

Librerie e Compilazione separata

Programmazione – Canale M-Z

LT in Informatica
10 Aprile 2017



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

- La maggior parte dei programmi che abbiamo visto finora erano composti da un unico file `.cpp`
- Nel caso di programmi più grandi è conveniente **suddividere il codice in più file**:
 - migliore suddivisione e organizzazione del codice
 - facilità di manutenzione e correzione degli errori
 - possibilità di riutilizzare il codice per progetti diversi
- Il linguaggio C++ ci fornisce **due strade** per suddividere il codice:
 - 1 inclusione diretta di file sorgente e **compilazione unica**;
 - 2 inclusione di header file e **compilazione separata**

- Effettuata dal preprocessore mediante la direttiva

```
#include "mylib.cpp"
```

- Prima della compilazione il preprocessore inserisce il codice contenuto in `mylib.cpp` nel programma
- La compilazione si fa normalmente richiamando il compilatore sul file contenente il `main`:

```
g++ main.cpp -o main
```

- È il metodo più semplice ma meno efficiente:
 - la modifica di uno solo dei file richiede la ricompilazione completa di tutto il programma.

```
// mylib.cpp

void aggiungi(int &somma, int n)
{
    somma += n;
}
```

```
// main.c
#include <iostream>
#include "mylib.cpp"

using namespace std;

int main() {
    int i, n, somma = 0;

    for(i=1; i <= 3; i++) {
        cin >> n;
        aggiungi(somma, n);
    }
    cout << "TOTALE:" << somma << endl;
}
```

- Anziché tutto il sorgente si include un **header file**:
 - è un file con estensione `.h`
 - contiene solamente:
 - inclusioni di altri header file
 - definizione di `namespace`
 - definizione di strutture e di tipi (`typedef`)
 - prototipi di funzioni
 - si usa la direttiva `#include "mylib.h"`
- Il codice delle funzioni va scritto in un file `.cpp` separato:
 - che deve includere l'header!
- Compilazione e linking devono essere fatti separatamente:

```
g++ -c mylib.cpp
g++ -c main.cpp
g++ -o main main.o mylib.o
```

```
// mylib.h

namespace mylib {

    void aggiungi(int &somma, int n);

}
```

```
// mylib.cpp
#include "mylib.h"

namespace mylib {

    void aggiungi(int &somma, int n)
    {
        somma += n;
    }

}
```

```
// main.cpp
#include <stdio.h>
#include "mylib.h"

using namespace std;

int main() {
    int i, n, somma = 0;

    for(i=1; i <= 3; i++) {
        cin >> n;
        mylib::aggiungi(somma, n);
    }
    cout << "TOTALE:" << somma << endl;
}
```

- 1 Nell'header file `.h` vanno inseriti:
 - le definizioni dei tipi (`struct`, `typedef ...`);
 - i prototipi delle funzioni (senza il codice);
- 2 Il codice delle funzioni viene scritto nel file `.cpp` della libreria
 - che deve includere l'header file
- 3 Il programma include solamente l'header file
 - `#include "mylib.h"`
- 4 I file `.cpp` vanno compilati separatamente
 - `g++ -c mylib.cpp ; gcc -c main.cpp`
- 5 Per ottenere l'eseguibile si deve fare il linking
 - `g++ -o main main.o mylib.o`