

Лабораторна робота №8.

Тема: "Блоковий ввід-вивід"

Мета: Робота із двійковими файлами, організація вводу-виводу структурованої інформації і її зберігання на зовнішніх носіях.

1. Короткі теоретичні відомості

1.1. Ввід і вивід в C

Особливістю C є відсутність у цій мові структурованих файлів. Всі файли розглядаються як не структурована послідовність байтів. При такому підході поняття файлу поширюється й на різні пристрої.

У C відсутні засоби вводу-виводу. Всі операції вводу-виводу реалізуються за допомогою функцій, які перебувають у бібліотеці C. Бібліотека C підтримує три рівні вводу-виводу:

- потоковий ввід-вивід;
- ввід-вивід нижнього рівня;
- ввід-вивід для консолі й портів (залежить від ОС).

1.2. Потоковий ввід-вивід

На рівні потокового вводу-виводу обмін даними відбувається побайтно, тобто за одне звертання до пристрою (файлу) відбувається зчитування або запис фіксованої порції даних (512 чи 1024 байти). При вводі з диска або при зчитуванні з файлу дані поміщаються в буфер ОС, потім побайтно або порціями передаються в програмі користувача. При виводі у файл дані накопичуються в буфері, а при заповненні буфера записуються у вигляді єдиного блоку на диск. Буфери ОС реалізуються у вигляді ділянок основної пам'яті. Функції бібліотеки C, що підтримують обмін, з даними на рівні потоку дозволяють обробляти дані різних розмірів і форматів.

Потік - це файл разом з наданими засобами буферизації. При роботі з потоком можна:

- 1) Відкривати й закривати потоки (зв'язувати вказівники на потік з конкретними файлами);

- 2) вводить і виводити рядок, символ, форматовані дані, порцію даних довільної довжини;
 - 3) аналізувати помилки вводу-виводу й досягнення кінця файлу;
 - 4) управляти буферизацією потоку й розміром буфера;
 - 5) отримувати й встановлювати вказівник поточної позиції у файлі;
- Функції бібліотеки вводу-виводу перебувають у заголовковому файлі <stdio.h>.

1.3. Відкриття й закриття потоку

Перш ніж почати працювати з потоком, його треба ініціювати, тобто відкрити. При цьому потік зв'язується зі структурою визначеного типу FILE, визначення якої перебуває в бібліотечному файлі <stdio.h>. У структурі перебуває вказівник на буфер, вказівник на поточну позицію файлу й т.ін. При відкритті потоку, повертається вказівник на потік, тобто на об'єкт типу FILE.

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

```
.....
```

```
FILE *fp;
```

```
.....
```

```
fp= fopen( "t.txt", "r");
```

де fopen(<ім'я_файлу>,<режим_відкриття>) - функція для ініціалізації файлу.

Існують такі режими для відкриття файлу:

"w" - відкрити файл для запису, якщо файл існує, то він стирається;

"r" - відкрити файл для читання;

"a" - відкрити файл для додавання, якщо файл існує, то він не стирається і можна писати в кінець файлу;

"w+" - відкрити файл для запису й виправлення, якщо файл існує, то він стирається, а далі можна і читати, і писати, розміри файлу можна збільшувати;

"r+" - відкрити файл для читання й запису, але збільшити розмір файлу не можна;

"a+" - відкрити файл для додавання, тобто можна й читати й писати, у тому числі й у кінець файлу.

Потік можна відкрити в текстовому (t) або двійковому (b) режимі. За замовчуванням - текстовий режим. У явному виді режим вказується в такий спосіб: "r+b" або "rb" - двійковий (бінарний) режим.

Приклад:

```
if ((fp=fopen("t.txt", "w"))==NULL)
{
    perror("\nпомилка при відкритті файлу");
    // виводить рядок символів з повідомленням
    // про помилку
    exit(0);
}
```

Після роботи з файлом, його треба закрити

```
fclose(<вказівник_на_потік>);
```

1.4. Блоковий ввід-вивід

Для блокового вводу і виводу використовуються функції:

1) int fread(void *ptr, int size, int n, FILE *fp) , де

void *ptr - вказівник на область пам'яті, у якій розташовуються зчитувані з файлу дані;

int size - розмір одного зчитуваного елемента;

int n - кількість зчитуваних елементів;

FILE *fp - вказівник на файл, з якого відбувається зчитування.

У випадку успішного зчитування інформації функція повертає число прочитаних елементів (а не байтів), інакше повертає EOF.

2) int fwrite(void *ptr, int size, int n, FILE *fp) , де

void *ptr - вказівник на область пам'яті, у якій розташовуються записувані у файл дані;

int size - розмір одного записуваного елемента;

int n - кількість записуваних елементів;

FILE *fp - вказівник на файл, у який відбувається запис.

У випадку успішного запису інформації функція повертає число записаних елементів, інакше повертає EOF.

Приклад:

.....

```
typedef struct
{
char name [40];
char post [40];
float rate;
}EMPLOYEE;

void main ()
{
FILE *f;    // вказівник пов'язаний з файлом
EMPLOYEE e;    // змінна
EMPLOYEE mas[10]    //масив
//відкриваємо файл
if ((f=fopen("f.dat", "wb"))==NULL) exit(1);
// якщо при відкритті файлу виникає
//помилка, то виходимо з функції
int i;
for(i=1; i<=10;i++)
{
//формуємо запис e
printf("name="); scanf("%s",&e.name);
printf("post="); scanf("%s",&e.post);
printf("rate="); scanf("%f",e.rate);
// записуємо запис e у файл
fwrite(&e, sizeof(EMPLOYEE),1,f);
if (ferror(f)==NULL) exit(2);
}
fclose(f);
```

```

//читання записів з файлу
if ((f=fopen("f.dat", "rb")==NULL) exit(3);
// якщо при відкритті файлу виникає
//помилка, то виходимо з функції
i=0;
while(!feof(f)&& i<=10)
{
    fread(&mas[i], sizeof(EMPLOYEE),1,f);
    i++;
}
fclose(f);
}

```

2. Постановка завдання

Сформувати двійковий файл із елементів, заданої у варіанті структури, роздрукувати його вміст, виконати знищення й додавання елементів у відповідності зі своїм варіантом, використовуючи для пошуку елементів що знищуються чи додаються, функцію. Формування, друк, додавання й знищення елементів оформити у вигляді функцій. Передбачити повідомлення про помилки при відкритті файлу й виконанні операцій вводу/виводу.

3. Варіанти

1. Структура "Абітурієнт":

- прізвище, ім'я, по батькові;
- рік народження;
- оцінки вступних іспитів (3);
- середній бал атестата.

Знищити елемент із зазначеним номером, додати елемент після елемента із зазначеним прізвищем.

2. Структура "Співробітник":

- прізвище, ім'я, по батькові;

- посада
- рік народження;
- заробітна плата.

Знищити елемент із зазначеним прізвищем, додати елемент після елемента із зазначеним номером.

3. Структура "Держава":

- назва;
- столиця;
- чисельність населення;
- займана площа.

Знищити всі елементи, у яких чисельність менше заданої, додати елемент після елемента із зазначеним номером.

4. Структура "Людина":

- прізвище, ім'я, по батькові;
- домашня адреса;
- номер телефону;
- вік.

Знищити усі елементи із заданим віком, додати елемент після елемента із заданим номером.

5. Структура "Людина":

- прізвище, ім'я, по батькові;
- рік народження;
- ріст;
- вага.

Знищити усі елементи із зазначеним ростом і вагою, додати елемент після елемента із зазначеним прізвищем.

6. Структура "Школяр":

- прізвище, ім'я, по батькові;
- клас;
- номер телефону;
- оцінки по предметах (математика, фізика, українська мова, література).

Знищити всі елементи, у яких є 2 хоча б з одного предмету, додати елемент у початок файлу.

7. Структура "Студент":

- прізвище, ім'я, по батькові;
- домашня адреса;
- група;
- рейтинг.

Знищити всі елементи, у яких рейтинг менше заданого, додати 1 елемент у кінець файлу.

8. Структура "Покупець":

- прізвище, ім'я, по батькові;
- домашня адреса;
- номер телефону;
- номер кредитної картки.

Знищити 3 елементи з початку файлу, додати 3 елементи в кінець файлу.

9. Структура "Пацієнт":

- прізвище, ім'я, по батькові;
- домашня адреса;
- номер медичної карти;
- номер страхового поліса.

Знищити елемент із заданим номером медичної карти, додати 2 елементи в початок файлу.

10. Структура "Інформація":

- носій;
- об'єм;
- назва;
- автор.

Знищити перший елемент із заданим об'ємом інформації, додати елемент перед елементом із зазначеним номером.

11. Структура "Відеокасета":

- назва фільму;

- режисер;
- тривалість;
- ціна.

Знищити всі елементи із ціною вищою заданої, додати 3 елементи в кінець файлу.

12. Структура "Музичний диск":

- назва;
- автор;
- тривалість;
- ціна.

Знищити перший елемент із заданою тривалістю, додати 2 елементи після елемента із заданим номером.

13. Структура "Спортивна команда":

- назва;
- місто;
- кількість гравців;
- кількість набраних очків.

Знищити всі елементи з кількістю очків менше заданого, додати 2 елементи на початок файлу.

14. Структура "Стадіон":

- назва;
- адреса;
- місткість;
- види спорту.

Знищити елемент із заданою назвою, додати 2 елементи після елемента із зазначеним номером.

15. Структура "Автомобіль":

- марка;
- рік випуску;
- ціна;
- кольори.

Знищити всі елементи, у яких рік випуску менше заданого, додати елемент на початок файлу.

16. Структура "Власник автомобіля":

- прізвище, ім'я, по батькові;
- номер автомобіля;
- телефон;
- номер техпаспорта.

Знищити елемент із заданим номером, додати 2 елементи перед елементом із заданим прізвищем.

17. Структура "Фільм":

- назва;
- режисер;
- рік випуску;
- вартість.

Знищити всі елементи, у яких вартість перевищує задану, додати елемент на початок файлу.

18. Структура "Книга":

- назва;
- автор;
- рік видання;
- кількість сторінок.

Знищити 3 елементи з початку файлу, додати елемент перед елементом із зазначеною назвою.

19. Структура "Фільм":

- назва;
- режисер;
- країна;
- прибуток.

Знищити 2 елементи з кінця файлу, додати елемент після елемента із зазначеною назвою.

20. Структура "Держава":

- назва;
- державна мова;
- грошова одиниця;
- курс валюти відносно \$.

Знищити елемент із зазначеною назвою, додати 2 елементи в кінець файлу.

21. Структура "Автомобіль":

- марка;
- серійний номер;
- реєстраційний номер;
- рік випуску.

Знищити 3 елементи з початку файлу, додати елемент поле елемента із зазначеним реєстраційним номером.

22. Структура "Власник автомобіля":

- прізвище, ім'я, по батькові;
- номер автомобіля;
- номер техпаспорта;
- відділення реєстрації ДАІ.

Знищити елемент із заданим номером, додати 2 елементи перед елементом із заданим прізвищем.

23. Структура "Стадіон":

- назва;
- рік будівлі;
- кількість площадок;
- види спорту.

Знищити всі елементи, у яких рік будівлі менше заданого, додати 2 елементи перед елементом із зазначеним номером.

24. Структура "Студент":

- прізвище, ім'я, по батькові;
- номер телефону;
- група;
- оцінки по 3 основних предметах.

Знищити всі елементи із групи із зазначеним номером, у яких середнє арифметичне оцінок менше заданого, додати елемент після елемента із заданим прізвищем.

25. Структура "Студент":

- прізвище, ім'я, по батькові;
- дата народження;
- домашня адреса;
- рейтинг.

Знищити елементи, у яких дати народження збігаються, додати елемент перед елементом із заданим прізвищем.

4. Методичні вказівки

1. Для заповнення файлу можна використовувати функцію, що формує одну структуру, зазначеного у варіанті типу. Значення елементів структури вводяться із клавіатури. Для вводу можна використовувати операцію >> і функцію gets().
2. При вводі структур можна реалізувати один з таких механізмів:
 - ввід заздалегідь обраної кількості структур (не менше 5);
 - ввід до появи структури із заданою кількістю ознак;
 - діалог з користувачем про необхідність продовжувати ввід.
3. Для запису структури у файл і читання структури з файлу використовувати функції блокового вводу/виводу fread й fwrite.
4. Для знищення/додавання елементів у файл використовувати допоміжний файл.

При програмній реалізації можна скористатись наступним фрагментом програми.

```
#include <stdio.h>
#include <stdlib.h>
typedef struct
{
    char marka[40];
    char ser_numder[40];
    char reest_numder[40];
    int year;
}CAR;
```

```

int main()
{

FILE *f;      // вказівник пов'язаний з файлом
CAR m;        // змінна
CAR mas[10]; //масив
//відкриваємо файл
if ((f=fopen("f.dat", "wb"))==NULL) exit(1);
// якщо при відкритті файлу виникає
//помилка, то виходимо з функції
int i;
for(i=1; i<=2;i++)
{
//формуємо запис m
printf("marka="); scanf("%s", m.marka); printf("%s\n", &m.marka);
printf("ser_number="); scanf("%s", m.ser_numder); printf("%s\n", &m.ser_numder);
printf("reest_number="); scanf("%s", m.reest_numder); printf("%s\n", &m.reest_numder);
printf("year="); scanf("%i", &m.year); printf("%i\n", m.year);

// записуємо запис m у файл
fwrite(&m, sizeof(CAR),1,f);
//if (ferror(f)==NULL) exit(2);
}
fclose(f);
//читання записів з файлу
if ((f=fopen("f.dat", "rb"))==NULL) {printf("!!2\n"); return 1;}
// якщо при відкритті файлу виникає
//помилка, то виходимо з функції
i=0;
while(!feof(f)&& i<=2)
{
fread(&mas[i], sizeof(CAR),1,f);
i++;
}
for(i=0; i<=1; i++)
{
printf("object: %i\n", i);
printf("%s\n", mas[i].marka);
printf("%s\n", mas[i].reest_numder);
printf("%s\n", mas[i].ser_numder);
printf("%i\n", mas[i].year);
}
fclose(f);

return 0;
}

```

5. Зміст звіту

1. Постановка завдання.
2. Опис використовуваних типів даних.

3. Текст функцій для:

- формування файлу,
- друку файлу,
- додавання запису у файл,
- знищення запису з файлу
- пошуку структури для знищення.

4. Результат розв'язку конкретного варіанту.