

Робота із бінарними файлами

(символи та числа у файлі
задаються у вигляді
послідовності байтів)

Порядок роботи із бінарними файлами
аналогічний роботі із текстовими
файлами

1. Описати файлову змінну

FILE *filename;

2. Відкрити файл

FILE *fopen(filename, mode)

Режими відкриття

- «**rb**»
- «**wb**»
- «**ab**»

Режими відкриття

- **«rb+»** – існуючий двійковий файл в режимі читання та запису;
- **«wb+»** – двійковий файл в режимі читання і запису (існуючий файл автоматично очищується)
- **«ab+»** – двійковий файл відкривається чи створюється для правки існуючої інформації, або добавки нової в кінець файлу.

Після відкриття файлу вказівник файлу вказує на 0-й байт файлу, і по мірі читання чи запису зміщується на зчитану (записану) кількість байтів.

Текуче значення вказівника файлу – номер байту, починаючи з якого буде виконуватися операція читання чи запису.

Закриття файлу

```
fclose(FILE *filename);
```

Можливості роботи із файлами

- **Remove(*filename);**
- **Rename(*oldfilename, *newfilename);**

Читання із двійкового файлу

- **fread (void *ptr, size, n, FILE *filename)**

Функція fread читає із файлу filename в масив ptr n елементів розміру size.

Функція повертає кількість прочитаних елементів. Після читання із файлу вказівник файлу зміщується на $n * size$ байтів.

Запис у двійковий файл

- **`fwrite (const void *ptr, size, n, FILE *filename);`**

Функція `fwrite` записує в файл `filename` із масиву `ptr` `n` елементів розміру `size`. Функція повертає кількість записаних елементів. Після запису інформації в файл вказівник файлу зміщується на `n*size` байтів.

Функція перевірки досягнення кінця файлу

feof(filename);

Функція sizeof(Variable) – повертає розмір
(в байтах) необхідний для збереження
змінної **Variable**

Наприклад

sizeof(int)

Приклад. Створити двійковий файл, куди записати n чисел

```
#include <iostream>
using namespace std;
int main()
{
FILE *f;
int i, n;
double a;
f=fopen("abc.dat", "wb");
```

```
cout<<"n="; cin>>n;
fwrite(&n,sizeof(int),1,f);
for(i=0;i<n;i++)
{
cout<<"a="; cin>>a;
fwrite(&a,sizeof(double),1,f);
}
fclose(f);
return 0;
}
```

Вивід на екран вмісту бінарного файлу

```
#include <iostream>
using namespace std;
int main()
{
FILE *f;
int i,n;
double *a;
f=fopen("E:\\STUDENT\\abc.dat", "rb");
fread(&n,sizeof(int),1,f);
cout<<"n="<<n<<"\n";
a=new double[n];
fread(a,sizeof(double),n,f);
for(i=0;i<n;i++)
cout<<a[i]<<"\t";
cout<<endl;
fclose(f);
return 0;
}
```

- Для довільного переміщення в середині файлу служить функція `fseek`.
- **`int fseek(FILE *F, long int offset, int origin);`**

Функція встановлює вказівник текучої позиції файлу `F`, у відповідності до початку відліку `origin` та зміщення `offset`. Параметр `offset` рівний кількості байтів, на які буде зміщено вказівник файлу відносно початку відліку, заданого параметром `origin`. В якості значення для параметра `origin` можна взяти одне із наступних значень, визначених в `stdio.h`.

- `SEEK_SET` – відлік зміщення `offset` вести від початку файлу;
- `SEEK_CUR` – відлік зміщення `offset` вести від текучої позиції файлу;
- `SEEK_END` – відлік зміщення `offset` вести від кінця файлу.

Функція повертає нульове значення при успішному виконанні операції, ненульове – при виникненні проблем при виконанні операції.

Функція `fseek` фактично дозволяє реалізувати прямий доступ до любого значення у

файлі. Необхідно тільки знати місцезнаходження (номер байту) значення у файлі.

Приклад: В заданому бінарному масиві дійкових чисел поміняти місцями найбільше та найменше значення

```
#include <iostream>
using namespace std;
int main()
{
FILE *f;
```

```
int i,n,imax, imin;
double *a, max,min;
f=fopen("abc.dat", "rb+");
fread(&n,sizeof(int),1,f);
cout<<"n="<<n<<"\n";
a=new double[n];
fread(a,sizeof(double),n,f);
for(imax=imin=0, max=min=a[0],i=1;i<n;i++)
{
if (a[i]>max)
{
max=a[i];
imax=i;
}
if (a[i]<min)
{
min=a[i];
imin=i;
}
}
```

//Переміщує вказівник до максимального
елементу.

```
fseek(f,sizeof(int)+imax*sizeof(double),SEEK_SET);
```

//Запис min замість максимального елементу
файлу.

```
fwrite(&min,sizeof(double),1,f);
```

//Переміщення вказівника до мінімального
елементу.

```
fseek(f,sizeof(int)+imin*sizeof(double),SEEK_SET);
```

//Запис max замість мінімального елементу
файла.

```
fwrite(&max,sizeof(double),1,f);
```

```
fclose(f);
```

//Видалення динамічного масиву a.

```
delete []a;
```

```
return 0;
```