

Лабораторная работа № 6

Файловый ввод-вывод.

Задание 1.1

Постановка задачи:

Напишите программу, которая записывает массив одинаковых структур с данными о студентах в текстовый файл формате CSV в режиме "добавления" ("a"), а затем читает построчно этот файл в другой массив структур в режиме "чтения" ("r").

```
struct Student {  
    unsigned int id;  
    char[50] name;  
    char[50] faculty;  
    float rating;  
};
```

В структуре перечислены:

1. id - уникальный идентификатор,
2. name - фамилия и имя,
3. faculty - название факультета,
4. rating - текущий рейтинг студента.

Данные о студентах вводятся вами с клавиатуры. Используйте функции файлового ввода-вывода.

Математическая модель:

-

Список идентификаторов:

Имя	Тип	Назначение
n	int	Количество студентов на ввод и далее на вывод.
students	struct Student *	Динамический массив вводимых структур студентов.
i	int	Аргумент цикла.
file	FILE	Файл для записи/чтения.

buf	char *[255]	Буфер для строки с данными одного студента.
tmp	char *	Указатель на искомое значение в строке.
students2	struct Student *	Массив в который будут считываться структуры из файла.

Код программы:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 struct Student {
6     unsigned id;
7     char name[50];
8     char faculty[50];
9     float rating;
10 };
11
12
13 int
14 main()
15 {
16     int n;
17     printf("Enter the ammount of students:");
18     scanf("%d", &n);
19     getchar();
20
21     struct Student *students = NULL;
22     students = malloc(sizeof(struct Student)*n);
23
24     if(!students){
25         printf("Malloc failed\n");
26         exit(1);
27     }
28
29     for(int i = 0; i < n; i++){
30         printf("\nEnter student's № %d info:\n", i);
31
32         printf("\tid:");
33         scanf("%u", &students[i].id);
34         getchar();
35
36         printf("\tname:");
37         fgets(students[i].name, sizeof(students[i].name), stdin);
38         students[i].name[strcspn(students[i].name, "\n")] = 0;
39
40         printf("\tfaculty:");
41         fgets(students[i].faculty, sizeof(students[i].faculty), stdin);
42         students[i].faculty[strcspn(students[i].faculty, "\n")] = 0;
43
44         printf("\trating:");
45         scanf("%f", &students[i].rating);
46         getchar();
47     }
```

```

48
49 FILE *file = NULL;
50 file = fopen("students.csv", "a");
51
52
53 for(int i = 0; i < n; i++){
54     fprintf(file, "%u,%s,%s,%f\n", students[i].id, students[i].name, students[i].faculty, students[i].rating);
55 }
56
57 fclose(file);
58 free(students);
59
60
61 puts("Students to read:");
62 scanf("%d",&n);
63 getchar();
64
65 file = fopen("students.csv","r");
66 if(!file){
67     puts("Could not open file!");
68     exit(0);
69 }
70
71 char *buf = NULL;
72 buf = malloc(255);
73 char *tmp;
74
75 struct Student *students2 = NULL;
76 students2 = malloc(sizeof(struct Student)*n);
77
78 if(!buf || !students2){
79     puts("Malloc failed!");
80     exit(0);
81 }
82
83 int i = 0;
84 while(fgets(buf, sizeof(struct Student), file) != NULL){
85     if ((strlen(buf)>0) && (buf[strlen (buf) - 1] == '\n'))
86         buf[strlen (buf) - 1] = '\0';
87
88     tmp = strtok(buf, ",");
89     students2[i].id = atoi(tmp);
90
91     tmp = strtok(NULL, ",");
92     strcpy(students2[i].name, tmp);
93
94     tmp = strtok(NULL, ",");
95     strcpy(students2[i].faculty, tmp);
96
97     tmp = strtok(NULL, ",");
98     students2[i].rating = atof(tmp);
99     i++;
100 }
101 fclose(file);
102 puts("From file:");
103
104 for(int i = 0; i < n; i++){
105     printf("Student %d\n\tid:%u\n\tname: %s\n\tfaculty: %s\n\ttrating: %f\n", i, students2[i].id, students2[i].name, students2[i].faculty, students2[i].rating);
106 }
107 free(students2);
108 free(buf);
109 return 0;
110 }

```

Результат выполненной работы:

```
Enter the amount of students:2
```

```
Enter student's № 0 info:
```

```
id:125  
name:qwerty  
faculty:ytrewq  
rating:0
```

```
Enter student's № 1 info:
```

```
id:126  
name:dvorak  
faculty:karvod  
rating:12
```

```
Students to read:
```

```
4
```

```
From file:
```

```
Student №0
```

```
id:123  
name:Maxwel  
faculty:IT  
rating:5.800000
```

```
Student №1
```

```
id:124  
name:anna  
faculty:humanities  
rating:6.200000
```

```
Student №2
```

```
id:125  
name:qwerty  
faculty:ytrewq  
rating:0.000000
```

```
Student №3
```

```
id:126  
name:dvorak  
faculty:karvod  
rating:12.000000
```

Файл students.csv:

```
123,Maxwel,IT,5.800000  
124,anna,humanities,6.200000  
125,qwerty,ytrewq,0.000000  
126,dvorak,karvod,12.000000
```

Задание 1.2

Постановка задачи:

Напишите программу, которая записывает массив одинаковых структур с данными о студентах в двоичный (бинарный) файл в режиме "записи" (перезаписи) ("wb"), а затем читает этот файл в другой массив структур в режиме "чтения" ("rb").

```
struct Student {  
    unsigned int id;  
    char[50] name;  
    char[50] faculty;  
    float rating;  
};
```

В структуре перечислены:

1. id - уникальный идентификатор,
2. name - фамилия и имя,
3. faculty - название факультета,
4. rating - текущий рейтинг студента.

2

Используйте текст лекции и документацию по ссылкам выше для чтения/записи в двоичном режиме.

Математическая модель:

-

Список идентификаторов:

Имя	Тип	Назначение
n	int	Количество студентов на ввод и далее на вывод.
students	struct Student *	Динамический массив вводимых структур студентов.
i	int	Аргумент цикла.
file	FILE	Файл для записи/чтения.
students2	struct Student *	Массив в который будут считываться структуры из файла.

Код программы:

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 struct Student {
6     unsigned id;
7     char name[50];
8     char faculty[50];
9     float rating;
10 };
11
12
13 int
14 main()
15 {
16     int n;
17     printf("Enter the ammount of students:");
18     scanf("%d", &n);
19     getchar();
20
21     struct Student *students = NULL;
22     students = malloc(sizeof(struct Student)*n);
23
24     if(!students){
25         printf("Malloc failed\n");
26         exit(1);
27     }
28     for(int i = 0; i < n; i++){
29         printf("\nEnter student's № %d info:\n", i);
30
31         printf("\tid:");
32         scanf("%u", &students[i].id);
33         getchar();
34
35         printf("\tname:");
36         fgets(students[i].name, sizeof(students[i].name), stdin);
37         students[i].name[strcspn(students[i].name, "\n")] = 0;
38
39         printf("\tfaculty:");
40         fgets(students[i].faculty, sizeof(students[i].faculty), stdin);
41         students[i].faculty[strcspn(students[i].faculty, "\n")] = 0;
42
43         printf("\trating:");
44         scanf("%f", &students[i].rating);
45         getchar();
46     }

```

```

47
48 FILE *file = NULL;
49 file = fopen("students.bin", "wb");
50
51 fwrite(&students[0], sizeof(struct Student), n, file);
52
53 fclose(file);
54 free(students);
55
56
57 puts("Students to read:");
58 scanf("%d",&n);
59 getchar();
60
61 file = fopen("students.bin","rb");
62 if(!file){
63     puts("Could not open file!");
64     exit(0);
65 }
66
67 struct Student *students2 = NULL;
68 students2 = malloc(sizeof(struct Student)*n);
69
70 if(!students2){
71     puts("Malloc failed!");
72     exit(0);
73 }
74
75 fread(students2, sizeof(struct Student), n, file);
76
77 fclose(file);
78 puts("From file:");
79
80 for(int i = 0; i < n; i++)
81     printf("Student №%d\n\tid:%u\n\tname:%s\n\tfaculty:%s\n\trating:%f\n", i, students2[i].id, students2[i].name, students2[i].faculty, students2[i].rating);
82
83 free(students2);
84 return 0;
85 }

```

Результат выполненной работы:

```

Enter the ammount of students:2

Enter student's № 0 info:
    id:1
    name:qwerty
    faculty:ytrewq
    rating:12

Enter student's № 1 info:
    id:2
    name:dvorak
    faculty:karovd
    rating:10

Students to read:
2
From file:
Student №0
    id:1
    name:qwerty
    faculty:ytrewq
    rating:12.000000

Student №1
    id:2
    name:dvorak
    faculty:karovd
    rating:10.000000

```