35. La gestione dei file in C++

Corso di Informatica

Corso di Laurea in Matematica (D.M. 270/04) - A.A. 2020/2021
Angelo Cardellicchio
angelo.cardellicchio@uniba.it
11/01/2021

Outline

- Le classi per la gestione dei file
- Esempio 1: copia riga per riga
- Esempio 2: copia carattere per carattere
- Un (rapido) confronto con la gestione in C

Le classi per la gestione dei file

- Il C++ permette la gestione dei file usando una serie di classi derivate dalla classe base **`ios`**
  - *Questo permette di avere un'interfaccia comune, un po' come avviene nel C, ma in maniera semplificata*
- Le classi maggiormente utilizzate sono **`ofstream`** (per operazioni in sola scrittura), **`ifstream`** (per operazioni in sola lettura), ed **`fstream`** (per entrambi i tipi di operazioni)
- Vediamo di seguito un paio di possibili applicazioni

Esempio 1: copia riga per riga (1)

- Iniziamo includendo gli header necessari, ovvero **`fstream`** ed **`iostream`**, oltre che **`string`**
- Dichiariamo all'interno del **`main`** un oggetto di tipo **`ifstream`**, che chiameremo **`input`**, e che avrà il compito di leggere il file **`input.txt`**

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

using namespace std;

int main() {
    ifstream input("input.txt")
    // ...
}
```

Esempio 1: copia riga per riga (2)

- Verifichiamo l'esistenza del file mediante il metodo **`is_open()`** associato alla variabile **`input`**
  - *Utilizzeremo questo metodo ogni qual volta dovremo verificare che un file sia effettivamente aperto*
  - *Ciò vale quindi anche per i file aperti in sola scrittura, infatti lo usiamo anche in seguito*
- Apriamo quindi un file in scrittura usando un'istanza della classe **`ofstream`**, che chiameremo **`output`**
  - *Il file* **`output.txt`** *di default sarà creato se non esiste, o sovrascritto se esiste*

```
int main() {
    // ...
    if (input.is_open()) {
        // ...
        ofstream output("output.txt");
        if (output.is_open()) {
            // ...
        }
    }
    // ...
}
```

Esempio 1: copia riga per riga (3)

- Per accedere al file riga per riga, useremo il metodo **`getline()`**
- Questo metodo ci permette di 'aprire' lo stream associato all'oggetto input, che gli passeremo come primo parametro, ed associare la riga attualmente letta ad un oggetto di tipo **`string`**, passato come secondo parametro
- Nel ciclo, controlliamo che una riga stia venendo effettivamente letta, e che lo stream di output sia tutt'ora valido mediante il metodo **`good()`**

```
int main() {
    // ...
    if (output.is_open()) {
        string line;
        while (getline(input, line)
            && output.good()) {
            output << line << endl;
        }
        output.close();
    }
    // ...
}
```

Esempio 2: copia carattere per carattere (1)

- Le parti di accesso e verifica dei file di questo esempio sono esattamente le stesse usate per l'esempio precedente
- Quello che cambia sta nel metodo con cui viene copiato il file: questa volta, infatti, usiamo la funzione **`get()`** per ottenere il carattere attuale da input, che viene inserito quindi in output mediante la funzione **`put()`**
  - *Quest'ultima ci permette di valutare se l'inserimento è andato o meno a buon fine*

```
if (output.is_open()) {
    char c = NULL;
        while (input.get(c)
            && output.good()) {
            if (output.put(c)) {
                cout << "Ok";
            }
            else {
                exit(EXIT_FAILURE);
            }
        }
        output.close();
    }
}
```

Un (rapido) confronto con la gestione in C

Operazione	C	C++
Apertura di un file	Richiede l'uso di un puntatore a `FILE`	Richiede l'uso di un oggetto di classe `ifstream`, `ofstream` o `fstream`
Verifica dell'apertura	E' necessario verificare che il puntatore a `FILE` non sia `null`	Si utilizza il metodo `is_open()`
Scrittura su file	Viene gestita mediante la funzione `fprintf`	Viene gestita come tutte le operazioni di output in C++
Lettura da file	Viene gestita mediante la funzione `fscanf`	Viene gestita come tutte le operazioni di input in C++

Domande?

42