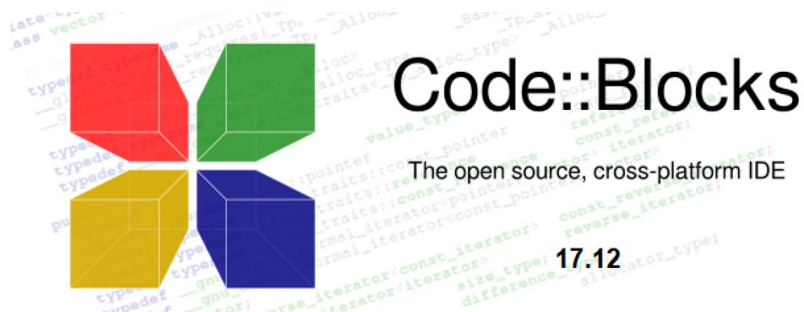


ITIS-LS “Francesco Giordani” Caserta

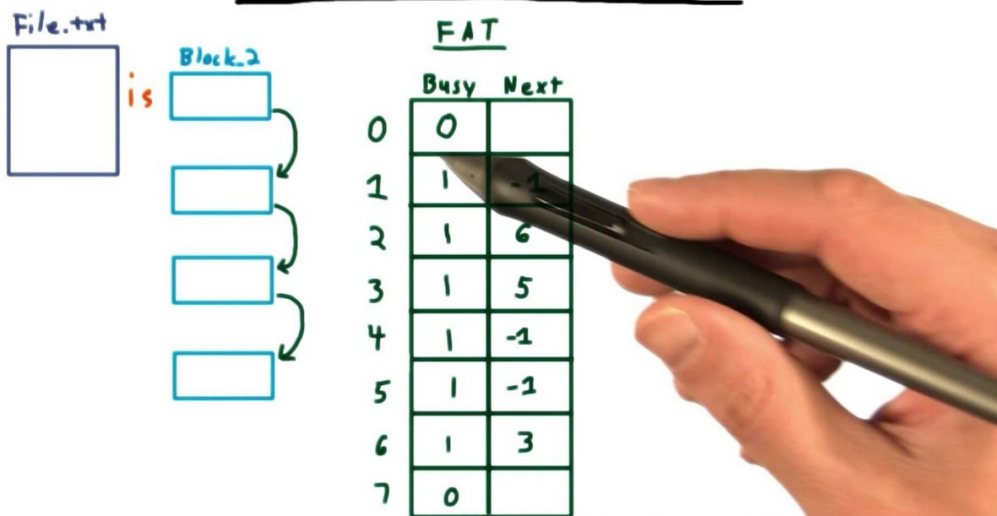
prof. Ennio Ranucci

a.s. 2020-2021

*Esercitazioni facile facile FILE C++*



File Allocation Table



*/\**

*ITIS-LS F.Giordani Caserta*

*Anno scolastico 2020/2021*

*Classe 4^ sez.C spec. Informatica*

*Data: 01/02/2021*

*Numero es: 1*

*Versione:1.0*

*Programmatore/i: Lezione collettiva*

*Sistema Operativo:Windows 10*

*Compilatore/Interprete: Code::Blocks Release 17.12 rev 11256*

*Obiettivo didattico: L'alunno e' in grado di scrivere e leggere in un file;*

*Obiettivo del programma: Caricare e leggere un numero intero in un file binario;*

*\*/*

*#include <iostream>*

*#include<fstream>*

*using namespace std;*

*fstream numeriFile;*

*int main()*

*{*

***//scrittura di un numero intero nel file***

*int num1, num2;*

*num1=124;*

*numeriFile.open ("numeri.dat", ios::out | ios::binary);*

*numeriFile.write((char \*) &num1, sizeof(num1));*

*numeriFile.close();*

***//lettura del numero intero dal file***

*numeriFile.open ("numeri.dat", ios::in | ios::binary);*

*numeriFile.read((char \*) &num2, sizeof(num2));*

*numeriFile.close();*

*cout<<num2;*

*return 0;*

*}*

***//124 corrisponde al carattere pipe "|". Aprire numeri.dat con blocco note per osservare che il numero binario 124 e' stato interpretato come carattere ascii***

```

/*
ITIS-LS F.Giordani Caserta
Anno scolastico 2020/2021
Classe 4^ sez.C spec. Informatica
Data: 02/02/2021
Numero es: 2
Versione:1.0
Programmatore/i: Lezione collettiva
Sistema Operativo:Windows 10
Compilatore/Interprete: Code::Blocks Release 17.12 rev 11256
Obiettivo didattico: L'alunno e' in grado di scrivere e leggere in un file di testo;
Obiettivo del programma: Caricare e leggere un carattere in un file di testo;
*/

#include <iostream>
#include <fstream>
using namespace std;

int main ()
{
    char car1, car2;
    car1='a';

    // scrittura nel file di testo
    ofstream out("caratteriFile.txt");
    out<<car1;
    out.close();

    // lettura dal file di testo
    ifstream in("caratteriFile.txt");
    in>>car2;
    in.close();
    cout<<car2;
    return 0;
}

```

```
/*  
ITIS-LS F.Giordani Caserta  
Anno scolastico 2020/2021  
Classe 4^ sez.C spec. Informatica  
Data: 02/02/2021  
Numero es: 3  
Versione:1.0  
Programmatore/i: Lezione collettiva  
Sistema Operativo:Windows 10  
Compilatore/Interprete: Code::Blocks Release 17.12 rev 11256  
Obiettivo didattico: L'alunno e' in grado di scrivere,leggere ed aggiungere in un file;  
Obiettivo del programma: Caricare e leggere deu numeri interi in un file di binario;  
*/
```

VERSIONE CON UNA SOLA APERTURA E DUE SCRITTURE CONSECUTIVE

```
#include <iostream>  
#include<fstream>  
using namespace std;  
  
fstream numeriFile;  

```

## VERSIONE CON DUE APERTURE E DUE SCRITTURE

```
#include <iostream>
#include<fstream>
using namespace std;

fstream numeriFile;

int main()
{
    //due aperture e due scritture nel file

    int num, num2;
    num=124;
    numeriFile.open ("numeri.dat", ios::out | ios::binary);
    numeriFile.write((char *) &num, sizeof(num));
    numeriFile.close();
    numeriFile.open ("numeri.dat", ios::out | ios::binary);
    num=150;
    numeriFile.write((char *) &num, sizeof(num));
    numeriFile.close();

    //due letture dal file

    numeriFile.open ("numeri.dat", ios::in | ios::binary);
    numeriFile.read((char *) &num, sizeof(num));
    cout<<num;
    cout<<endl;
    numeriFile.read((char *) &num2, sizeof(num2));
    cout<<num2;
    numeriFile.close();
    return 0;
}
```

## VERSIONE CON DUE APERTURE, UNA SCRITTURA E UN APPEND

```
#include <iostream>
#include<fstream>
using namespace std;

fstream numeriFile;

int main()
{
    //due aperture e due scritture nel file

    int num, num2;

    num=124;
    numeriFile.open ("numeri.dat", ios::out | ios::binary);
    numeriFile.write((char *) &num, sizeof(num));
    numeriFile.close();
    numeriFile.open ("numeri.dat", ios::app | ios::binary);
    num=150;
    numeriFile.write((char *) &num, sizeof(num));
    numeriFile.close();

    //due letture dal file

    numeriFile.open ("numeri.dat", ios::in | ios::binary);
    numeriFile.read((char *) &num, sizeof(num));
    cout<<num;
    cout<<endl;
    numeriFile.read((char *) &num2, sizeof(num2));
    cout<<num2;
    numeriFile.close();

    return 0;
}
```