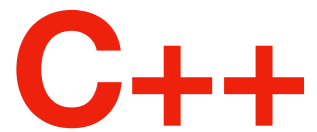


Informatica

C++

Prof. Bortolai Gariele



Stream da file

Supponiamo che si voglia aprire e leggere il contenuto di un file, con il quale si vorrà successivamente eseguire alcune operazioni.

Un esempio potrebbe essere un elenco di nomi che si vuole riordinare in ordine alfabetico.

Vediamo adesso la sintassi per aprire e leggere il contenuto di un file di questo tipo:

```
0 █  
1  
2
```

C++

Stream

Il file contiene tre numeri incolonnati, il programma é molto più articolato rispetto a ciò che abbiamo visto fino ad ora.

Innanzitutto dobbiamo usare una nuova libreria ovvero `fstream` essa ci permette di operare sui file, infatti viene inclusa all'inizio.

```
#include <iostream>
#include <fstream>

using namespace std;

int main(){

    int n;
    double d;

    ifstream ifile;

    ifile.open("dati1.dat");

    double* x=new double [n];

    n=0;
    while(ifile>>d){

        n++;

    }

    ifile.clear();
    ifile.seekg(ios::beg);

    for(int i=0;i<n;i++){

        ifile>>x[i];

        cout<<x[i]<<endl;

    }

    ifile.close();

    return 0;

}
```

Definizione dello stream, nel nostro caso di input

Cerco e apro il file che devo leggere

Puntatore

Conto il numero di oggetti contenuti nel file

Ripristino il file nella situazione iniziale

Riempo il puntatore

C++

Stream

Procediamo nell'analisi del programma, nel primo fumetto ho impostato lo stream del file in input utilizzando ifstream dandogli il nome ifile, nel secondo fumetto ad ifile di aprire il file dati1.dat.

Successivamente ho definito un puntatore, un puntatore é una array ma OP (over power) perché é capace di variare la sua dimensione in base alle nostre necessità, nella vignetta successiva ho controllato quanto fosse grande il file, per farlo ho usato un ciclo while al cui interno é contenuto un numero n che di volta in volta sale finché non sono arrivato infondo al file, n sarà la dimensione del nostro puntatore.

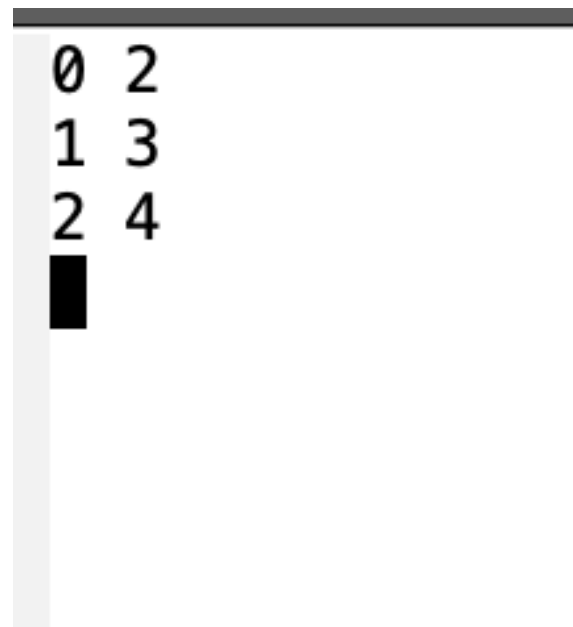
Infine ho ripristinato il file alla condizione iniziale e ho riempito il puntatore e stampate le sue componenti per vedere se funzionasse.

C++

Stream

Sorge spontanea una domanda, ma se il file fosse fatto da più colonne?

Molto semplicemente si generalizza il procedimento precedente, basta definire tanti puntatore quante le colonne.



0	2
1	3
2	4

```
#include <iostream>
#include <fstream>
```

```
using namespace std;
```

```
int main(){
```

```
    int n;
    double d,k;
```

```
    ifstream ifile;
```

```
    ifile.open("dati1.dat");
```

```
    double* x=new double [n];
    double* y=new double [n];
```

```
    n=0;
```

```
    while(ifile>>d>>k){
```

```
        n++;
```

```
    }
```

```
    ifile.clear();
```

```
    ifile.seekg(ios::beg);
```

```
    for(int i=0;i<n;i++){
```

```
        ifile>>x[i]>>y[i];
```

```
        cout<<x[i]<<" "<<y[i]<<endl;
    }
```

```
    ifile.close();
```

```
    return 0;
```

```
}
```

Due colonne quindi due puntatori

Due colonne quindi qui metto due variabile

Infine riempio i due puntatori

C++

Stream

Ovviamente si può anche scrivere su un file, questa operazione é ancora più semplice della precedente.

Vediamo un esempio.

In questo programma si inizializza un vettore e dopo vengono scritte le sue componenti su un file.

In questo caso lo stream viene inizializzato in output usando ofstream chiamandolo ofile, con file.open apro o creo il file nel caso in cui non esistesse.

```
#include <iostream>
#include <fstream>

using namespace std;

int main(){

    const int n=3;

    double x[n]={1,2,3};

    ofstream ofile;

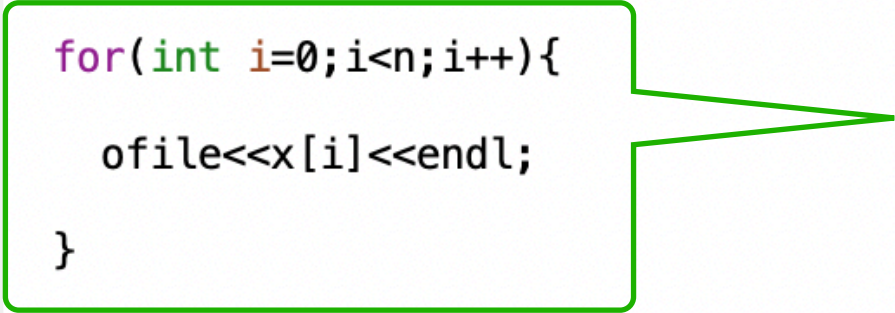
    ofile.open("risultati.dat");

    for(int i=0;i<n;i++){
        ofile<<x[i]<<endl;
    }

    ofile.close();

    return 0;

}
```



Scrivo sul file le componenti del vettore

C++

Stream

Nel caso in cui vogliate scrivere il file su più colonne basta aggiungere una componente ad ofile nel come nell'immagine a destra.

Il risulta sarà un file il cui contenuto é nella slide successiva.

```
#include <iostream>
#include <fstream>

using namespace std;

int main(){

    const int n=3;

    double x[n]={1,2,3};
    double y[n]={3,6,9};

    ofstream ofile;

    ofile.open("risultati.dat");

    for(int i=0;i<n;i++){

        ofile<<x[i]<<y[i]<<endl;

    }

    ofile.close();

    return 0;

}
```

1	3
---	---

2	6
---	---

3	9
---	---

C++

Stream

Chi volesse può provare a scrivere un programma che:

- Apra il file dati.dat composto da una sola colonna.
- Calcoli la somma di tutti i dati al suo interno.
- Scriva il risultato su un file chiamato risultati.dat