МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Московский Авиационный Институт»

(Национальный Исследовательский Университет)

Институт №8: «Информационные технологии и прикладная математика»

Кафедра 806: «Вычислительная математика и программирование»

**КУРСОВАЯ РАБОТА №6**

По курсу «Вычислительные системы»

**II** семестр

Тема:

**Разреженные матрицы«Обработка последовательной файловой структуры на языке Си»**

**Группа:** М80-106Б-22

**Студент:** Ларченко А.О

**Преподаватели:** Дубинин А.В.

**Оценка: \_\_\_\_\_\_\_\_\_ Подпись: \_\_\_\_\_\_\_\_**

**Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Москва, 2023

# Содержание

[**Содержание 2**](#_5bjxas8qpzef)

[**Введение 3**](#_uyifey50bwp5)

[**Программа 4**](#_9jo81si2g25l)

[Задание 4](#_bfqk89p80hab)

[Алгоритм 4](#_3h3ck1r78l1l)

[Описание файлов и функций 5](#_ivl8bapzslky)

[**Заключение 23**](#_be10rxpujcie)

# Введение

6 Курсовой проект направлен на знакомство файловыми дескрипторами в языке си.

# Программа

## Задание

Разработать последовательную структуру данных для представления простейшей базы данных на файлах в СП Си в соответствии с заданным вариантом. Составить программу генерации внешнего нетекстового файла заданной структуры, содержащего представительный набор записей (не менее 20). Распечатать содержимое сгенерированного файла в виде таблицы и выполнить над ним заданное действие для 3–5 значений параметров запроса p и распечатать результат.

*Структура :* информация о рейсах и пассажирах

*Доп действия: 1)печать файла, 2)запрос информации из базы и сортировка по указанным критериям 3) выбор пассажира, и его место прибытия, вывод рейсов из этого города в этот день*

***Немного теории***

Файловый дескриптор — это неотрицательное целое число. Когда создается новый поток ввода-вывода, ядро возвращает процессу, создавшему поток ввода-вывода, его файловый дескриптор. Ко всем потокам ввода-вывода можно получить доступ через так называемые файловые дескрипторы.

Бинарный файл - файл, в котором информация записана при помощи последовательности байт.

## Алгоритм

На вход программе пользователь передает 2 файла исходный текстовый с таблицей данных, и выходной бинарный. Программа преобразует текстовый файл в бинарный, после чего работает именно с бинарным файлом, т.к. это позволяет нам осуществлять доступ к элементам файла за О(1).

Чтение файла - О(n), n - количество строк в текстовом файле. Все функции тоже за О(n).

\*Прикольная штука fseek с доступом к элементу файла за О(1)\*

## Описание файлов и функций

table.txt - табличка данных

Ivanov I.S. M NO Aeroflot SU1886 26.05.23 10.15 Moscow Saint-Petrburg

Malkin A.M. M YES Pegas\_Air PA1290 24.07.23 10.15 Moscow Istambul

Protosoviskiy N.V. M YES Aeroflot SU1886 26.05.23 10.15 Moscow Saint-Petrburg

Protosoviskaya S.R. F NO Aeroflot SU1886 01.06.23 10.15 Moscow Saint-Petrburg

Nikulin B.G. M YES Aeroflot SU1893 26.05.23 18.00 Moscow Kazan

Obozov V.G. M YES Lot MD8761 28.05.23 10.10 Kishinev Berlin

Lebedev A.A. M YES Ural\_Airlines UA6798 09.05.23 23.07 Moscow Saint-Petrburg

Suchno A.A. M NO Air\_India AI1451 26.05.23 16.00 Goa Saint-Petrburg

Suchno A.V. F YES Air\_India AI1451 26.05.23 16.00 Goa Saint-Petrburg

Averin S.A. M NO Dubai\_Air DU4547 06.06.23 21.45 Roma London

Malevich V.N. M YES Aeroflot SU1886 26.05.23 10.15 Moscow Saint-Petrburg

Korotkevich V.S. F YES Aeroflot SU1999 26.05.23 13.26 Saint-Petrburg Vladivostok

Akinfeev I.V. M NO Pobeda PO1267 06.05.23 04.05 Moscow Barselona

Ignatov A.O. M YES Aeroflot SU1886 26.05.23 10.15 Moscow Saint-Petrburg

Romanov R.P. M YES Ryanair FR1886 23.05.23 03.17 Venice Helsinki

Kommisarenko V.S. M YES Belavia SU1886 28.06.23 16.40 Moscow Minsk

Tamochkin N.N. M NO Turkish\_Airlines TK9081 26.05.23 10.15 Istambul Saratov

Pushkina O.V. F NO Emirates AE1886 29.05.23 19.55 Moscow London

flight/h - структура данных

#include <string.h>

#ifndef flights\_h

#define flights\_h

#define N 10

#define SIZE 50

const char \*names[N]={"Sername", "Initials", "Sex", "Baggage", "Air Company", "Flight", "Date", "Time", "From", "To"};

char\* Air\_company[SIZE];

char\* Flight[SIZE];

char\* Dates[SIZE];

char\* Times[SIZE];

char\* From[SIZE];

char\* To[SIZE];

typedef struct flights{

char sername[21];

char initials[6];

char sex[3];

char baggage[5];

char company[21];

char flight[9];

char date[10];

char time[7];

char from[21];

char to[21];

} flights;

#endif

Txt\_to\_bin.c - преобразование текстового в бинарный

#include <stdio.h>

#include "flights.h"

int line\_counter(char \*arg){

int cnt=0;

FILE \*tmp=fopen(arg, "r");

while (!feof(tmp)){

fscanf(tmp, "%\*[^\n]%\*c");

cnt++;

}

fclose(tmp);

return cnt;

}

int main(int argc, char \*argv[]){

if (argc != 3){

fprintf(stderr, "Error! Uncorrect input");

return 1;

}

FILE\* input = fopen(argv[1], "rw");

FILE\* output = fopen(argv[2], "wb");

if (input==NULL){

fprintf(stderr, "Error! There is no input file");

return 2;

}

if (output==NULL){

fprintf(stderr, "Error! There is no output file");

return 3;

}

int count= line\_counter(argv[1]);

if(count<=1){

fprintf(stderr, "Error! There is empty txt-file");

return 4;

}

printf("counts = %d \n", count);

flights fl;

while(fscanf(input, "%s%s%s%s%s%s%s%s%s%s", fl.sername, fl.initials, fl.sex, fl.baggage, fl.company, fl.flight, fl.date, fl.time, fl.from, fl.to)!=EOF){

fwrite(&fl, sizeof(flights), 1, output);

}

fclose(input);

fclose(output);

return 0;

}

//Фамилия инициалы пол багаж Авиакомпания Рейс Дата Время Откуда Куда

main.c - основной код

#include <stdio.h>

#include <stdlib.h>

#include "flights.h"

int line\_counter(FILE\* tmp){

int cnt=0;

fseek(tmp, 0, SEEK\_SET);

while (!feof(tmp)){

fscanf(tmp, "%\*[^\n]%\*c");

cnt++;

}

return cnt;

}

static void no\_val\_print(char\* arr[], int size){

char str[]="No\_val";

for (int i=0; i< size; ++i){

arr[i]=(char\*)malloc(21);

strcpy(arr[i], "No\_val");

}

}

void line\_print(flights fl){

printf("%s\t", fl.sername);

printf("%s\t", fl.initials);

printf("%s ", fl.sex);

printf("%s\t", fl.baggage);

printf("%s ", fl.company);

printf("%s ", fl.flight);

printf("%s ", fl.date);

printf("%s ", fl.time);

printf("%s ", fl.from);

printf("%s ", fl.to);

printf("\n");

}

void t\_print(FILE\* f){

flights fl;

int cnt=1;

fseek(f,0, SEEK\_SET);

for(int i=0; i<N; ++i){

printf(" %s |", names[i]);

}

printf("\n");

while(fread(&fl, sizeof(flights), 1, f)){

printf("%d) ", cnt);

line\_print(fl);

cnt++;

}

}

void create\_dictionary(const int size, int &cnt, char \*arr[], char tmp[]){

for(int i=0;i<=size;++i){

if (strcmp(arr[i], "No\_val")==0){

strcpy(arr[i], tmp);

cnt++;

return;

} else if(strcmp(arr[i], tmp)==0){

return;

}

}

}

void interest(FILE\* f){

// FILE\* f=fopen(arg, "rb");

fseek(f,0, SEEK\_SET);

printf("Enter number of interesting column for you\n");

for(int i=2; i<N; ++i){

printf("%d)%s ",i-1, names[i]);

}

printf("\n");

int request;

flights fl;

scanf("%d", &request);

if (request==1){

printf("Choose 1)M or 2)F\n");

scanf("%d", &request);

while(fread(&fl, sizeof(flights), 1, f)){

if(request==1 && strcmp(fl.sex, "M")==0){

line\_print(fl);

} else if( request==2 && strcmp(fl.sex, "F")==0){

line\_print(fl);

}

}

} else if(request==2){

printf("Choose 1)YES or 2)NO\n");

scanf("%d", &request);

while(fread(&fl, sizeof(flights), 1, f)){

if(request==1 && strcmp(fl.baggage, "YES")==0){

line\_print(fl);

} else if( request==2 && strcmp(fl.baggage, "NO")==0){

line\_print(fl);

}

}

} else if(request==3){

printf("Choose ");

fseek(f,0, SEEK\_SET);

int size=0;

no\_val\_print(Air\_company, SIZE);

while(fread(&fl, sizeof(flights), 1, f)){

char tmp[21];

strcpy(tmp, fl.company);

create\_dictionary(SIZE, size, Air\_company, tmp);

}

// printf("%d ,", size);

for(int i=0; i<size; ++i){

printf("%d) %s ",i+1, Air\_company[i]);

}

printf("\n");

scanf("%d", &request);

char search[21];

strcpy(search, Air\_company[request-1]);

fseek(f,0, SEEK\_SET);

while(fread(&fl, sizeof(flights), 1, f)){

if(strcmp(fl.company, search)==0){

line\_print(fl);

}

}

}else if(request==4){

printf("Choose ");

fseek(f,0, SEEK\_SET);

int size=0;

no\_val\_print(Flight, SIZE);

while(fread(&fl, sizeof(flights), 1, f)){

char tmp[9];

strcpy(tmp, fl.flight);

create\_dictionary(SIZE, size, Flight, tmp);

}

// printf("%d ,", size);

for(int i=0; i<size; ++i){

printf("%d) %s ",i+1, Flight[i]);

}

printf("\n");

scanf("%d", &request);

char search[9];

strcpy(search, Flight[request-1]);

fseek(f,0, SEEK\_SET);

while(fread(&fl, sizeof(flights), 1, f)){

if(strcmp(fl.flight, search)==0){

line\_print(fl);

}

}

}else if(request==5){

printf("Choose ");

fseek(f,0, SEEK\_SET);

int size=0;

no\_val\_print(Dates, SIZE);

while(fread(&fl, sizeof(flights), 1, f)){

char tmp[10];

strcpy(tmp, fl.date);

create\_dictionary(SIZE, size, Dates, tmp);

}

// printf("%d ,", size);

for(int i=0; i<size; ++i){

printf("%d) %s ",i+1, Dates[i]);

}

printf("\n");

scanf("%d", &request);

char search[10];

strcpy(search, Dates[request-1]);

fseek(f,0, SEEK\_SET);

while(fread(&fl, sizeof(flights), 1, f)){

if(strcmp(fl.date, search)==0){

line\_print(fl);

}

}

}else if(request==6){

printf("Choose ");

fseek(f,0, SEEK\_SET);

int size=0;

no\_val\_print(Times, SIZE);

while(fread(&fl, sizeof(flights), 1, f)){

char tmp[10];

strcpy(tmp, fl.time);

create\_dictionary(SIZE, size, Times, tmp);

}

// printf("%d ,", size);

for(int i=0; i<size; ++i){

printf("%d) %s ",i+1, Times[i]);

}

printf("\n");

scanf("%d", &request);

char search[10];

strcpy(search, Times[request-1]);

fseek(f,0, SEEK\_SET);

while(fread(&fl, sizeof(flights), 1, f)){

if(strcmp(fl.time, search)==0){

line\_print(fl);

}

}

}else if(request==7){

printf("Choose ");

fseek(f,0, SEEK\_SET);

int size=0;

no\_val\_print(From, SIZE);

while(fread(&fl, sizeof(flights), 1, f)){

char tmp[10];

strcpy(tmp, fl.from);

create\_dictionary(SIZE, size, From, tmp);

}

// printf("%d ,", size);

for(int i=0; i<size; ++i){

printf("%d) %s ",i+1, From[i]);

}

printf("\n");

scanf("%d", &request);

char search[10];

strcpy(search, From[request-1]);

fseek(f,0, SEEK\_SET);

while(fread(&fl, sizeof(flights), 1, f)){

if(strcmp(fl.from, search)==0){

line\_print(fl);

}

}

}else if(request==8){

printf("Choose ");

fseek(f,0, SEEK\_SET);

int size=0;

no\_val\_print(To, SIZE);

while(fread(&fl, sizeof(flights), 1, f)){

char tmp[10];

strcpy(tmp, fl.to);

create\_dictionary(SIZE, size, To, tmp);

}

// printf("%d ,", size);

for(int i=0; i<size; ++i){

printf("%d) %s ",i+1, To[i]);

}

printf("\n");

scanf("%d", &request);

char search[10];

strcpy(search, To[request-1]);

fseek(f,0, SEEK\_SET);

while(fread(&fl, sizeof(flights), 1, f)){

if(strcmp(fl.to, search)==0){

line\_print(fl);

}

}

}else {

printf("Uncorrect input. Try again");

}

}

void counting(FILE\* f){

t\_print(f);

fseek(f,0, SEEK\_SET);

printf("\nIn this program you can choose line and it will be output flight at this day from this city\n");

printf("Enter interesting string\n");

int request;

flights fl;

scanf("%d", &request);

fseek(f, sizeof(flights)\*(request-1), SEEK\_SET);

fread(&fl, sizeof(flights), 1, f);

line\_print(fl);

fseek(f,0, SEEK\_SET);

flights tmp;

while(fread(&tmp, sizeof(flights), 1, f)){

if(strcmp(tmp.from, fl.to)==0 &&strcmp(tmp.date, fl.date)==0){

printf("%s %s %s %s %s %s", tmp.company, tmp.flight, tmp.date, tmp.time, tmp.from, tmp.to );

}

}

}

int main(int argc, char\* argv[]){

if (argc != 3){

fprintf(stderr, "Error! Uncorrect input");

return 1;

}

FILE\* input = fopen(argv[1], "rb");

if (input==NULL){

fprintf(stderr, "Error! There is no input file");

return 2;

}

char\* c=argv[2];

if(strcmp(c, "p")==0){

t\_print(input);

} else if(strcmp(c, "i")==0){

interest(input);

} else if(strcmp(c, "c")==0){

counting(input);

} else {

printf("Uncorrect input. Try again");

}

printf("\n");

fclose(input);

return 0;

}

// p - печатает файл; i - узнает информацию о файле; с - интересное задание!

# 

# 

# Заключение

Благодаря этой работе я познакомился с файловым дескриптором языка си и научился с ним работать. Считаю, что с заданием справился успешно.