

**Московский Авиационный Институт  
(Национальный Исследовательский Университет)**

**Факультет информационных технологий и прикладной  
математики**

**Кафедра вычислительной математики и  
программирования**

**Курсовой проект VI  
по курсу «Практикум на ЭВМ»  
II семестр**

**«Обработка последовательной файловой системы на языке Си»**

<b>Студент</b>	<b>Сыроежкин Кирилл Геннадьевич</b>
<b>Группа</b>	<b>М8О-104Б-18</b>
<b>Руководитель</b>	<b>Доцент кафедры 806 Никулин С.П</b>
<b>Оценка</b>	
<b>Дата</b>	

**Москва 2019.**

## Содержание

1.Постановка задачи.....	3
2.Общий метод решения.....	3
3.Сведения о программе.....	3
4.Описание логической структуры программы.....	4
5. Переменные и константы.....	5
6. Входные данные.....	6
7. Выходные данные.....	6
8. Программа на СИ.....	7
9. Демонстрация работы программы.....	11
Заключение.....	16
Литература.....	16

## **1. Постановка задачи.**

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.

Действие по выборке данных из файла оформить в виде отдельной программы с параметрами запроса, вводимыми из стандартного входного тестового файла, или получаемыми из командной строки UNIX. Параметры задаются с помощью ключей -f (печать) и -p (параметры конкретного варианта задания).

### **1.1. Содержимое и структура файла (32-39)**

Информация о пассажирах аэропорта: фамилия, инициалы, количество вещей, общий вес вещей, пункт назначения, время вылета, наличие пересадок, сведения о детях.

### **1.2. Действия (35).**

Найти пассажиров, число вещей которых превосходит среднее число вещей не менее, чем на  $p$  штук.

## **2. Общий метод решения**

Работа идет с двумя программами на языке программирования СИ, одного заголовочного файла (содержит описание структуры), двумя текстовыми файлами (хранятся данные) и двумя бинарными файлами (туда записываются данные из текстовых файлов).

Первая программа читает данные из текстового файла и записывает их в бинарный.

Вторая программа производит печать данных из бинарного файла в виде таблицы (если указан ключ -f), или результат выполнения поставленной задачей (ключ -p).

## **3. Общие сведения о программе**

Имена файлов:

- 1) pass.h - заголовочный файл;
- 2) dump.c - программа конвертации файла в бинарный;
- 3) prog.c - основная программа;
- 4) file1.txt - входной текстовый файл 1 (для тестов);
- 5) file2.txt - входной текстовый файл 2 (для тестов);
- 6) file1.bin - выходной нетекстовый файл 1 (для тестов);
- 7) file2.bin - выходной нетекстовый файл 2 (для тестов);

Программное и аппаратное обеспечения для запуска данной программы на ПК не ограничено в выборе. Операционная система семейства Unix - Ubuntu. Язык программирования Си. Строк в программе prog.c: 97, dump.c: 28. В заголовочном файле pass.h: 15.

## 4. Описание логической структуры программы

### pass.h:

Объявлена структура со следующими атрибутами: **name** (фамилия), **initials** (инициалы), **num\_cloth** (количество вещей), **weight\_cloth** (вес вещей), **point** (точка назначения), **time** (время), **trans** (наличие пересадок), **childs** (информация о детях).

### dump.c:

Функция **main** должна принимать из терминала 2 параметра **argv[1]** (название исходного файла), **argv[2]** (название созданного нетекстового), а также параметр **argc**, который подсчитывает количество параметров **argv**. Здесь используется 2 функции: **war** и **binary**.

Функция **war(int argc)** принимает параметр, отвечающий за количество переданных параметров, и если параметры заданы неправильно, то выводит предупреждающее сообщение.

Функция **binary(char\* \*argv)**, в которую передаются массив **argv** (все переданные файлы в **main**), открывает текстовый файл, записанный в **argv[1]** и создает нетекстовый файл, записанный в **argv[2]**; затем в цикле с помощью функций **fscanf**, **fwrite**, пока не конец файла, происходит запись данных из текстового файла(**argv[1]**) в нетекстовый файл(**argv[2]**).

### prog.c:

Функция **main** должна принимать параметр **argv[1]** (название нетекстового файла, с которым будет проводиться дальнейшая работа), а также несколько

параметров на выбор: -f и -p(их можно использовать по отдельности или вместе). Параметр “-f” печатает таблицу, а параметр “-p” выполняет поставленное задание. Здесь используется всего 2 функции: **printtable**, **printwork**. О них мы поговорим позже.

Программа использует 3 вспомогательные функции: **table** (проверка наличия аргумента -f), **work** (проверка наличия аргумента -p), **medium** (находит среднее арифметическое общего кол-ва вещей).

Функция **table(int argc, char \*argv[])**, в которую передается **argc** (количество переданных аргументов в **main**) и массив **argv** (все переданные аргументы в **main**), с помощью параметрического цикла проверяет на совпадение любого параметра с “-f”. Если нашлось совпадение, то функция выводит 1, иначе 0.

Функция **work(int argc, char \*argv[])**, в которую передается **argc** (количество переданных аргументов в **main**) и массив **argv** (все переданные аргументы в **main**), с помощью параметрического цикла проверяет на совпадение любого параметра с “-p”. Если нашлось совпадение, то функция выводит 1, иначе 0.

Функция **medium(char\* argv)**, в которую передается значение одного из элементов массива **argv**(передается имя нетекстового файла **argv[1]**), открывает файл на чтение, а затем с помощью цикла и команды **fread** находит сумму всех **p.num\_cloth**(количество вещей у каждого пассажира), а также подсчитывается количество пассажиров(**n++**); затем файл закрывается и возвращается целочисленное деление суммы всех элементов на количество всех пассажиров.

Итак, теперь мы можем перейти к рассмотрению основных функций.

Функция **printtable(int argc, char\* \*argv)**, в которую передается **argc** (количество переданных аргументов в **main**) и массив **argv** (все переданные аргументы в **main**), проверяет наличие ключа “-f” с помощью функции **table**, и если он найден, то открывает файл(**argv[1]**) на чтение и с помощью нескольких циклов печатается таблица, далее файл закрывается.

Функция **printwork(int argc, char\* \*argv)**, в которую передается **argc** (количество переданных аргументов в **main**) и массив **argv** (все переданные аргументы в **main**), проверяет наличие ключа “-p”, с помощью функции **work**, и если он найден, то открывается файл(**argv[1]**) на чтение, переменной **count** присваивается значение параметра **p**(количество вещей, на которое должно превосходить количество вещей пассажира от среднего), задаваемое одним из аргументов в **main**. Далее с помощью цикла и условия проверяется выполнение задачи, и в зависимости от результата, печатается фамилия и инициалы пассажира.

## 5. Переменные и константы

### 5.1. Переменные структуры в pass.h

name	Фамилия
------	---------

<b>initials</b>	Инициалы
<b>num_cloth</b>	Количество вещей
<b>weight_cloth</b>	Вес вещей
<b>point</b>	Точка назначения
<b>time</b>	Время вылета
<b>trans</b>	Наличие пересадок
<b>childs</b>	Наличие детей

## 5.2. Переменные в файле dump.c

<b>argv</b>	Аргументы, передаваемые в <b>main</b> из терминала: название исходного файла и название нового бинарного файла
<b>argc</b>	Количество передаваемых аргументов в <b>main</b>

## 5.3 Переменные в файле prog.c

<b>argv</b>	Аргументы, передаваемые в <b>main</b> из терминала: название входного файла и различные ключи
<b>argc</b>	Количество передаваемых аргументов в <b>main</b>
<b>count</b>	количество вещей, на которое должно превосходить количество вещей пассажира от среднего.

# 6. Входные данные

## 6.1. dump.c

Первый аргумент - текстовый файл для чтения с данными о пассажирах.

Второй аргумент - бинарный файл для записи.

## 6.2. prog.c

Первый аргумент - бинарный файл с данными о пассажирах.

Второй аргумент - ключ -f или -p.

Третий аргумент - число p

# 7. Выходные данные

## 7.1. dump.c

Бинарный файл.

## 7.2. prog.c

**-f:** таблица со всеми пассажирами и данными о них

**-p:** пассажиры, число вещей которых превосходит среднее число вещей не менее, чем на *p* штук.

# 8. Программа на СИ

## 8.1. pass.h

```
typedef struct
{
    char name[50];
    char initials[3];
    int num_cloth;
    float weight_cloth;
    char point[23];
    char time[10];
    char trans[10];
    char childs[10];
} pass;
```

## 8.2. dump.c

```
#include <stdio.h>
#include "pass.h"
void war(int argc)
{
    if (argc!=3)
        printf("Предупреждение: некорректный ввод, могут возникнуть проблемы в выполнении работы программы\n");
    else
        printf("Конвертация прошла успешно.\n");
}
```

```

void binary(char* *argv)
{
    pass p;
    FILE *input;
    FILE *output;
    output = fopen(argv[2], "wb");
    input = fopen(argv[1], "r");
    while(fscanf(input,"%s %s %d %f %s %s %s %s", p.name,
p.initials, &p.num_cloth, &p.weight_cloth, p.point,
p.time, p.trans, p.childs)!=EOF)
        fwrite(&p,sizeof(p),1,output);
    fclose(input);
    fclose(output);
}

int main(int argc, char* argv[])
{
    war(argc);
    binary(argv);
}

```

### 8.3. prog.c

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "pass.h"

int table(int argc, char* argv[]) //Проверяет наличие
аргумента "-f", который отвечает за отображение таблицы.
{
    for (int n = 1; n<argc; n++)
    {
        if (strcmp(argv[n], "-f")==0)
            return 1;
    }
}

```



```

        return 0;
    }
    int work(int argc, char* argv[]) // Проверяет наличие
    аргумента "-p", который отвечает за выполнение задания.
    {
        for (int n = 1; n<argc; n++)
        {
            if (strcmp(argv[n], "-p")==0)
                return 1;
        }
        return 0;
    }
    int medium(char* argv) // находит среднее арифметическое
    общего кол-ва вещей.
    {
        pass p;
        FILE *input;
        input=fopen(argv, "rb");
        int medium = 0;
        int n=0;
        while (fread(&p, sizeof(p), 1, input) == 1)
        {
            medium+=p.num_cloth;
            n++;
        }
        fclose(input);
        return medium=medium/n;
    }
    void printtable(int argc, char* *argv) //печатает таблицу,
    если найден аргумент "-f"
    {
        if (table(argc, argv)==1)

```

```

{
    pass p;
    FILE *input;
    input=fopen(argv[1], "rb");
    int k=0;
    while (k<76)
    {
        k++;
        printf("-");
    }
    printf("\n| \tИмя\t |Вещи| Вес |\tточка\t |время|
Пересадки | Дети |\n");
    int l = 1;
    while (l<77)
    {
        l++;
        printf("-");
    }
    while (fread(&p, sizeof(p), 1, input) == 1)
    {
        printf("\n|%12s   %3s|%3d   |%3.3f\t|   %14s
|%5s|\t%3s|\t%3s|\n", p.name, p.initials,
p.num_cloth, p.weight_cloth, p.point, p.time,
p.trans, p.chlds);
        int s = 1;
        while (s<77)
        {
            s++;
            printf("-");
        }
    }
    printf("\n");
    fclose(input);
}

```

```

    }
}

void printwork(int argc, char* *argv) //Выполняет задание,
если найден аргумент -p и корректно задан параметр p.
{
    if (work(argc, argv)==1)
    {
        printf("\n");
        pass p;
        FILE *input;
        int count;
        input=fopen(argv[1], "rb");
        count=atoi(argv[3]);
        while (fread(&p, sizeof(p), 1, input) == 1)
        {
            if (p.num_cloth-count>=medium(argv[1]))
                printf("%s %s\n", p.name, p.initials);
        }
        fclose(input);
    }
}

void main(int argc, char* argv[])
{
    printtable(argc, argv);
    printwork(argc, argv);
}

```

## **9. Демонстрация работы программы**

### **9.1. Файлы с данными о пассажирах**

**file1.txt:**

Syroezhkin KG 32 30 Shardja 23:00 no no

Habirov AR 2 2.1 Tokyo 22:00 yes yes

Lolos SL 4 2.5 Kirov 21:22 no yes  
Kirov SK 14 2.1 Washington 19:20 no yes  
Ilad VD 8 2.9 London 7:00 no no  
Ursa KD 17 22 Irkuts 9:00 no no  
Old TU 18 22 Irjan 6:00 no yes  
Vafel OK 11 12 Tushino 5:22 no no  
Iardan US 5 90 New-York 2:22 yes no  
Puchkov DM 1 99 Tokyo 9:00 no no  
Gaydamak KV 2 2 Hong-Kong 15:20 no no  
Rijkin DT 13 3.99 Ierusalim 13:13 yes yes  
Kadichkina LL 6 6.32 Omsk 19:22 yes yes  
Getzel VD 1 0.01 Taimir 23:59 yes yes  
Ararat JH 8 8.01 Akava 13:17 no yes  
Agapovkin IB 3 9.2 Ayato 12:32 no no  
Lolos FK 15 4.2 Akito 23:21 no yes

**file2.txt:**

Akarin KJ 40 33 Akihabara 0:00 yes no  
Uyka US 22 80 Undergraund 8:59 yes no  
Ufan LL 41 32 Ahaba 23:23 yes yes  
Prewf DD 29 43 Ikarus 12:43 no no  
Lands SO 72 12 Iunda 15:23 yes no  
Ikwer FO 63 34 Oukar 0:50 no yes  
Korn SD 21 53 Moscow 1:32 yes no  
Jacobs IO 64 21 SPB 3:32 no yes  
Nescafe UY 21 2.2 Omae 5:32 yes no  
Coffe YT 44 32 Wa 6:32 no yes  
Undsos VK 1 31 Mou 7:23 no yes  
Gang DA 32 42 Shindeiru 8:31 yes no

Bang AD 12 43.1 Na 5:59 yes no

Taski MB 31 53.3 Ni 5:32 yes no

Ucvol AM 5 33.4 AJIN 4:32 yes no

Yigo UE 31 31.4 Poper 3:53 no no

## 9.2. Тестирование

```
akrills@DESKTOP-70J5NO3:~/Dev/LabsMAI/SecondSem/Kurs6$ gcc -o  
dump.exe dump.c
```

```
akrills@DESKTOP-70J5NO3:~/Dev/LabsMAI/SecondSem/Kurs6$ gcc -o  
prog.exe prog.c
```

```
akrills@DESKTOP-70J5NO3:~/Dev/LabsMAI/SecondSem/Kurs6$ ./dump.exe  
file1.txt file1.bin
```

Конвертация прошла успешно.

```
akrills@DESKTOP-70J5NO3:~/Dev/LabsMAI/SecondSem/Kurs6$ ./prog.exe  
file1.bin -f
```

Имя	Вещи	Вес	точка	время	Пересадки	Дети
Syroezhkin	KG  32  30.000		Shardja	23:00	no	no
Habirov	AR  2  2.100		Tokyo	22:00	yes	yes
LoLos	SL  4  2.500		Kirov	21:22	no	yes
Kirov	SK  14  2.100		Washington	19:20	no	yes
Ilad	VD  8  2.900		London	7:00	no	no
Ursa	KD  17  22.000		Irkuts	9:00	no	no
Old	TU  18  22.000		Irjan	6:00	no	yes
Vafel	OK  11  12.000		Tushino	5:22	no	no
Iardan	US  5  90.000		New-York	2:22	yes	no
Puchkov	DM  1  99.000		Tokyo	9:00	no	no
Gaydamak	KV  2  2.000		Hong-Kong	15:20	no	no
Rijkin	DT  13  3.990		Ierusalim	13:13	yes	yes
Kadichkina	LL  6  6.320		Omsk	19:22	yes	yes
Getzel	VD  1  0.010		Taimir	23:59	yes	yes
Anarat	JH  8  8.010		Akava	13:17	no	yes
Agapovkin	IB  3  9.200		Ayato	12:32	no	no
LoLos	FK  15  4.200		Akito	23:21	no	yes

```
Akrills@DESKTOP-70J5NO3:~/Dev/LabsMAI/SecondSem/Kurs6$ ./prog.exe  
file1.bin -p 9
```

Syroezhkin KG

Old TU

```
akrills@DESKTOP-70J5NO3:~/Dev/LabsMAI/SecondSem/Kurs6$ ./prog.exe  
file1.bin -p 5
```

Syroezhkin KG

Kirov SK

Ursa KD

Old TU

Lolos FK

```
akrills@DESKTOP-70J5NO3:~/Dev/LabsMAI/SecondSem/Kurs6$ ./prog.exe  
file1.bin -p 0
```

Syroezhkin KG

Kirov SK

Ursa KD

Old TU

Vafel OK

Rijkin DT

```
akrills@DESKTOP-70J5NO3:~/Dev/LabsMAI/SecondSem/Kurs6$ ./prog.exe  
file1.bin -p 23
```

Syroezhkin KG

```
akrills@DESKTOP-70J5NO3:~/Dev/LabsMAI/SecondSem/Kurs6$ ./dump.exe  
file2.txt file2.bin
```

Конвертация прошла успешно.

```
akrills@DESKTOP-70J5NO3:~/Dev/LabsMAI/SecondSem/Kurs6$ ./prog.exe  
file2.bin -f
```

Имя	Вещи	Вес	точка	время	Пересадки	Дети	
Akarin	KJ	40	33.000	Akihabara	0:00	yes	no
Uyka	US	22	80.000	Undergraund	8:59	yes	no
Ufan	LL	41	32.000	Ahaba	23:23	yes	yes
Prewf	DD	29	43.000	Ikarus	12:43	no	no
Lands	SO	72	12.000	Iunda	15:23	yes	no
Ikwer	FO	63	34.000	Oukar	0:50	no	yes
Korn	SD	21	53.000	Moscow	1:32	yes	no
Jacobs	IO	64	21.000	SPB	3:32	no	yes
Nescafe	UY	21	2.200	Omae	5:32	yes	no
Coffe	YT	44	32.000	Wa	6:32	no	yes
Undsos	VK	1	31.000	Mou	7:23	no	yes
Gang	DA	32	42.000	Shindeiru	8:31	yes	no
Bang	AD	12	43.100	Na	5:59	yes	no
Taski	MB	31	53.300	Ni	5:32	yes	no
Ucvol	AM	5	33.400	AJIN	4:32	yes	no
Yigo	UE	31	31.400	Poper	3:53	no	no

```
akrills@DESKTOP-70J5NO3:~/Dev/LabsMAI/SecondSem/Kurs6$ ./prog.exe
file2.bin -p 33
```

Lands SO

```
akrills@DESKTOP-70J5NO3:~/Dev/LabsMAI/SecondSem/Kurs6$ ./prog.exe
file2.bin -p 0
```

Akarin KJ

Ufan LL

Lands SO

Ikwer FO

Jacobs IO

Coffe YT

```
akrills@DESKTOP-70J5NO3:~/Dev/LabsMAI/SecondSem/Kurs6$ ./prog.exe
file2.bin -p 9
```

Lands SO

Ikwer FO

Jacobs IO

Coffe YT

## **Заключение**

Данное задание курсовой работы позволяет:

1. Научиться работать со структурами, задавать их в библиотеках;
2. Работать с файлами (запись и чтение бинарных файлов );
3. Создавать базы данных.

Эти навыки внесли большой вклад в мой опыт прикладного программирования.

## **Литература**

Методические указания к выполнению курсовых работ. Зайцев В. Е.

[https://learnc.info/c/text\\_files.html](https://learnc.info/c/text_files.html)

<http://www.sbp-program.ru/c/sbp-file-c.htm>

[http://files.mai.ru/site/priem/documents/orders/2017/111\\_03.08.2017.pdf](http://files.mai.ru/site/priem/documents/orders/2017/111_03.08.2017.pdf)