

Министерство науки и высшего образования РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский Авиационный Институт»
Национальный Исследовательский Университет

Факультет №8 «Информационные технологии и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»

КУРСОВОЙ ПРОЕКТ

По дисциплине «Языки и методы программирования»

По курсу: «Инструментальные средства UNIX, алгоритмы и структуры данных»

Задание: «Обработка последовательностей файловой структуры на языке Си»

Студент:	Хренникова А. С.
Группа:	М80-108-19
Преподаватель:	Поповкин А. В.
Подпись:	
Оценка:	
Дата:	

Содержание

Задание	3
Теоретическая часть.....	4
Общие сведения о программе	5
Описание логической структуры.....	6
Описание переменных, функций, входные и выходные данные	7
Протокол	9
Заключение	15
Список использованных источников	16

Задание

Разработать последовательность структур 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. Распечатать содержимое сгенерированного файла в виде таблицы и выполнить над ним заданное действие для 2-3 значений параметров запроса *p* и распечатать результат.

Действие по выборке данных из файла оформить в виде отдельной программы с параметрами запроса, вводимыми из стандартного входного текстового файла, или получаемыми из командной строки UNIX. Второй способ задания обязателен для работ, оцениваемых на хорошо и отлично. Параметры задаются с помощью ключей -f (распечатка файла) или -p <parameter> (параметры конкретного варианта задания). Получение параметров из командной строки производится с помощью стандартных библиотечных функций *argc* и *argv*.

Структуры данных и константы, совместно используемые программами, следует вынести в отдельный заголовочный файл.

В процессе отладки и тестирования рекомендуется использовать команды обработки текстовых файлов ОС UNIX и переадресацию ввода-вывода. Сгенерированные и отформатированные текстовые данные необходимо заранее поместить в текстовые файлы и распечатывать при протоколировании. Рекомендуется подобрать реальные или правдоподобные тестовые данные. Число наборов тестовых данных должно быть не менее трех: Имя файла с бинарными данными является обязательным параметром второй программы. Отчет должен содержать оценку пространственной и временной сложности использованного алгоритма. В состав отчета также рекомендуется включить графическую иллюстрацию структуры файла и запроса на выборку.

Вариант: определить абитуриентов, получивших неудовлетворительную оценку.

Теоретическая часть

Если файл открыт в бинарном режиме, его можно записывать или считывать побайтно. Функция *fseek()* позволяет обращаться с файлом как с массивом и переходить к любой позиции в файле, обеспечивая возможность произвольного доступа. Если текстовые файлы являются файлами с последовательным доступом, то произвольный доступ чаще всего применяется к бинарным файлам.

Бинарные файлы могут содержать любую информацию. Чаще всего используются файлы, содержащие структуры. Для чтения и записи в бинарные файлы можно использовать функции *fread()*, *fwrite()* или *fscanf()*, *fprintf()*.

fread – функция для чтения из файла:

```
int fread(void *ptr, unsigned size, unsigned count, FILE *f);
```

Из файла *f* считывается и по адресу *ptr* записываются *count* элементов размером *size* каждый. Функция возвращает число фактически считанных элементов.

fwrite – функция для записи в файл:

```
int fwrite(void *ptr, unsigned size, unsigned count, FILE *f);
```

В файл записываются, начиная с адреса *ptr*, элементов размером *size* каждый. Функция возвращает число фактически записанных элементов.

Когда мы определяем новую структуру с помощью служебного слова *struct*, в пространстве имен структур создается новый идентификатор. Для доступа к нему необходимо использовать служебное слово *struct*. Можно определить новый тип с помощью служебного слова *typedef*. Тогда будет создан новый псевдоним для нашей структуры, видимый в глобальном контексте.

Общие сведения о программе

Необходимое программное и аппаратное обеспечение: ОС семейства UNIX (Linux Ubuntu), среда программирования Си (язык Си, компилятор gcc), процессор с 64-битной архитектурой (как на лабораторном компьютере).

Система программирования: GUN C.

Местонахождение файлов на домашнем компьютере: /home/lina_tucha/dir/крб. Сам файл компилируется с помощью написания «gcc -Wall persons.c -o persons» в командной строке интерпретатора команд. Запуск файла осуществляется вызовом исполняемого файла «./persons»(-o задает имя исполняемого файла)/

Описание логической структуры

- Создается база данных, в которую пользователь вводит необходимые данные(фамилию, инициалы, пол, школу, оценку абитуриента).
- Программа считывает все данные и выводит фамилии абитуриентов, данные которых удовлетворяют условию задачи.

Описание переменных, функций, входные и выходные данные

Переменные и константы вспомогательных функций

Таблица 1 - Переменные и константы функции main(основы программы)

Имя	Тип	Значение(начальное)	Назначение
nameFile	char	Отсутствует	Имя файла
ch	int	0	Оператор цикла
c	int	-1	Аргумент оператора выбора
surname	double	Отсутствует	Фамилия абитуриента
number	int	Отсутствует	Порядковый номер абитуриента
gender	char	Отсутствует	Пол абитуриента
inc	char	Отсутствует	Инициалы абитуриента
school	char	Отсутствует	Школа абитуриента
result	int	Отсутствует	Оценка абитуриента

Вспомогательные функции

Функция fam_search:

Сравнивает необходимые параметры и выводит фамилии необходимых абитуриентов.

Встроенные функции

Функция strcat:

`char * strcat(char * destptr, const char * srcptr);`

Объединение строк. Функция добавляет копию строки `srcptr` в конец строки `destptr`. Нулевой символ конца строки `destptr` заменяется первым символом строки `srcptr`, и новый нуль-символ добавляется в конец уже новой строки, сформированной объединением символов двух строк в строке `destptr`.

Функция gets:

```
char * gets( char * string );
```

Получить строку из стандартного потока ввода. Функция gets считывает символы из стандартного потока ввода до символа новой строки `n` или до тех пор, пока не будет достигнут конец файла EOF, после чего сохраняет считанные символы в строку типа `char`.

Символ новой строки `n` не копируется в строку.

Нулевой символ автоматически добавляется после последнего копируемого символа в `string`, чтобы сигнализировать о конце строки.

Функция puts:

```
int puts( const char * string );
```

Функция puts выводит строку типа `char*`, на которую указывает параметр `string` в стандартный поток вывод и добавляет символ новой строки `'n'`. Функция начинает копировать строку с адреса, указанного в `string`, пока не достигнет нулевого символа `"`. Этот заключительный, нулевой символ не копируется в стандартный поток вывод.

Функция rewind:

```
void rewind ( FILE * filestream );
```

Устанавливает внутренний указатель положения файла в начальное положение (начало файла).

Входные данные:

Входные данные – параметры, вводимые пользователем через консоль(фамилия, инициалы, пол, школа, оценка), для каждого абитуриента.

Выходные данные:

Фамилии и оценки абитуриентов, баллы которых меньше(или равны) 2.

Протокол

```
lina_tucha@LAPTOP-44CRFC1U:~/dir/kp6$ cat main.c
```

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include "stud.h"
```

```
int main(int argc, char* argv[])
```

```
{
```

```
    int c;
```

```
    char nameFile[20];
```

```
    FILE *tf;
```

```
    while (c!=0) {
```

```
        //очищаем консоль
```

```
        printf("\n-----\n");
```

```
        printf("Выберите пункт меню \n");
```

```
        printf("1 – Создать новый файл\n");
```

```
        printf("2 – Открыть уже существующий файл\n");
```

```
        printf("3 – Просмотр файла\n");
```

```
        printf("4 – Добавление данных в файл\n");
```

```
        printf("0 – Выход из программы\n");
```

```
        scanf("%d", &c);
```

```
        switch(c) {
```

```
            case 1:
```

```
                //Очищаем консоль
```

```
                printf("\033c");
```

```
                printf("Вы выбрали первый пункт меню\n");
```

```
                printf("Дайте название файлу: ");
```

```
                scanf("%s", nameFile);
```

```
                strcat(nameFile, ".dat");
```

```
                tf=fopen(nameFile, "wb");
```

```
                printf(" %s\n", nameFile);
```

```
                fclose(tf);
```

```

        break;
case 2:
    printf("\033c");
    printf("Введите имя базы данных: ");
    scanf("%s", nameFile);
    strcat(nameFile, ".dat");
    break;
case 3:
    tf = fopen(nameFile, "rb");
    fread(&stud, sizeof(stud), 1, tf);
    while(!feof(tf)) {
        printf("\n%i %s\t%s\t%s\t%s\t%i", stud.number, stud.surname, stud.inc, stud.gender,
stud.school, stud.result);
        fread(&stud, sizeof(stud), 1, tf);
    }
    fclose(tf);
    break;
case 4:
    tf=fopen(nameFile, "ab");
    int ch=0;
    printf("Вы выбрали третий пункт меню\n");
    printf("Ввод сведений в базу\n");
    while (ch!=1) {
        puts(" ");
        gets(stud.tab);
        puts("\nПорядковый номер: ");
        gets(stud.number);
        puts("\nФамилия: ");
        gets(stud.surname);
        puts("\nИнициалы: ");
        gets(stud.inc);
        puts("\nПол: ");
        gets(stud.gender);
        puts("\nШкола ");

```

```

        gets(stud.school);
        puts("\nОценка ");
        gets(stud.result);
        fwrite(&stud, sizeof(stud), 1, tf);
        printf("Закончить ввод? (1 – ‘да’, 0 – ‘нет’)\n");
        scanf("%d", &ch);
    }
    fclose(tf);
    break;
}
}
return 0;
}

```

lina_tucha@LAPTOP-44CRFC1U:~/dir/kp6\$ cat stud.h

```

#ifndef DataBase
#define DataBase
struct Stud{
    int number;
    char surname[20];
    char tab[20];
    char inc[20];
    char gender[20];
    char school[20];
    int result;
} stud;
struct Stud data;
#endif

```

lina_tucha@LAPTOP-44CRFC1U:~/dir/kp6\$ cat persons.c

```

#include <stdio.h>
#include <string.h>

```

```

#include "stud.h"

int fam_search(char nameFile[20], int number, int result) {
    nf = fopen(nameFile, "rb");
    fread(&stud, sizeof(stud), 1, nf);
    if(feof(nf))
        return 0;
    if(stud.number>number && stud.number!=number && stud.result<result) {
        printf("%s\n", stud.surname);
    }
}

int main() {
    char nameFile[20];
    FILE *nf;
    printf("Программа выполняет поиск учеников, получивших неудовлетворительную оценку\n");
    printf("Ведите имя базы данных: \n");
    scanf("%s", nameFile);
    strcat(nameFile, ".dat");
    printf("Список учеников\n");
    int c=-1;
    nf=fopen(nameFile, "rb");
    fread(&stud, sizeof(stud), 1, nf);
    while(!feof(nf)) {
        printf("\n %i %s\t %s\t %s\t %s\t %i", stud.number, stud.surname, stud.inc, stud.gender, stud.school,
stud.result);
        fread(&stud, sizeof(stud), 1, nf);
    }
    printf("\n-----\n");
    rewind(nf);
    while(c!=0) {
        printf("Выберите пункт меню \n");
        printf("1 – выполнить задание \n");
        printf("0 – выход из программы\n");
    }
}

```

```

scanf("%d", &c);
switch(c) {
    case 1:
        while(1) {
            fread(&search, sizeof(stud), 1, nf);
            if(feof(nf)) {
                break;
            }
            fam_search(nameFile, stud.number, stud.surname, stud.result);
        }
    return 0;
}

```

lina_tucha@LAPTOP-44CRFC1U:~/dir/kp6\$ gcc -Wall persons.c -o persons

lina_tucha@LAPTOP-44CRFC1U:~/dir/kp6\$./persons

Программа выполняет поиск учеников, получивших неудовлетворительную оценку

Ведите имя базы данных: kp6

Список учеников

- 1 Иванов С.О. муж. Школа№84 3
- 2 Петров И.В. муж. Лицей№7 5
- 3 Абрамова Л.Г. жен. Школа№4 4
- 4 Самсонов Е.С, муж. Гимназия№114 5
- 5 Григорьева А.М. жен. Гимназия№15 2
- 6 Семенов И.К. муж. Лицей№10 5
- 7 Ковалева У.П. жен. Школа №785 4
- 8 Ткачев К.А. муж. Школа№210 1
- 9 Петренко В.Э. жен. Гимназия№14 4
- 10 Голубев А.А. муж. Лицей№45 2
- 11 Донцов Н.Ю. муж. Школа№7063 2
- 12 Колесова Д.Д. жен. Гимназия№7 5
- 13 Лапенко А.В. муж. Школа№210 2
- 14 Марченко К.И. жен. Школа№4 3
- 15 Новиков В.В. муж. Лицей№30 5

Выберите пункт меню

1 – выполнить задание

0 – выход из программы

1

Григорьева 2

Ткачев 1

Донцов 2

Лапенко 2

Выберите пункт меню

1 – выполнить задание

0 – выход из программы

0

lina_tucha@LAPTOP-44CRFC1U:~/dir/kp6\$ nano kp6.dat

lina_tucha@LAPTOP-44CRFC1U:~/dir/kp6\$ gcc -Wall persons.c -o persons

lina_tucha@LAPTOP-44CRFC1U:~/dir/kp6\$./persons

1 Иванов С.О. муж. Школа№84 3

2 Петров И.В. муж. Лицей№7 5

3 Абрамова Л.Г. жен. Школа№4 4

4 Самсонов Е.С, муж. Гимназия№114 5

5 Семенов И.К. муж. Лицей№10 5

Выберите пункт меню

1 – выполнить задание

0 – выход из программы

1

Выберите пункт меню

1 – выполнить задание

0 – выход из программы

0

Заключение

В процессе выполнения работы мы познакомились с работой с простейшими базами данных, генерацией внешнего нетекстового файла заданной структуры. Также получены полезные навыки работы с функциями: считывание из файла, запись в файл. Кроме того, работа с несколькими функциями дает возможность применять их в дальнейшем для реализации более сложных задач.

Файл – структура, состоящая из последовательности компонент одного типа. Свойства последовательности определяет последовательный доступ к элементам, т.е. в каждый момент времени может быть доступен только один элемент файла.

Реализация подобных структур помогает улучшить понимания представления, хранения и обработки данных в ЭВМ.

Список использованных источников

1. РосДиплом, Оформление таблиц в дипломной работе, особенности и требования ГОСТ/Электронный диплом/Режим доступа: <https://www.rosdiplom.ru/rd/pubdiplom/view.aspx?id=288>
2. Диплом Журнал, Оформление курсовой работы по ГОСТу 2019(образец)/Электронный диплом/Режим доступа: <https://journal.diplom.ru/kurovaya/oformlenie-kursovoj-raboty-po-gostu-2019-obrazec/>
3. Vyuchit.work – универсальная методичка/Электронный диплом/Режим доступа: <https://vyuchit.work/samorazvitie/sekretyi/oformlenie-risunkov-po-gostu.html>
4. Компилятор GCC/Электронный диплом/Режим доступа: <http://parallel.uran.ru/book/export/html/25>
5. Керниган, Брайан У., Ритчи, Деннис М. Язык программирования С, 2-е издание. :Пер. с англ. – М. : Издательский дом «Вильямс», 2009. – 304 с. : ил. – Парал. тит. англ.
6. CppStudio/Электронный диплом/Режим доступа: <http://cppstudio.com/post/1576/>
7. Примеры программ на Си/Электронный диплом/Режим доступа: http://www.school30.spb.ru/csd/ogt/samp/c_task_001.htm