

Тема 3

Двоични файлове. Big endian. Запозване на обекти в двоичен файл.

Двоичен файл е компютърен файл, който не е текстов. Двоичните файлове са трудни за работа от хора, понеже са трудно четими, но компютрият лесно работи с тях. Те съдържат байтове, които са предназначени да бъдат интерпретирани като нещо различно от текстови знаци. Байтовете се записват в обратна посока. Най-старият байт е последен.

Пример:

Ако искаме да запазим числото 155 555 в двоичен файл, тъй като шестнадесетичният запис на числото е 00 02 5f a3, то съвъртването на файла ще е a3 5f 02 00.

Потокът към двоичния файл се отваря по следния начин:

поток (име на файл, std::ios::binary);

В двоичните форми не можем да използваме:

>, <, cin.get(), cin.getline()

Вместо това, използваме следните функции:

- read((char*) memoryBlock, size_t size) - за четене от файл

- write((const char*) memoryBlock, size_t size) - за писане във файл

Нека обърнем внимание, че функциите приемат char*/const char*. Ако искаме да запазим/прочетем променлива от друг вид, ще трябва експлицитно да преобразуваме указателя към тип char*/const char*.

Защо точно char*? Защо не bool* или int*?

Понеже четем и записваме байт по байт, всеки път указателят се инкрементира с 1 byte. Ако беше int* инкрементацията щеше да е 4 bytes.

Пример за писане в двоичен файл:

```
ofstream file("file.dat", std::ios::binary);
```

```
int a = 155555;
```

```
file.write((const char*)&a, sizeof(a));
```

Пример за четене от двоичен файл

```
ifstream file("file.dat", std::ios::binary);
```

```
int a;
```

```
file.read((char*)&a, sizeof(a));
```


Можем да запазваме и обекти, които не използват динамична памет.

Пример:

```
struct Test          int main()
{                    {
    char ch;          ofstream file("output.dat", ios::binary);
    int a;            if (!file.is_open())
{                      {
                    cout << "Error" << endl;
                    return -1;
                    }
                    Test t = {'a', 400};
                    file.write((const char*)&t, sizeof(t));
                    file.close();
                    }
}
```

Ако искаме да запазим обект, който използва динамична памет, трябва да запазваме всяка елемент-данна координатно. При запазване на елемент-данна, използваща динамична памет, трябва първо да запазим големината ѝ и след това самата елемент-данна.