

Лекція. Робота з файлами

робота з файлом

Базові функції для роботи з файлами описані в бібліотеці **stdio.h**.
Вся робота з файлом виконується через файлову змінну - покажчик на структуру типу FILE:

FILE * fp;

Для відкриття файлу використовуються функція `fopen`, що має 2 параметри:

FILE * fopen (char * ім'я файлу, char * режим доступу);

Параметр *режим_доступу* визначає, які дії будуть дозволені з файлом:

“r” - відкриваємо для читання текстовий файл;

“a” - відкриваємо текстовий файл для додавання даних в кінець файлу;

“w” - відкриваємо файл для запису.

Команди для роботи з файлом

Назва	Що робить
<code>fopen()</code>	Відкриває файл
<code>fclose()</code>	Закриває файл
<code>putc()</code>	Записує символ у файл
<code>fputc()</code>	Записує символ у файл
<code>getc()</code>	Читає символ з файлу
<code>fgetc()</code>	Читає символ з файлу
<code>fgets()</code>	Читає рядок з файлу
<code>fputs()</code>	Записує рядок у файл
<code>fseek()</code>	Встановлює покажчик поточної позиції на певний байт файлу
<code>ftell()</code>	Повертає поточне значення покажчика у файлі
<code>fprintf()</code>	Для файлу те саме, що <code>printf()</code> для консолі
<code>fscanf()</code>	Для файлу те саме, що <code>scanf()</code> для консолі
<code>feof()</code>	Повертає значення true (істина), якщо досягнуто кінець файлу
<code>error()</code>	Повертає значення true, якщо виникла помилка
<code>rewind()</code>	Встановлює покажчик поточної позиції на початок файлу
<code>remove()</code>	Знищує файл
<code>fflush()</code>	Дозапис потоку у файл. Примусово записує дані з буфера у файл незалежно від ступеня заповнення буфера

Робота з текстовим файлом

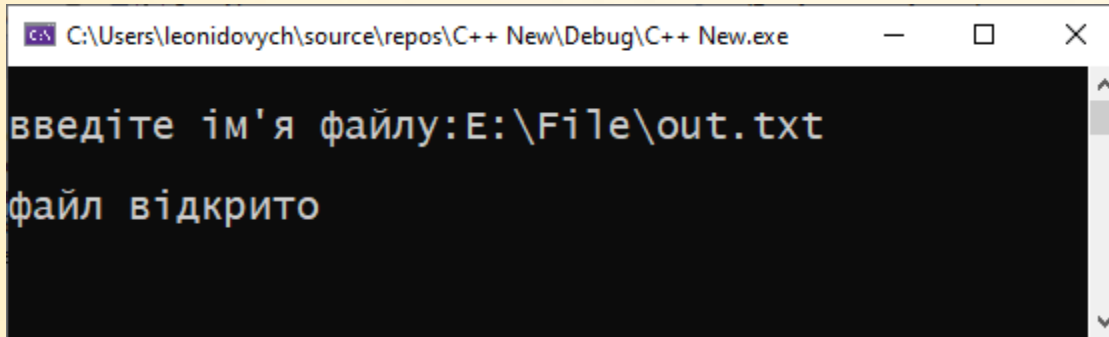
```
#include <stdio.h>
#include <Windows.h>

int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    int a, b, c;
    FILE* fout;
    fout = fopen("E:\\File\\out.txt", "w");
    printf("введіть перше число \n");
    scanf_s("%d", &a);
    printf("введіть друге число \n");
    scanf_s("%d", &b);
    c = a + b;
    fprintf(fout, "%d %d %d", a, b, c);
    fclose(fout);
}
```

Робота з текстовим файлом

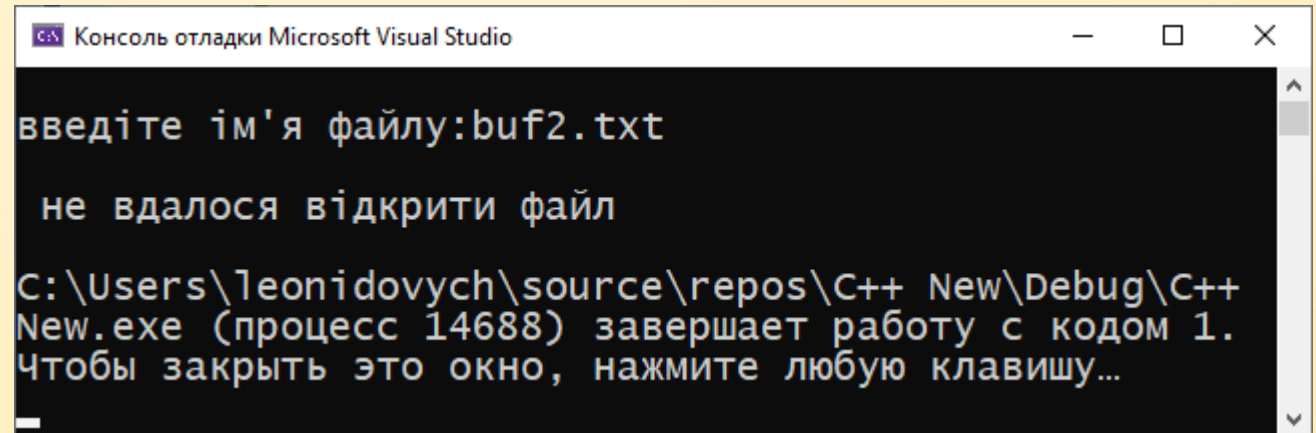
```
#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
int main() {
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    FILE* fp;
    char buf[80];
    printf("\nВведіть ім'я файлу:");
    fflush(stdin);
    gets_s(buf);
    fp = fopen(buf, "r");
    if (fp == NULL) {
        printf("\n не вдалося відкрити файл");
        getchar();
        exit(1); // Вийти з кодом завершення 1
    }
    else
        printf("\nфайл відкрито");
    getchar();}
```

Робота з текстовим файлом



C:\Users\leonidovych\source\repos\C++ New\Debug\C++ New.exe

```
введіте ім'я файлу:E:\File\out.txt  
файл відкрито
```



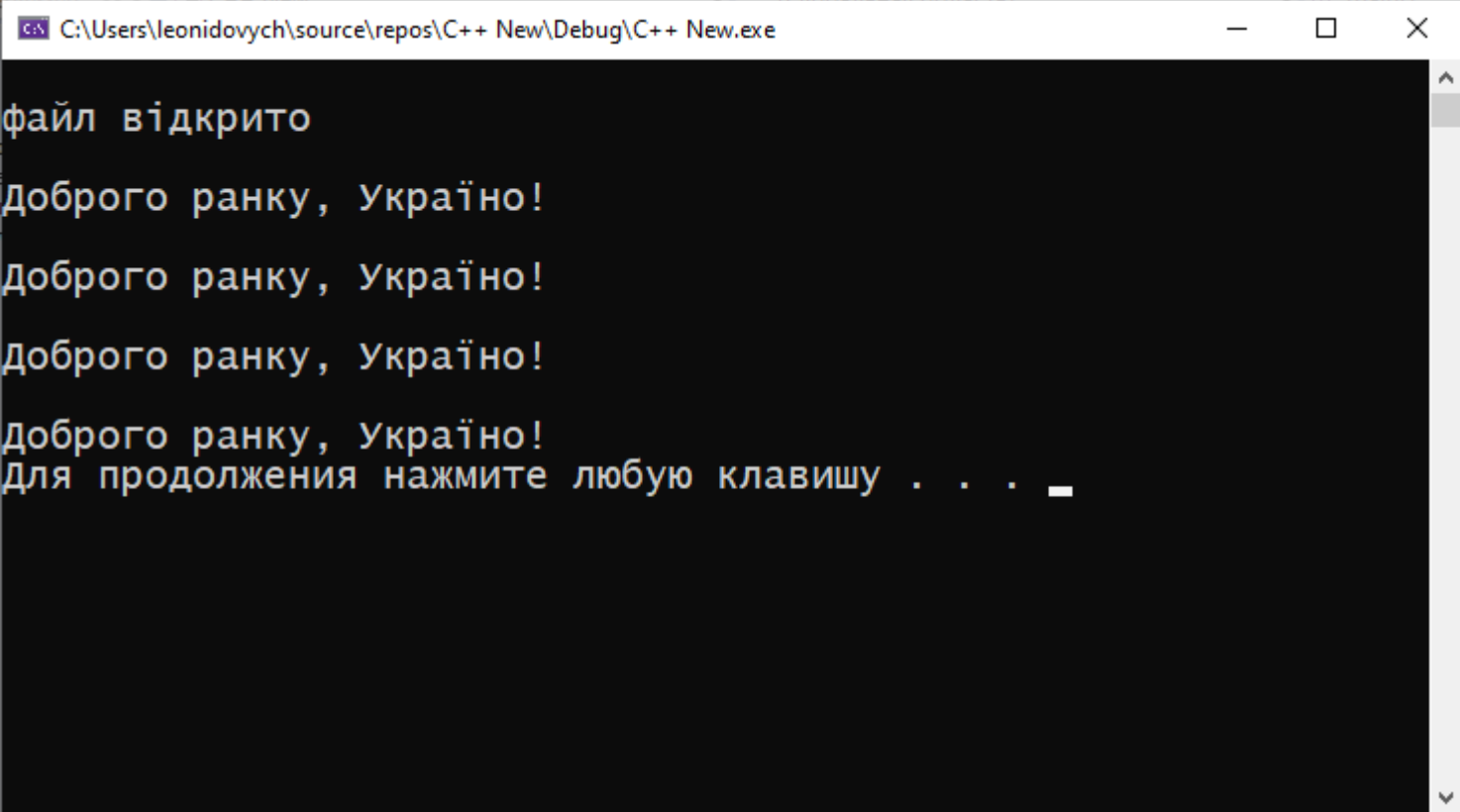
Консоль отладки Microsoft Visual Studio

```
введіте ім'я файлу:buf2.txt  
не вдалося відкрити файл  
C:\Users\leonidovych\source\repos\C++ New\Debug\C++  
New.exe (процесс 14688) завершает работу с кодом 1.  
Чтобы закрыть это окно, нажмите любую клавишу...
```

Робота з текстовим файлом

```
#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
int main() {
SetConsoleCP(1251); SetConsoleOutputCP(1251);
char buf[] = "E:\\File\\myFile.txt";
FILE* fp = fopen(buf, "a");
if (fp == NULL) {
printf("\n не вдалося відкрити файл\n"); getchar();
exit(1); // Вийти з кодом завершення 1 }
else printf("\n файл відкрито\n");
fputs("Доброго ранку, Україно!\n", fp);
fclose(fp);
fp = fopen(buf, "r");
char str[100];
while (!feof(fp))
{ fgets(str, 100, fp);
printf("\n%s", str);
}}
}
```

Робота з текстовим файлом



```
C:\Users\leonidovych\source\repos\C++ New\Debug\C++ New.exe
файл відкрито
Доброго ранку, Україно!
Доброго ранку, Україно!
Доброго ранку, Україно!
Доброго ранку, Україно!
Для продовження натисніть будь-яку клавішу . . . _
```


Робота з бінарним файлом

Створення структури

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

struct Film
{
char name[32];
char Genre[32];
char Producer[32];
double Rating;
double Price;
}*arr = NULL;

int N = 0;
```

Робота з бінарним файлом

Функція введення

```
void setData()  
{  
for (int i = 0; i < N; i++)  
{  
printf("\nНазва фільму: ");  
scanf("%s", &arr[i].name);  
printf("\nЖанр ");  
scanf("%s", &arr[i].Genre);  
printf("\nРежисер ");  
scanf("%s", &arr[i].Producer);  
printf("\nРейтинг ");  
scanf("%lf", &arr[i].Rating);  
printf("\nЦіна за диск ");  
scanf("%lf", &arr[i].Price);  
}  
}
```

Робота з бінарним файлом

Функція виведення даних на консоль

```
void showData()
{
    system("cls");
    for (int i = 0; i < N; i++)
    {
        printf("\n=====");
        printf("\n%d \nНазва фільму: %s ", i + 1, arr[i].name);
        printf("\nЖанр: %s ", arr[i].Genre);
        printf("\nРежисер: %s ", arr[i].Producer);
        printf("\nРейтин: %lf ", arr[i].Rating);
        printf("\nЦіна за диск: %lf $", arr[i].Price);
    }
    system("pause");
    system("cls");
}
```

Робота з бінарним файлом

Функція пошуку

```
void Search()
{
char Genre[32];
printf("\nВведіть жанр:");
scanf("%s", &Genre);
system("cls");
for (int i = 0; i < N; i++)
{
if (strstr(Genre, arr[i].Genre) && arr[i].Price == 100)
{
printf("\n=====");
printf("\n%d \nНазва фільму: %s ",i+1, arr[i].name);
printf("\nЖанр: %s ", arr[i].Genre);
printf("\nРежисер: %s ", arr[i].Producer);
printf("\nРейтин: %lf ", arr[i].Rating);
printf("\nЦіна за диск: %lf $", arr[i].Price);
}}
system("pause"); system("cls"); }
```

Робота з бінарним файлом

Функція запису даних в файл

```
void writeStruct(const char* path)
{
FILE* f = fopen(path, "wb");
for (int i = 0; i < N; i++)
fwrite(&arr[i], sizeof(Film), 1, f);
fclose(f);
}
```

Робота з бінарним файлом

Функція зчитування даних з файлу

```
void readStruct(const char* path)
{
    FILE* f;
    f = fopen(path, "rb");
    if (f == NULL)
        writeStruct(path);
    else
    {
        fseek(f, 0, SEEK_END);
        N = ftell(f) / sizeof(Film);
        rewind(f);
        arr = (Film*)malloc(N * sizeof(Film));
        fread(arr, sizeof(Film), N, f);
        fclose(f);
    }
}
```

Робота з бінарним файлом

Управління функціями в main

(потрібно робити у вигляді меню)

```
int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);

    readStruct("d:\\practice.bin");
    setData();
    showData();
    Search();
    writeStruct("d:\\practice.bin");

    return 0;
}
```