**Отчет по лабораторной работе №VI** по курсу практикум на ЭВМ

Студент группы М8О-101Б-20 Ядров Артем Леонидович, № по списку 28

Контакты www, e-mail, icq, skype temayadrow@gmail.com

Работа выполнена: « » \_\_\_\_\_\_\_\_\_\_\_\_202\_\_\_г.

Преподаватель: доцент каф. 806 Никулин Сергей Петрович

Входной контроль знаний с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Отчет сдан « » \_\_\_\_\_\_\_\_\_202 \_\_ г., итоговая оценка \_\_\_\_\_

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. **Тема:** Обработка последовательной файловой структуры на Си

1. **Цель работы:** Научиться обрабатывать последовательные файловые структуры на Си

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Задание** (*вариант №* 3*8* )**:**  Информация о пассажирах аэропорта: фамилия, инициалы, количество вещей, общий вес вещей, пункт назначения, время вылета, наличие пересадок, информация о детях

Дать сведения о пассажирах, число вещей которых не меньше, чем в любом другом багаже, а общий вес вещей не больше, чем в любом другом багаже с этим же числом вещей

1. **Оборудование** (лабораторное):

ЭВМ Intel Pentium G2140, процессор 3.30 GHz , имя узла сети Cameron с ОП 8096 Мб, НМД 7906 Мб. Терминал ASUS адрес dev/pets/3 Принтер HP Laserjet 6P

Другие устройства \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Оборудование ПЭВМ студента, если использовалось:*

Процессор \_\_Intel core i5-7300HQ 2.50 GHz с ОП 8096 Мб, НМД 131072 Мб. Монитор ASUS

Другие устройства \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Программное обеспечение (лабораторное):**

Операционная система семейства Unix , наименование Ubuntu версия 18.15.0

интерпретатор команд bash версия 4.4.20

Система программирования GNU версия 5.8.13

Редактор текстов emacs версия 25.2.2

Утилиты операционной системы cat

Прикладные системы и программы

Местонахождение и имена файлов программ и данных stud/208104

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Программное обеспечение ЭВМ студента, если использовалось:*

Операционная система семейства Unix , наименование Fedora версия 33

интерпретатор команд bash версия 5.0.17

Система программирования Clion версия 2020.3

Редактор текстов emacs версия 25.2.2 Утилиты операционной системы cat, gcc

Прикладные системы и программы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Местонахождение и имена файлов программ и данных на домашнем компьютере home/Temi4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**6. Идея, метод, алгоритм** решения задачи(в формах:словесной,псевдокода,графической[блок-схема,диаграмма,рисунок,таблица] или формальные спецификации с пред- и постусловиями)

Для представления базы данных создадим в заголовочном файле структуру:

* typedef struct {

char surname[25];

char initials[2];

char s;

int num\_class;

char ch\_class;

char univ[10];

char work[10];

int rgt;

} graduater;

Структура «graduater» будет представлять нашу базу данных. В ней содержатся поля, необходимые для инициализации выпускника школы (фамилия, инициалы, пол, номер класса, буква класса, название университета, место работы, полк в котором служил (или 0 в противном случае))

Опишем **первую программу** (graduater\_dump.c ), которая считывает базу данных с текстового файла и заносит данные в бинарный файл. С помощью функции стандартной библиотечной функции определим количество параметров, поданных в терминале во время запуска программы. Если argc возвращает значение, не равное 3 (программа, входной файл, выходной файл), то вызовем функцию void Usage(), выводящую подсказку по использованию программы. Если значение функции argc равно 3, то зададим входной (in) и выходной файл (out) с помощью функций fopen и argv(В argv[1] должно лежать имя входного файла, argv[2] - выходного). Если хотя бы один файл не удалось открыть, то завершаем программу. Далее запускаем цикл while, в котором условие задается функцией считывания данных выпускника (int readpassanger), которая выводит 1, если удалось считать данные выпускника, и 0 - в противном случае. Записываем с помощью функции fwrite данные выпускника в бинарный файл.

Опишем **вторую программу** (cool\_graduaters.c), которая считывает базу данных с бинарного файла и определяет, есть ли однофамильцы в параллельных классах. Для начала с помощью функции argc определим количество заданных параметров. Если количество не равно 3 (имя программы, ключ, входной файл), то выведем при помощи функции void Usage() подсказку по использованию программы. Далее проверим, какой ключ задан. Ключ содержится в argv[1], а имя файла - в argv[2]. Будем хранить 2 переменные (f, p), принимающие 1, если задан соответствующий ключ, и 0 в противном случае. Будем использовать функцию strcmp для сравнения строк (возвращает 0, если строки одинаковые, и 1, если разные). Если файл невозможно открыть, то выведем соответствующую ошибку. Если задан ключ «-f», то выведем шапку таблицы. Далее пройдемся в цикле по бинарному файлу, считывая данные выпускников. Если задан ключ «-f», то просто выведем очередную строку таблицы. После завершения цикла завершаем программу, если задан ключ «-f». Если задан ключ «-p», то проходим по файлу, считая количество выпускников, затем в цикле для каждого выпускника будем «подбирать пару» - сравнивать каждого следующего с текущим. Если у них одинаковая фамилия, класс и разные буквы класса, то выведем об этом сообщение и завершим программу.

Оценим **временную сложность** алгоритма: В худшем случае мы совершаем O(n\*(n-1)) итерацию цикла, то есть наша сложность равна O(n2).

Теперь оценим **пространственную сложность** алгоритма: Мы храним лишь ссылку на файл, переменные, которые не зависят от размера файла. Следовательно, пространственная сложность алгоритма равна O(1).

1. **Сценарий выполнения работы** [план работы,первоначальный текст программы в черновике(можно на отдельном листе)итесты либо соображения по тестированию].

* Ivanov II m 11 a MAI - 0

Petrov MA m 9 b - Yandex 123

Ivanov AA m 11 b MAI - 1124

Ivanov AK m 11 c MGU - 0

Pankratova KI f 11 a MAI - 0

Pankratova MA f 11 c MIREA - 0

* Ivanov II m 11 a MAI - 0

Petrov MA m 9 b - Yandex 123

Pankratova MA f 11 c MIREA - 0

*Пункты 1-7 отчета составляются строго до начала лабораторной работы.*

*Допущен к выполнению работы.* **Подпись преподавателя****\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. **Распечатка протокола** (подклеить листинг окончательного варианта программы с тестовыми примерами,подписанныйпреподавателем).

**[Temi4@localhost labs]$ cat input**

**Ivanov II m 11 a MAI - 0**

**Petrov MA m 9 b - Yandex 123**

**Ivanov AA m 11 b MAI - 1124**

**Ivanov AK m 11 c MGU - 0**

**Pankratova KI f 11 a MAI - 0**

**Pankratova MA f 11 c MIREA - 0**

**[Temi4@localhost labs]$ cat graduater\_dump.c**

**#include <stdio.h>**

**#include <string.h>**

**#include <errno.h>**

**#include "graduater.h"**

**void usage() {**

**printf("Usage: program input\_filename output\_filename\n");**

**}**

**int readgraduater(FILE \*in, graduater \*g) {**

**return fscanf(in, "%[^\t]\t%c%c\t%c\t%d\t%c\t%[^\t]\t%[^\t]\t%d\n", g->surname, &g->initials[0],&g->initials[1], &g->s,**

**&g->num\_class, &g->ch\_class, g->univ, g->work, &g->rgt) == 9;**

**}**

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

**if (argc != 3) {**

**usage();**

**return 1;**

**}**

**graduater g;**

**FILE \*out = fopen(argv[2], "w");**

**FILE \*in = fopen(argv[1], "r");**

**if (!(out && in)) {**

**perror("Can't open file");**

**return 2;**

**}**

**while (readgraduater(in, &g)) {**

**fwrite(&g, sizeof(g), 1, out);**

**}**

**return 0;**

**}[Temi4@localhost labs]$ gcc graduater\_dump.c**

**[Temi4@localhost labs]$ ./a.out**

**Usage: program input\_filename output\_filename**

**[Temi4@localhost labs]$ ./a.out input output**

**[Temi4@localhost labs]$ cat cool\_graduaters.c**

**#include <stdio.h>**

**#include <string.h>**

**#include <stdbool.h>**

**#include "graduater.h"**

**bool equal\_surnames(graduater g, graduater h) {**

**bool c = true;**

**for (int i = 0; i < 25; i++) {**

**c = c && (g.surname[i] == h.surname[i]);**

**}**

**return c;**

**}**

**void usage() {**

**printf("Usage: program [-key] filename\nKeys: -f xor -p\n");**

**}**

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

**FILE \*in;**

**int f = 0, p = 0;**

**if (argc != 3) {**

**usage();**

**return 1;**

**}**

**if (strcmp(argv[1], "-f") == 0) {**

**f = 1;**

**in = fopen(argv[2], "r");**

**} else if (strcmp(argv[1], "-p") == 0) {**

**p = 1;**

**in = fopen(argv[2], "r");**

**} else {**

**usage();**

**return 2;**

**}**

**graduater g, cur;**

**if (!in) {**

**perror("Can not open file\n");**

**return 3;**

**}**

**if (f) {**

**printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");**

**printf("| SURNAME | I | S | N | C |UNIVERSITY| WORK | RGT |\n");**

**printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");**

**while (fread(&g, sizeof(g), 1, in)) {**

**printf("|%-25s|%c%c | %c |%-2d | %c |%-10s|%-10s|%-5d|\n", g.surname, g.initials[0], g.initials[1], g.s,**

**g.num\_class,**

**g.ch\_class, g.univ, g.work, g.rgt);**

**printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");**

**};**

**return 0;**

**}**

**int n = 0;**

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

**while (fread(&g, sizeof(g), 1, in)) {**

**n++;**

**}**

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

**for (int i = 0; i < n; i++) {**

**fseek(in, 0, sizeof(graduater) \* i);**

**fread(&cur, sizeof(graduater), 1, in);**

**while (fread(&g, sizeof(graduater), 1, in)) {**

**if (equal\_surnames(g, cur) && g.num\_class == cur.num\_class && g.ch\_class != cur.ch\_class) {**

**printf("There are namesakes in parallel classes\n");**

**return 0;**

**}**

**}**

**}**

**printf("There aren't namesakes in parallel classes\n");**

**return 0;**

**}[Temi4@localhost labs]$ gcc cool\_graduaters.c**

**[Temi4@localhost labs]$ ./a.out**

**Usage: program [-key] filename**

**Keys: -f xor -p**

**[Temi4@localhost labs]$ ./a.out -f output**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**| SURNAME | I | S | N | C |UNIVERSITY| WORK | RGT |**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**|Ivanov |II | m |11 | a |MAI |- |0 |**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**|Petrov |MA | m |9 | b |- |Yandex |123 |**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**|Ivanov |AA | m |11 | b |MAI |- |1124 |**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**|Ivanov |AK | m |11 | c |MGU |- |0 |**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**|Pankratova |KI | f |11 | a |MAI |- |0 |**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**|Pankratova |MA | f |11 | c |MIREA |- |0 |**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**[Temi4@localhost labs]$ ./a.out -p output**

**There are namesakes in parallel classes**

**[Temi4@localhost labs]$ cat input**

**Ivanov II m 11 a MAI - 0**

**Petrov MA m 9 b - Yandex 123**

**Pankratova MA f 11 c MIREA - 0**

**[Temi4@localhost labs]$ gcc graduater\_dump.c**

**[Temi4@localhost labs]$ ./a.out input output**

**[Temi4@localhost labs]$ gcc cool\_graduaters.c**

**[Temi4@localhost labs]$ ./a.out -p output**

**There aren't namesakes in parallel classes**

1. **Дневник отладки** должен содержать дату и время сеансов отладки, и основные события(ошибки в сценарии и программе,нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Лаб. | Дата | Время | Событие | Действие по исправлению | Примечание |
|  | или |  |  |  |  |  |
|  | дом. |  |  |  |  |  |
|  |  |  |  |  |  |  |

1. **Замечания автора** по существу работы

1. **Выводы**

Я научился обрабатывать файловые структуры на Си.

Недочёты при выполнении задания могут быть устранены следующим образом: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись студента \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_