateSTR, word );

for( ; res >= 1; )

{

pos = ftell( file );

count += wordNEW( file, word, pos ); //plus 1 if word is new

fseek( file, pos, SEEK\_SET );//return the cursor to the pos

res = fscanf( file, templateSTR, word );

}

return count;

}

void histogram( FILE \* file )

{

int count = 0, pos, res;

char word[ STR\_SIZE ];

char templateSTR[ TEMPL\_SIZE ];//read a given number of characters

sprintf( templateSTR, "%%%is", STR\_SCANF );

fseek( file, 0, SEEK\_SET );

res = fscanf( file, templateSTR, word );

for( ; res >= 1; )

{

pos = ftell( file );

if( wordNEW( file, word, pos ) )

{

count = wordValue( file, word );

putWORD( word, count );

}

fseek( file, pos, SEEK\_SET );//return cursor in old position

res = fscanf( file, templateSTR, word );

}

}

//return how many times does the word occur

int wordValue( FILE \* file, char \*word

{

int res, count = 0;

char newWORD[ STR\_SIZE ];

char templateSTR[ TEMPL\_SIZE ];//to read a given number of characters

sprintf( templateSTR, "%%%is", STR\_SCANF );

fseek( file, 0, SEEK\_SET );

res = fscanf( file, templateSTR, newWORD );

for( ; res >= 1; )

{

if( strcmp( word, newWORD ) == 0 )

count++;

res = fscanf( file, templateSTR, newWORD );

}

return count;

}

//return 1 if the word met before

int wordNEW( FILE \* file, char \*word, int endPOS )

{

int res, pos;

char newWORD[ STR\_SIZE ];

char templateSTR[ TEMPL\_SIZE ];//read a given number of characters

sprintf( templateSTR, "%%%is", STR\_SCANF );

fseek( file, 0, SEEK\_SET );

res = fscanf( file, templateSTR, newWORD );

for( ; res >= 1 ; )

{

if( strcmp( word, newWORD ) == 0 )

break;

res = fscanf( file, templateSTR, newWORD );

}

pos = ftell( file );

if( pos >= endPOS )

return 1;

return 0;

}

**Файл statistick.h**

#ifndef \_FILESTAT\_H\_

#define \_FILESTAT\_H\_

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define STR\_SIZE ( 100 )

#define STR\_SCANF ( STR\_SIZE - 1 )

#define TEMPL\_SIZE ( 20 )

#define TEMPL\_OUT ( 15 )

#define ERROR\_OPEN ( 1 )

void statistick( FILE \* file );

int allWords( FILE \* file );//count all words

int diffWords( FILE \* file );//count different words

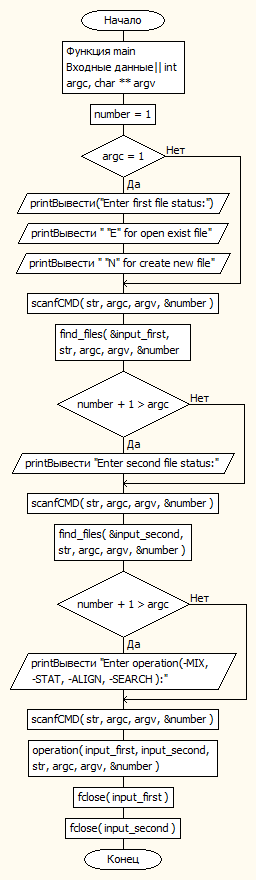
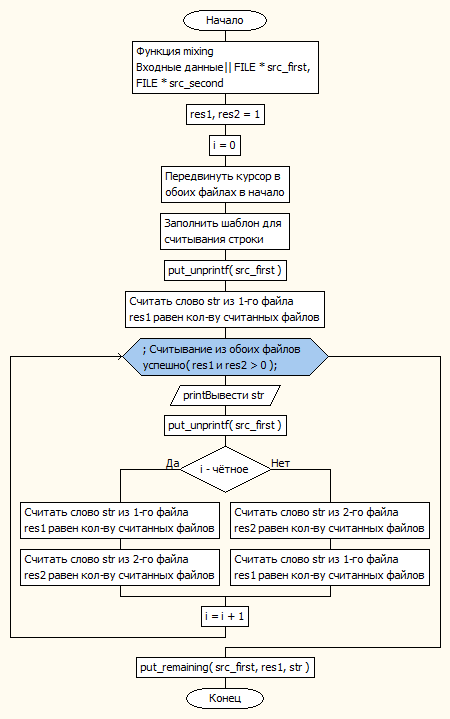
void histogram( FILE \* file );

int wordValue( FILE \* file, char \*word );//return how many times does the word occur

int wordNEW( FILE \* file, char \*word, int endPOS );//return how many times does the word occur

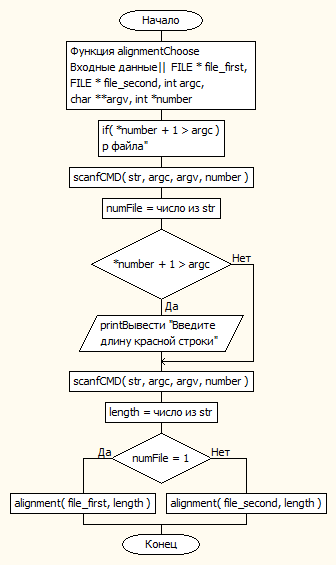
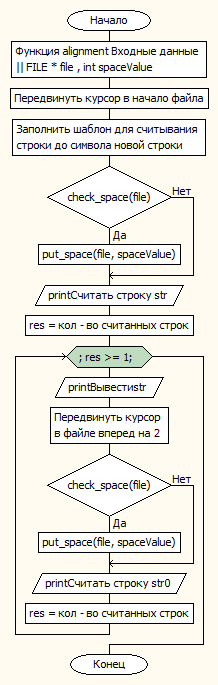
#endif

**Блок-схема алгоритма**



Функция main( int, char \*\* )

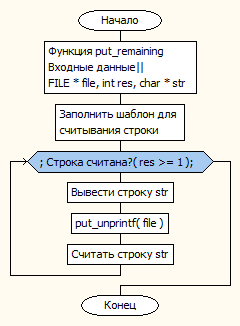
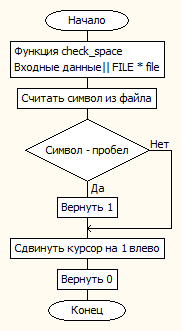
Функция mixing( FILE \*, FILE \* )

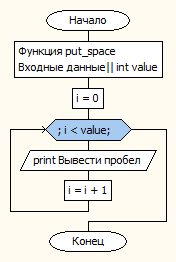
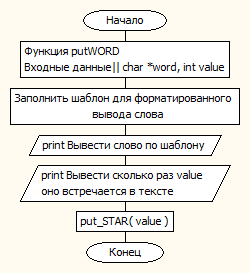
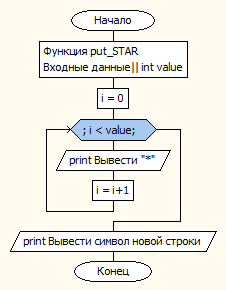
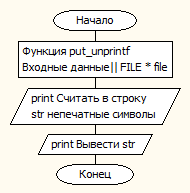


Функция alignmentChoose

(FILE \*, FILE \*, int, char \*\*, int \* )

Функция alignment( FILE \*, int )





Функция putWORD( char \*, int )

Функция put\_STAR( int )

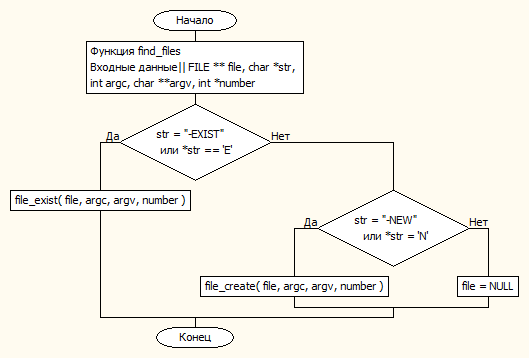
Функция put\_unprintf( FILE \* )

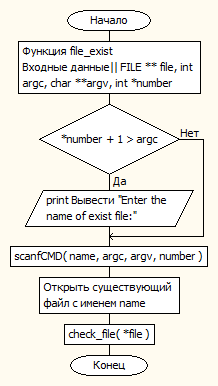
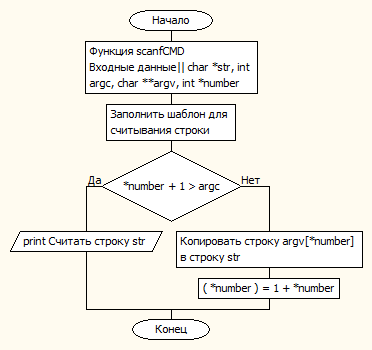
Функция put\_remaining

(FILE \*, int, char \*\* )

Функция check\_space( FILE \* )

Функция put\_space( int )





Функция file\_exist

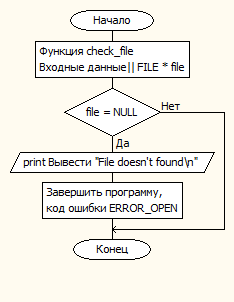
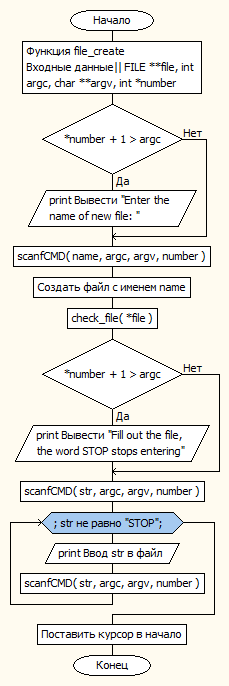
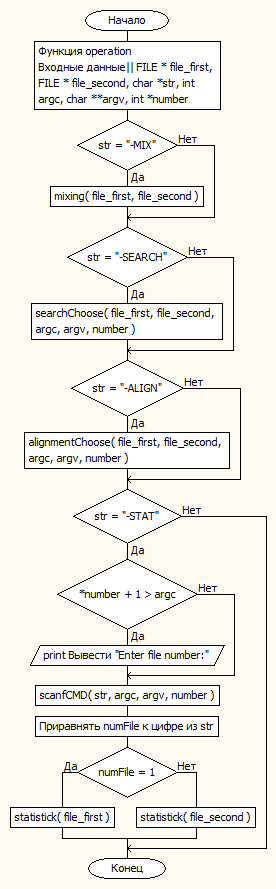
( FILE \*\*, int, char \*\*, int \* )

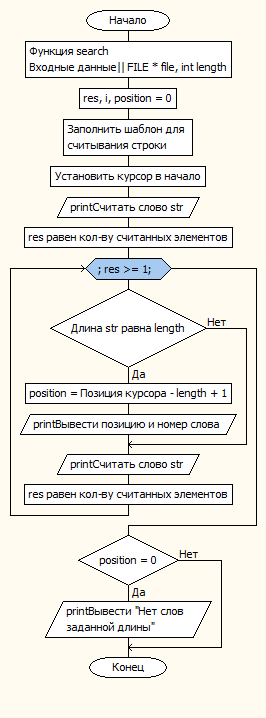
Функция scanfCMD

( char \*, int, char \*\*, int \* )

Функция find\_files

( FILE \*\*, char \*, int, char \*\*, int \* )





Функция search( FILE \*, int )

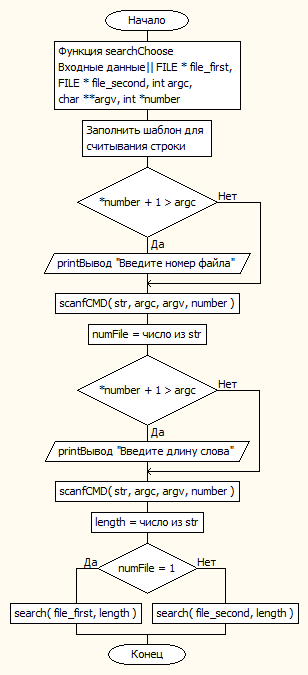
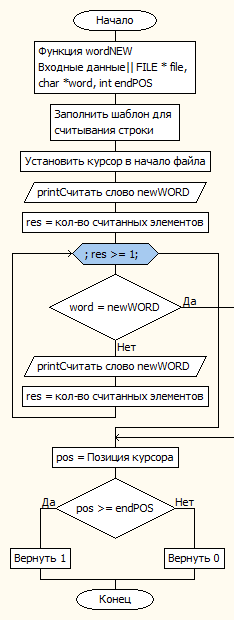
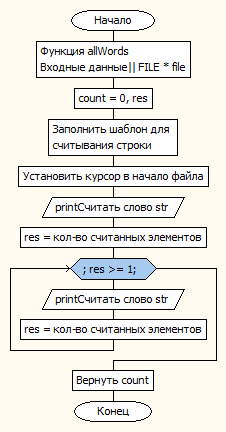
Функция operation

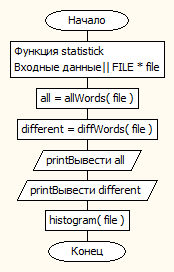
( FILE \*,FILE \*,char \*, int, char \*\*, int\* )

Функция file\_create

( FILE \*\*, int, char \*\*, int \* )

Функция check\_file( FILE \* )





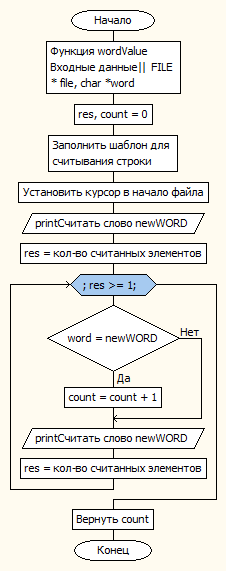
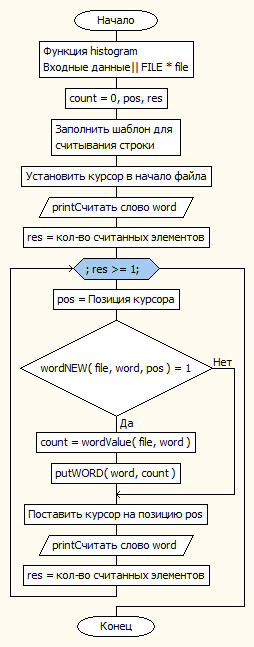
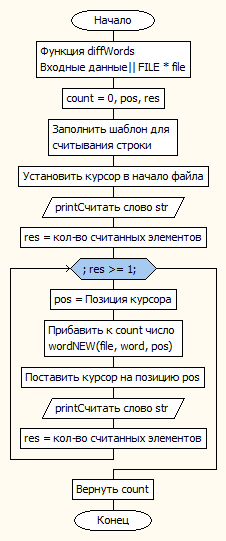
Функция wordNEW( FILE\*, char\*, int )

Функция searchChoose

(FILE \*, FILE \*, int, char \*\*, int \* )

Функция statistick( FILE \* )

Функция allWords( FILE \* )



Функция wordValue

( FILE \*, char \* )

Функция histogram( FILE \* )

Функция diffWords( FILE \* )

**Данные текстовых файлов**

**1.txt**

Somebody once told me the world is gonna roll me

I ain't the sharpest tool in the shed

She was looking kind of dumb with her finger and her thumb

In the shape of an "L" on her forehead.

**2.txt**

Napalm, son. Nothing else in the world smells like that.

I love the smell of napalm in the morning. You know, one time we had a hill bombed, for 12 hours. When it was all over, I walked up. We didn’t find one of 'em, not one stinkin' dink body. The smell, you know that gasoline smell, the whole hill. Smelled like… victory.

Someday this war’s gonna end…

**Результат работы программы**

**Смешивание** ( Аргументы: -EXIST 1.txt –EXIST 2.txt –MIX )

Somebody Napalm, told Nothing the in is world roll like

I I the the tool of the in

She morning. looking know, of time with had finger hill her for

In hours. shape it an all on I forehead.

**Выравнивание** ( Аргументы: -EXIST 1.txt –EXIST 2.txt –ALIGN 1 4 )

Somebody once told me the world is gonna roll me

I ain't the sharpest tool in the shed

She was looking kind of dumb with her finger and her thumb

In the shape of an "L" on her forehead.

**Поиск** ( Аргументы: -EXIST 1.txt –EXIST 2.txt –SEARCH 1 5 )

Position: 28, word: world

Position: 37, word: gonna

Position: 54, word: ain't

Position: 145, word: thumb

Position: 159, word: shape

**Статистика** ( Аргументы: -EXIST 1.txt –EXIST 2.txt –STAT 1 )

All words: 68

Different words: 59

Napalm, 1 \*

son. 1 \*

Nothing 1 \*

else 1 \*

in 2 \*\*

the 4 \*\*\*\*

world 1 \*

smells 1 \*

like 1 \*

that. 1 \*

I 2 \*\*

love 1 \*

smell 1 \*

of 2 \*\*

napalm 1 \*

morning. 1 \*

You 1 \*

know, 1 \*

one 3 \*\*\*

time 1 \*

we 1 \*

had 1 \*

a 1 \*

hill 1 \*

bombed, 1 \*

for 1 \*

12 1 \*

hours. 1 \*

When 1 \*

it 1 \*

was 1 \*

all 1 \*

over, 1 \*

walked 1 \*

up. 1 \*

We 1 \*

didnТt 1 \*

find 1 \*

'em, 1 \*

not 1 \*

stinkin' 1 \*

dink 1 \*

body. 1 \*

The 1 \*

smell, 2 \*\*

you 1 \*

know 1 \*

that 1 \*

gasoline 1 \*

whole 1 \*

hill. 1 \*

Smelled 1 \*

likeЕ 1 \*

victory. 1 \*

Someday 1 \*

this 1 \*

warТs 1 \*

gonna 1 \*

endЕ 1 \*

**Заключение**

Целью работы - научиться работать с файлами. В результате работы создана программа для обработки текстовых файлов на 480 строк, соответствующая всем требованиям; получены знания для работы с файлами и командной строкой. Недостатком программы является наличие функций, выполняющих одновременный ввод и вывод данных.