Отчет по лабораторной работе № 6 по курсу “ Практикум на ЭВМ ”

Студент группы М8О-102Б-21, Кажекин Денис Андреевич, № по списку 9, вариант 41

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

Работа выполнена: « » _____201___г.

Преподаватель: Никулин Сергей Петрович
Каф.806_____

Входной контроль знаний с оценкой _____

Отчет сдан « » _____201 __ г., итоговая оценка _____

Подпись преподавателя _____

1. **Тема:** Обработка последовательной файловой структуры на Си
2. **Цель работы:** Научиться обрабатывать последовательные файловые структуры на Си
3. **Задание (41 вариант):** Выяснить, имеются ли однофамильцы в каких-либо параллельных классах.
4. **Оборудование:**

Оборудование ПЭВМ студента:
Процессор: Apple M1, с ОЗУ 8 гб (виртуальная машина), SSD 256 гб. Монитор: встроенный 13 дюймов IPS 2560×1600, 220 PPI

5. **Программное обеспечение ЭВМ студента, если использовалось:**

Операционная система семейства Linux, наименование Ubuntu, версия 22.04
Прикладные системы и программы: терминал ОС UNIX, текстовый редактор emacs

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

- Изучить содержимое и структуру файла, соответствующие варианту
- Подготовить три текстовых файла input.txt, в которых будут находится входные данные
- Создать файл со структурой student.h
- Создать программу dump.c, которая считает входные данные и создаст новый файл output.txt с бинарными данными
- Создать программу cool.c, которая выведет людей, у которых одинаковые фамилии. (или просто выведет всех людей, если ключ был -f)

7. Сценарий выполнения работы [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты либо соображения по тестированию].

Для представления базы данных создадим в заголовочном файле структуру student:

```
typedef struct {
   char surname[15];
   char initials[2];
   char sex;
   int class_number;
   char class_letter;

} student;
```

=Эта структура и будет представлять нашу базу данных. В ней содержатся поля, необходимые для инициализации студента: фамилия, инициалы, пол, номер класса, буква класса.

Опишем **первую программу** (dump.c), которая считывает базу данных с текстового файла и заносит данные в бинарный файл. С помощью 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 студента (int readstudent), которая выводит 1, если удалось считать данные пассажира, и 0 - в противном случае. Записываем с помощью функции fwrite данные пассажира в бинарный файл.

Опишем **вторую программу** (cool.c), которая считывает базу данных с бинарного файла и выводит данные лицах, имеющих одинаковые фамилии в параллельных классах. Для начала с помощью функции argc определим количество заданных параметров. Если количество не равно 3(имя программы, ключ -f, входной файл), то выведем при помощи функции void usage() подсказку по использованию программы. Далее проверим, какой ключ задан. Ключ содержится в argv[1]. Если был задан ключ -f, то переменная f станет раной 1. Если был задан ключ -p, то переменная p станет раной 1. Будем использовать функцию strcmp для сравнения строк (возвращает 0, если строки одинаковые, или другое число, если разные). Если файл невозможно открыть, то выведем соответствующую ошибку. Если задан ключ -f, то выведем шапку таблицы и всех людей. После завершения цикла завершаем программу, если задан ключ -f. Если задан ключ -p, то с помощью двух вложенных циклов “while” и условия на то, чтобы соблюдалось равенство номера класса и фамилий двух лиц, определяем однофамильцев в параллели. Так как мы используем два цикла “while”, то количество повторений фамилий в переменной “count” всегда будет равно 1 (каждый человек считается сам с собой), следовательно, надо поставить дополнительное условие на то, что если переменная “count” будет равна 2, то мы нашли однофамильца и информацию о нем необходимо вывести.

Входные файлы:

Input1.txt:

Kazhekin	DA	M	11	A
Boguzh	VA	M	11	B
Yatsenko	AV	M	11	A
Vovin	AG	M	10	M
Golovin	GO	M	10	B
Sashin	MO	M	8	M
Sashin	MO	M	10	Z
Sashin	AK	M	8	A

Input2.txt:

Martishenko	DA	M	11	A
Makarov	VA	M	11	B
Momin	AV	M	11	A
Risov	AG	M	10	A
Uvarov	GO	M	10	B
Bobin	MO	M	8	A
Dimin	MO	M	10	B

Robin	AK	M	8	C

Input3.txt:

Vlasov	DA	M	11	A
Korneev	VA	M	7	D
Korneev	AV	M	7	A
Vlasov	AG	M	11	V
Klebov	GO	M	10	M
Galkin	MO	M	8	K
Markizov	MO	M	10	A
Galkin	AK	M	8	C

Input4.txt:

Markov	DA	M	7	D	
Markov	VA	M	7	D	
Mishin	AV	M	8	A	
Magomedov		AG	M	11	M
Magomedov		GO	M	11	M
Mishin	MO	M	8	K	
Viselkov	MO	M	10	A	
Viselkov	AK	M	10	A	

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

Допущен к выполнению работы. **Подпись преподавателя**

_____

8. Распечатка протокола (подклеить листинг окончательного варианта программы с тестовыми примерами, подписанный преподавателем)

```
denis@ubuntu:~/Downloads$ cat student.h
#ifndef LABS_STUDENTS_H
#define LABS_STUDENTS_H

typedef struct {
   char surname[15];
   char initials[2];
   char sex;
   int class_number;
   char class_letter;

} student;

#endif //LABS_STUDENTS_H
denis@ubuntu:~/Downloads$ cat dump.c
#include <stdio.h>
#include <string.h>
#include <errno.h>

#include "student.h"

void usage() {
   printf("Usage: program input_filename output_filename\n");
}

int readstudent(FILE *in, student *p) {
   return fscanf(in, "%[^D\t]\t%[^\t]\t%c\t%d\t%c\t", p->surname, p->initials, p->sex, &p->class_number,
               &p->class_letter) == 5;
}

int main(int argc, char *argv[]) {
   if (argc != 3) {
      usage();
      return 1;
   }
   student p;
   FILE *out = fopen(argv[2], "w");
   FILE *in = fopen(argv[1], "r");
   if (!(out && in)) {
      perror("Can't open file");
      return 2;
   }
   while (readstudent(in, &p)) {
      fwrite(&p, sizeof(p), 1, out);
   }
   return 0;
}
denis@ubuntu:~/Downloads$ cat cool.c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <limits.h>
```

```
#include "student.h"

void usage() {
   printf("Usage: program [-key] filename\nKeys: -f xor -p\n");
}

int main(int argc, char *argv[]) {
   FILE *in;
   int f = 0;
   int 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;
   }
   student stud;
   if (!in) {
      perror("Can not open file\n");
      return 3;
   }
   if (f) {

printf("_____
_____\n");
      printf("|     Surname      | I |   Sex   | Class | Letter |\n");

printf("_____
_____\n");
      while (fread(&stud, sizeof(stud), 1, in) == 1) {
            printf("|%-15s|%-2s|%-1c|%-1d|%-1c|\n", stud.surname, stud.initials,
               stud.sex, stud.class_number,
               stud.class_letter);

printf("_____
_____\n");
      }
      return 0;
   }
   fseek(in, 0, SEEK_SET);
   printf("\t\t\t\t\tANSWER\n");
```

```
printf("_____
_____\n");
        printf("|     Surname      | I |   Sex   | Class | Letter |\n");

printf("_____
_____\n");
    while (fread(&stud, sizeof(stud), 1, in) == 1) {
        // char sur1[15] = stud.surname;
        char sur1[20];
        for(int i = 0; i < sizeof(stud.surname); ++i){
            sur1[i] = stud.surname[i];
        }
        // char init1[2] = stud.initials;
        char init1[20];
        for(int i = 0; i < sizeof(stud.initials); ++i){
            init1[i] = stud.initials[i];
        char sex1 = stud.sex;
        int num1 = stud.class_number;
        char letter1 = stud.class_letter;
        int count = 0;
        while (fread(&stud, sizeof(stud), 1, in) == 1) {
            char sur2[20];
            int num2 = stud.class_number;
            if (strcmp(sur1, stud.surname) == 0 && num1 == num2) {
                count++;
                if (count == 2) break;
            }
        }
        if (count >= 2) {
            printf("|%-15s|%-2s|%-1c|%-1d|%-1c|\n", sur1, init1,
                  sex1, num1, letter1);

printf("_____
_____\n");
        }
    }
    return 0;
}
}
denis@ubuntu:~/Downloads$ gcc dump.c
denis@ubuntu:~/Documents$ ./a.out
Usage: program input_filename output_filename
denis@ubuntu:~/Documents$ ./a.out input1.txt output1.txt.
denis@ubuntu:~/Documents$ ./a.out input2.txt output2.txt
denis@ubuntu:~/Documents$ ./a.out input3.txt output3.txt
denis@ubuntu:~/Documents$ ./a.out input4.txt output4.txt
denis@ubuntu:~/Downloads$ gcc cool.c
denis@ubuntu:~/Downloads$ ./a.out
```

Usage: program [-key] filename
Keys: -f xor -p
denis@ubuntu:~/Documents$ ./a.out -f output1.txt

Surname	I	Sex	Class_number	Class_letter
Kazhekin	DA	M	11	A
Boguzh	VA	M	11	B
Yatsenko	AV	M	11	A
Vovin	AG	M	10	M
Golovin	GO	M	10	B
Sashin	MO	M	8	M
Sashin	MO	M	10	Z
Sashin	AK	M	8	A

denis@ubuntu:~/Documents$ ./a.out -p output1.txt

ANSWER

Surname	I	Sex	Class_number	Class_letter
Sashin	MO	M	8	M
Sashin	AK	M	8	A

denis@ubuntu:~/Documents$ ./a.out -p output2.txt

Surname	I	Sex	Class_number	Class_letter

denis@ubuntu:~/Documents$ ./a.out -p output3.txt

Surname	I	Sex	Class_number	Class_letter
Korneev	VA	M	7	D
Korneev	AV	M	7	A
Galkin	MO	M	8	K
Galkin	AK	M	8	C

Vlasov	DA	M	11	A

Vlasov	AG	M	11	V

denis@ubuntu:~/Documents$ ./a.out -p output4.txt

Surname	I	Sex	Class_number	Class_letter
Mishin	AV	M	8	A
Mishin	MO	M	8	K

9. 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. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание

10. Замечания автора по существу работы: -

11. Выводы:

Я научился обрабатывать файловые структуры на Си.

Подпись студента

_____