Отчёт по заданию №6 курсового проекта

Черыгова Елизавета
Группа 8О-104Б

Оглавление

Цель работы

Разработать последовательную структуру данных для представления простейшей базы данных на файлах СП Си в соответствии с заданным вариантом. Составить программу генерации внешнего нетекстового файла заданной структуры, содержащего представительный набор записей (15-20). 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 получаемых из командной строки Unix. Второй способ задания параметров обязателен для работ, оцениваемых на хорошо и отлично. Параметры задаются с помощью ключей -f (распечатка файла) или -p (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 тестовые данные необходимо заранее поместить в текстовые файлы и распечатывать при протоколировании. Рекомендуется подобрать реальные и 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.

Задание:

Общая информация о выпускниках школы студента: фамилия, инициалы, пол, номер класса, буква класса, в каком ВУЗ-е учится, где работает, в каком полку служит и т. п.

Выяснить, имеются ли однофамильцы в каких-либо параллельных классах.

Алгоритм решения задачи

Есть условие: одинаковые фамилии, одинаковые классы(параллельные). проверяем это условие для всех, т.е. сравниваем первого со вторым, третьим, и до конца, потом второго с третьим, четвертым и так далее. Если мы уже нашли каких-то однофамильцев, то строки, в которых они записаны, отмечены, что бы их не читать и не проверять их по второму разу. В итоге получаются отмеченными строки, в которых сидят однофамильцы.

Кодпрограммы

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


int main(){
	char com[10], cur_file[50]; // com - тут хранится текущая команда, cur_file - имя открытого файла.
	strcpy(cur_file, "");
	FILE *f = NULL;
	print_help();


	while(strcmp(com, "exit")!=0){
		printf("Current file: %s\n", cur_file);
		printf("=:> ");
		scanf("%s", com); // считываем команду. Дальше в зависимости от выбора управление передается одной из функций.

			//описание функций в заголовочном файле .h
			if(strcmp(com, "help") == 0){
				print_help();
			}
			else if(strcmp(com, "clear")==0){
				system("cls"); // UNIX: system("clear");
			}
			else if(strcmp(com, "open")==0){
				fclose(f);
				f=open_file();
				strcpy(cur_file, filename);
			}
			else if(strcmp(com, "gen") == 0){
				srand(time(NULL));
				gen_db();
			}
			else if(strcmp(com,"exit")==0){
				;
			}
			else if(f==NULL){
				puts("There is no opened file!");
			}
			else if(f!=NULL){
				if(strcmp(com, "add")==0){
					add(f);
				}
```

```
                else if(strcmp(com, "del") == 0){
                        del(f);
                }
                else if(strcmp(com, "print") == 0){
                        print(f);
                }
                else if(strcmp(com, "close") == 0){
                        fclose(f);
                        strcpy(cur_file, "");
                        f=NULL;
                }
                else if(strcmp(com, "sol") == 0){
                        solution(f);
                }
        }
        else{
                if(f==NULL){
                        puts("There is no opened file!");
                }
                else{
                        puts("Unknown command.");
                }
        }
    }
    fclose(f);
    return 0;
}
```

```
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#ifndef __graduate_h__
#define __graduate_h__
#define MAX_GR 9

typedef struct{
	char name[50];
	char init[3];
	char gender[8];
	int cl_num;
	char cl_alph;
	char work[50];
}graduate;

char filename[50];

//-------------------------
int scan(graduate *g); //заполняет структуру, ввод с клавиатуры. возвращает количество считанных аргументов
void add(FILE *out); // заполняет файл, ввод из клавиатуры/из текстового файла файла
void print_help(); //очевидно
FILE* open_file(); // открывает файл и возвращает указатель на него
void rewrite(FILE *f, int size, int str_num); //перезаписывает файл, исключая нужную строку(str_num). f- укзатель на файл
// size - размер структуры
void del(FILE *f); //удаляет запись из f
void print(FILE *f); //печатает бд
graduate gen(); // генерирует структуру
void gen_db(); // заполняет файл с помощью функции выше
int count(FILE *f);//считает, сколько записей размера sizeof(graduate) имеется в файле(количество выпускников)
void solution(FILE* f); // собственно, задание
//-------------------------

void print_help(){
	puts("-------------------------------------");
	puts("open \"file\"                                  - open file.");
	puts("close                                          - close current file.");
	puts("add                                            - add graduate.");
	puts("  -f \"filename\"                              - from file.");
	puts("  -m \"information\"*                    - manually.");
	puts("gen \"filename\" \"number of graduates\"          - generate database.");
	puts("del \"student_name\"                           - delete student");
```

```
	puts("print                                    - print current file. ");
	puts("solution                              - solve the problem.");
	puts("help                                     - show help(this).");
	puts("clear                                    - clear screen.\n");
	puts("*graduate information format:\n\"name\" \"initials\" \"gender\" \"class
number\" \"class letter\" \"workplace\"\n");
	puts("-----------------------------------");
	return;
}

int scan(graduate *g){
	return scanf("%s %s %s %d %c %s", g->name, g->init, g->gender, &g->cl_num, &g-
>cl_alph, g->work);
}

void add(FILE *out){
	graduate g;
	char flag[3]; //переменная для параметра
	scanf("%s", flag);
	fseek(out,0,SEEK_END);
	if(strcmp(flag, "-f")==0){
		FILE *in;
		char o_filename[50]; //имя файла, из которого идет заполнение
		scanf("%s", o_filename);
		if ((in = fopen(o_filename, "r")) == NULL){
			printf("%s - no such file.\n", o_filename);
	      return;
		}
		//читаем из in и записываем в out
		while(fscanf(in, "%s %s %s %d %c %s", g.name, g.init, g.gender, &g.cl_num,
&g.cl_alph, g.work)==6){
			fwrite(&g, sizeof(g), 1, out);
		}
		puts("Base successfully filled.");
		fclose(in);
	}
	else if(strcmp(flag,"-m")==0){
		if(scan(&g)==6){ //считываем данные с клавиатуры
			fwrite(&g, sizeof(g), 1, out); //записываем в файл
		}
		else{
			puts("Something wrong.");
		}
	}
	else
		puts("Wrong parameter. Press ENTER.");
```

```
	return;
}

FILE* open_file(){
	FILE* f;
	scanf("%s", filename); //имя открываемого файла. переменная объявлена
глобальной, чтобы отображать открытый файл.
	if((f=fopen(filename,"a+b")) == NULL){
		puts("Error while opened file.");
		strcmp(filename, "");
	}
	else{
		puts("File opened.");
	}
	return f;
}

void rewrite(FILE *f, int size, int str_num){
  FILE *t;
  int i=0;
	graduate g;
  t=fopen("temp", "a+b");
  fseek(f, 0, SEEK_SET); //устанавливаем указатель на начало файла

  while(fread(&g, size, 1, f)==1){ //читаем запись
    if(i!=str_num){ // записываем все строки кроме i-й
      fwrite(&g, size, 1,t);
    }
    i++;
  }

  //записываем из временного файла обратно в основной
  fclose(f);
  f=fopen(filename, "w+b");
  fseek(t, 0, SEEK_SET);
  while(fread(&g, size, 1, t)==1){
    fwrite(&g, size, 1, f);
  }
  fclose(t);
  fclose(f);
  remove("temp"); // UNIX: system("rm temp");
  f=fopen(filename, "a+b");
	return;
}

void del(FILE *f){
```

```
	graduate g;
	int size=sizeof(g), i=0;
	char name[50]; //фамилия удаляемого
	char init[3]; // инициалы
	scanf("%s %s", name, init);
	fseek(f, 0, SEEK_SET);

	while(fread(&g, size, 1, f)==1){
		if(strcmp(name, g.name)==0 && strcmp(init, g.init)==0){ //сравниваем
имеющиеся фамилию и инициалы с записью
			rewrite(f,size,i);
			printf("%s %s deleted.\n", g.name, g.init);
			return;
		}
		++i;
	}
	printf("Graduate %s %s wasn't found.\n", name, init);
	return;
}

void print(FILE *f){
	graduate g;
	int i=1;
	fseek(f,0,SEEK_SET);
	printf("%s\n", filename);
	puts("=============================================");
	while(fread(&g, sizeof(g),1,f)==1){
		printf("%d. %s %s\t\t%s\t%d %c\t%s\n",i, g.name, g.init, g.gender, g.cl_num,
g.cl_alph, g.work);
		i++;
	}
	puts("=============================================");
	return;
}

graduate gen(){
  graduate g;
  int i;
  unsigned char
w[15][50]={{"Kalinina"},{"Mizulina"},{"Kuzmenko"},{"Mamedova"},{"Arefieva"},{"Basharina"},{"
Borisenko"},{"Gurskaya"},{"Krutikova"},
    {"Antonova"},{"Smertina"},{"Sinegubova"},{"Gorbunova"},{"Zlobina"},{"Rzhavina"}},

m[15][50]={{"Deryabin"},{"Milonov"},{"Zhirinovsky"},{"Malyshev"},{"Volok"},{"Zaycev"},{"Makar
ov"},{"Ilyin"},{"Shipunov"},{"Kovshov"},{"Senechkin"},
    {"Klimovich"},{"Sergeev"},{"Uvarov"},{"Kalashnikov"}},
```

```
        wplace[15][50]={{"IzGTU"},{"MPU"},{"MAI"},{"MIPT"},{"MSU"},{"MATI"},{"ZAVOD"},{"BIB
LIOTEKA"},{"UBORSHIK"},
        {"KASSIR"},{"LETCHIK"},{"BUS_IN_ASS_MAN"},{"PROGRAMMIST"},{"DANTIST"},{"MANA
GER"}};
    //список фамилий/мест работы

        g.init[0]='A'+rand()%26; //генерируем инициалы
  g.init[1]='A'+rand()%26;
  g.init[2]='\0';

  if(rand()%2){
    strcpy(g.name,w[rand()%15]); //если четное-девушка
    strcpy(g.gender,"female");
    strcpy(g.work,wplace[2+rand()%12]);
  }
  else{
    strcpy(g.name,m[rand()%15]); //нечетное - мужчина
    strcpy(g.gender,"male");
    strcpy(g.work,wplace[rand()%15]);
  }
  srand(rand());
  g.cl_num=7+rand()%5;
  g.cl_alph='A'+rand()%4;
  return g;
}

void gen_db(){
        FILE *db;
        char db_name[50];
        int number,i; //number-количество записей, i-счетчик
        graduate g;
        scanf("%s", db_name);
  scanf("%d", &number);

  db=fopen(db_name, "w+b");

  for(i=0;i<number;i++){
        g = gen();
        fwrite(&g, sizeof(g), 1, db);
  }
  fclose(db);
}

int count(FILE *f){
        graduate g;
        int begin, end;
```

```
	fseek(f, 0, SEEK_SET);
	begin=ftell(f);
	fseek(f,0,SEEK_END);
	end=ftell(f);
	return (end-begin)/sizeof(g);
}

void solution(FILE* f){ //функция, решающая задание
	graduate g1, g2;
	int sm_count=0; //счетчик однофамильцев
	int number=count(f); //количество записей в файле
	int size=sizeof(graduate); // размер одной записи
	int *fb_str,*sm; // строки, которые не надо читать(в них записаны однофамильцы людей, что выше по списку)
	int i,j,flag; // счетчики и флаг. флаг показывает, был записан первый однофамилец или нет.
	fb_str= (int*)malloc(number*sizeof(int));
	sm=(int*)malloc(number/2*sizeof(int));

	for(i=0;i<number;i++){
		fb_str[i]=-1; //-1 означает, что строку нужно читать. По умолчанию читаются все строки
		// строки, в которых записаны однофамильцы не читаются
	}
	//-------------
	fseek(f,0, SEEK_SET); //установили указатель на начало файла

	for(i=0;i<number-1;i++){ //начинаем перебирать записи. сравниваем выпускника со всеми последующими, и так с каждым, кроме уже найденных однофамильцев
		flag=0;

		if(fb_str[i]==-1){ //если в строке не однофамилец, читать
			fseek(f,i*size, SEEK_SET);
			fread(&g1, size, 1, f);

			for(j=i+1;j<number;j++){//ищем однофвмильцев
				if(fb_str[j]==-1){//если в строке не однофамилец, читать
					fseek(f,j*size, SEEK_SET);
					fread(&g2, size, 1, f);

					if(strcmp(g1.name,g2.name)==0 && g1.cl_num==g2.cl_num){ //если фамилии одинаковые и классы параллельные,
					// заносим номер записи в fb_str и инкрементируем счетчик
						if(!flag){ //Обладатель фамилии, сравниваемой с остальными
```

```
                                        sm[sm_count]=i;
                                        fb_str[i]=0;
                                        sm_count++;
                                        flag=1;
                                }
                        sm[sm_count]=j; //его однофамилец
                        fb_str[j]=0;
                        sm_count++;
                        }

                }
        }

    }

}
//-------------
// выводим на экран всех однофамильцев в параллельных классах.
for(i=0;i<sm_count;i++){
    fseek(f,sm[i]*size,SEEK_SET);
    fread(&g1, size, 1, f);
    printf("%3.d. %-20s %s, %2.d %c\n",sm[i]+1, g1.name, g1.init, g1.cl_num,
g1.cl_alph);
}
}


#endif
```

Вывод программы

```
-----------------------------------
open "file"                                     - open file.
close                                           - close current file.
add                                             - add graduate.
        -f "filename"                           - from file.
        -m "information"*                       - manually.
gen "filename" "number of graduates"            - generate database.
del "student_name"                              - delete student
print                                           - print current file.
solution                                        - solve the problem.
help                                            - show help(this).
clear                                           - clear screen.

*graduate information format:
"name" "initials" "gender" "class number" "class letter" "workplace"

-----------------------------------
Current file:
=:> gen spisok 30
Current file:
=:> open spisok
File opened.
Current file: spisok
=:> print
spisok
=============================================
  1. Basharina            NX female   11 B UBORSHIK
  2. Milonov              JR male      8 C BIBLIOTEKA
  3. Sergeev              HF male     10 C LETCHIK
  4. Borisenko            KL female    8 C LETCHIK
  5. Smertina             IB female   11 B MSU
  6. Mizulina             YE female   10 A PROGRAMMIST
  7. Gurskaya             UK female    9 B MIPT
  8. Sinegubova           TJ female    7 C PROGRAMMIST
  9. Kalinina             QZ female    8 C BIBLIOTEKA
 10. Volok                IB male     11 B MATI
 11. Gurskaya             JL female    7 C MAI
 12. Zaycev               FL male      9 D MAI
 13. Kalinina             SM female   11 A UBORSHIK
 14. Arefieva             XZ female    7 C KASSIR
 15. Kalashnikov          EZ male      9 A PROGRAMMIST
 16. Uvarov               RD male      7 D MIPT
 17. Deryabin             CF male     11 C UBORSHIK
 18. Kuzmenko             EV female    8 D UBORSHIK
 19. Shipunov             ZE male      8 C DANTIST
 20. Malyshev             UK male      7 C KASSIR
 21. Krutikova            ZJ female    8 C PROGRAMMIST
 22. Kalinina             YI female   11 D PROGRAMMIST
 23. Zlobina              RZ female    7 B UBORSHIK
 24. Borisenko            YN female    8 B DANTIST
 25. Uvarov               FV male     10 C MIPT
 26. Uvarov               NJ male     11 B BIBLIOTEKA
 27. Mamedova             DC female    8 D MAI
 28. Milonov              UX male      7 D ZAVOD
 29. Volok                OJ male      7 A MIPT
 30. Malyshev             UK male      7 A MATI
```

```
==========================================
Current file: spisok
=:> add -m Adrienko VV male 8 G MSU
Current file: spisok
=:> add -m Volkova AA female 11 D MAI
Current file: spisok
=:> add -m Shishkina KA female 11 B MIET
Current file: spisok
=:> add -m Karpov LA male 11 G VUMO
Current file: spisok
=:> add -m Kuznechov DA male 11 D IzGTU
Current file: spisok
=:> print
spisok
==========================================
  1. Basharina            NX female   11 B UBORSHIK
  2. Milonov              JR male      8 C BIBLIOTEKA
  3. Sergeev              HF male     10 C LETCHIK
  4. Borisenko            KL female    8 C LETCHIK
  5. Smertina             IB female   11 B MSU
  6. Mizulina             YE female   10 A PROGRAMMIST
  7. Gurskaya             UK female    9 B MIPT
  8. Sinegubova           TJ female    7 C PROGRAMMIST
  9. Kalinina             QZ female    8 C BIBLIOTEKA
 10. Volok                IB male     11 B MATI
 11. Gurskaya             JL female    7 C MAI
 12. Zaycev               FL male      9 D MAI
 13. Kalinina             SM female   11 A UBORSHIK
 14. Arefieva             XZ female    7 C KASSIR
 15. Kalashnikov          EZ male      9 A PROGRAMMIST
 16. Uvarov               RD male      7 D MIPT
 17. Deryabin             CF male     11 C UBORSHIK
 18. Kuzmenko             EV female    8 D UBORSHIK
 19. Shipunov             ZE male      8 C DANTIST
 20. Malyshev             UK male      7 C KASSIR
 21. Krutikova            ZJ female    8 C PROGRAMMIST
 22. Kalinina             YI female   11 D PROGRAMMIST
 23. Zlobina              RZ female    7 B UBORSHIK
 24. Borisenko            YN female    8 B DANTIST
 25. Uvarov               FV male     10 C MIPT
 26. Uvarov               NJ male     11 B BIBLIOTEKA
 27. Mamedova             DC female    8 D MAI
 28. Milonov              UX male      7 D ZAVOD
 29. Volok                OJ male      7 A MIPT
 30. Malyshev             UK male      7 A MATI
 31. Adrienko             VV male      8 G MSU
 32. Volkova              AA female   11 D MAI
 33. Shishkina            KA female   11 B MIET
 34. Karpov               LA male     11 G VUMO
 35. Kuznechov            DA male     11 D IzGTU
==========================================
Current file: spisok
=:> del Kuznechov DA male 11 D IzGTU
Kuznechov DA deleted.
Current file: spisok
=:> Current file: spisok
=:> Current file: spisok
=:> Current file: spisok
```

```
=:> Current file: spisok
=:> print
spisok
==========================================
  1. Basharina            NX female   11 B UBORSHIK
  2. Milonov              JR male      8 C BIBLIOTEKA
  3. Sergeev              HF male     10 C LETCHIK
  4. Borisenko            KL female    8 C LETCHIK
  5. Smertina             IB female   11 B MSU
  6. Mizulina             YE female   10 A PROGRAMMIST
  7. Gurskaya             UK female    9 B MIPT
  8. Sinegubova           TJ female    7 C PROGRAMMIST
  9. Kalinina             QZ female    8 C BIBLIOTEKA
 10. Volok                IB male     11 B MATI
 11. Gurskaya             JL female    7 C MAI
 12. Zaycev               FL male      9 D MAI
 13. Kalinina             SM female   11 A UBORSHIK
 14. Arefieva             XZ female    7 C KASSIR
 15. Kalashnikov          EZ male      9 A PROGRAMMIST
 16. Uvarov               RD male      7 D MIPT
 17. Deryabin             CF male     11 C UBORSHIK
 18. Kuzmenko             EV female    8 D UBORSHIK
 19. Shipunov             ZE male      8 C DANTIST
 20. Malyshev             UK male      7 C KASSIR
 21. Krutikova            ZJ female    8 C PROGRAMMIST
 22. Kalinina             YI female   11 D PROGRAMMIST
 23. Zlobina              RZ female    7 B UBORSHIK
 24. Borisenko            YN female    8 B DANTIST
 25. Uvarov               FV male     10 C MIPT
 26. Uvarov               NJ male     11 B BIBLIOTEKA
 27. Mamedova             DC female    8 D MAI
 28. Milonov              UX male      7 D ZAVOD
 29. Volok                OJ male      7 A MIPT
 30. Malyshev             UK male      7 A MATI
 31. Adrienko             VV male      8 G MSU
 32. Volkova              AA female   11 D MAI
 33. Shishkina            KA female   11 B MIET
 34. Karpov               LA male     11 G VUMO
==========================================
Current file: spisok
=:>close
Current file:
=:>help
-----------------------------------
open "file"                                    - open file.
close                                          - close current file.
add                                            - add graduate.
        -f "filename"                          - from file.
        -m "information"*                      - manually.
gen "filename" "number of graduates"           - generate database.
del "student_name"                             - delete student
print                                          - print current file.
solution                                       - solve the problem.
help                                           - show help(this).
clear                                          - clear screen.

*graduate information format:
"name" "initials" "gender" "class number" "class letter" "workplace"
```

```
-----------------------------------
Current file:
=:> open spisok.spisok
File opened.
Current file: spisok.spisok
=:> print
spisok.spisok
==========================================
  1. Smertina             LM female    8 B MSU
  2. Uvarov               SS male      9 B MANAGER
  3. Volok                QY male     10 A BIBLIOTEKA
  4. Shipunov             WY male      7 C NACHALNIK CEHA
  5. Senechkin            GJ male      7 B MANAGER
  6. Rzhavina             IX female    8 A MANAGER
  7. Zaycev               VS male      7 B MSU
  8. Volok                RE male      9 C ZAVOD
  9. Zaycev               ZR male     10 B MANAGER
 10. Smertina             OO female    7 C MIPT
 11. Antonova             HW female    9 A PROGRAMMIST
 12. Makarov              LR male      9 D BUS_IN_ASS_MAN
 13. Zhirinovsky          KM male      9 C UBORSHIK
 14. Deryabin             OL male     11 A UBORSHIK
 15. Gorbunova            FZ female    8 C BIBLIOTEKA
 16. Deryabin             PX male     11 D KASSIR
 17. Milonov              VK male      8 A PROGRAMMIST
 18. Kuzmenko             ZN female   11 B MSU
 19. Sinegubova           CP female    8 A MANAGER
 20. Kalinina             TV female    9 C LETCHIK
 21. Rzhavina             SI female    9 D ZAVOD
 22. Shipunov             SV male      8 A UBORSHIK
 23. Borisenko            VO female    8 D MANAGER
 24. Arefieva             OJ female   10 C MSU
 25. Deryabin             AA male      8 B PROGRAMMIST
 26. Kalashnikov          KD male      8 B PROGRAMMIST
 27. Smertina             OG female    8 B UBORSHIK
 28. Antonova             QC female   11 C MAI
 29. Uvarov               CY male      9 D LETCHIK
 30. Kalinina             DD female   10 A LETCHIK
 31. Kalinina             AA female   10 B KASSIR
 32. Zhirinovsky          AR male      9 A MANAGER
 33. Deryabin             OV male     11 G DEPUTAT
==========================================
Current file: spisok.spisok
=:> solution
  1. Smertina             LM,  8 B
 27. Smertina             OG,  8 B
  2. Uvarov               SS,  9 B
 29. Uvarov               CY,  9 D
 13. Zhirinovsky          KM,  9 C
 32. Zhirinovsky          AR,  9 A
 14. Deryabin             OL, 11 A
 16. Deryabin             PX, 11 D
 33. Deryabin             OV, 11 G
 30. Kalinina             DD, 10 A
 31. Kalinina             AA, 10 B
Current file: spisok.spisok
=:>
```

Заключение

Благодаря, этому заданию, я смогла поближе познакомиться и разобраться с представлением в Си простейшей базы данных, а также научилась работать с этими данными.