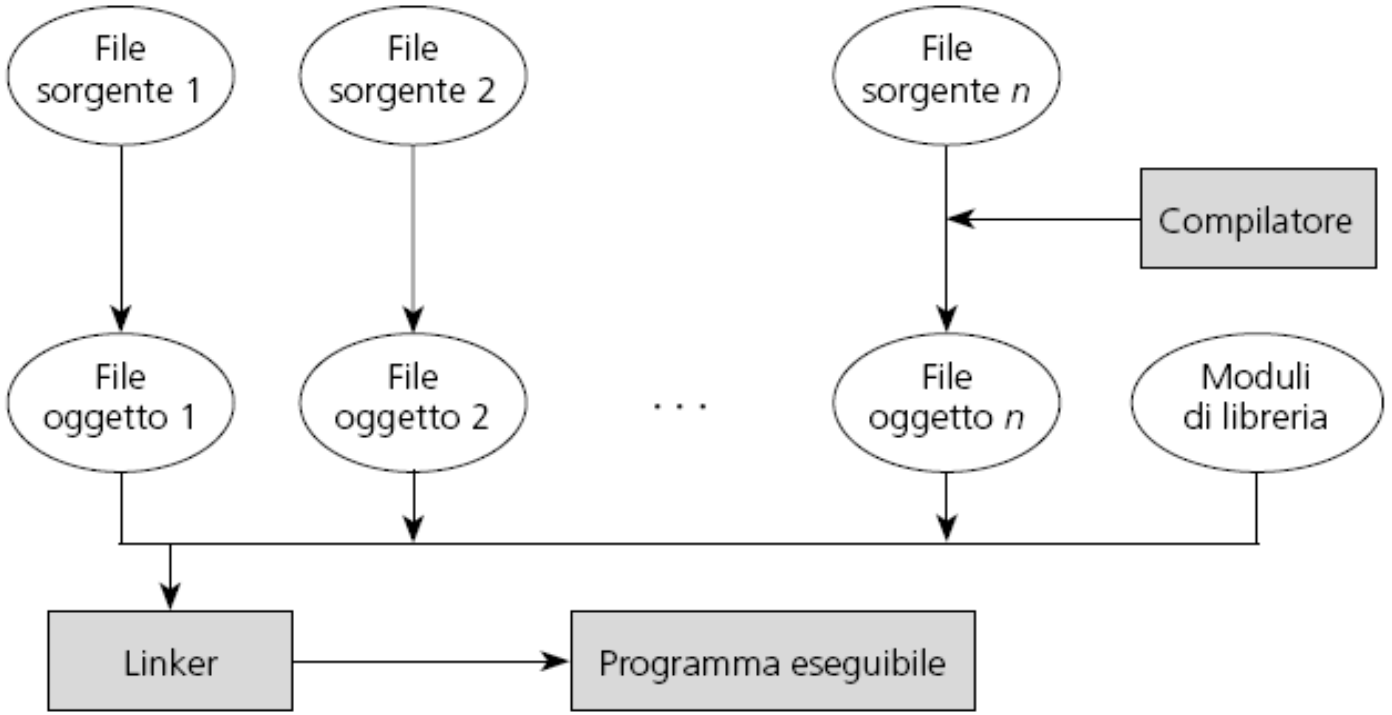
Programmazione a oggetti

Programmazione modulare

A.A. 2020/2021
Francesco Fontanella

Compilazione Modulare

- i programmi grandi sono più facili da gestire se divisi in vari file sorgente, chiamati anche moduli
- Questo meccanismo consente una migliore strutturazione dei programmi

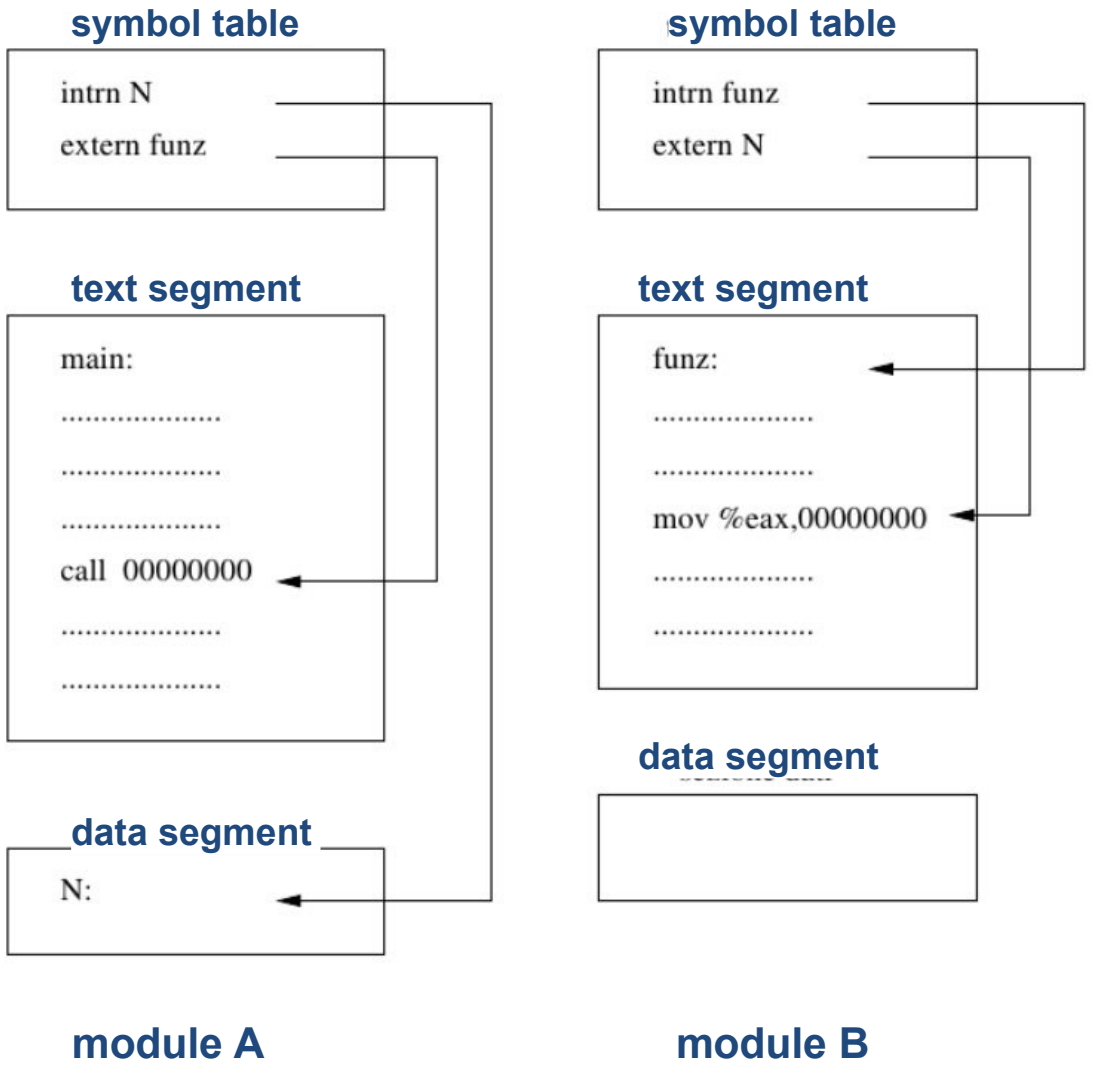


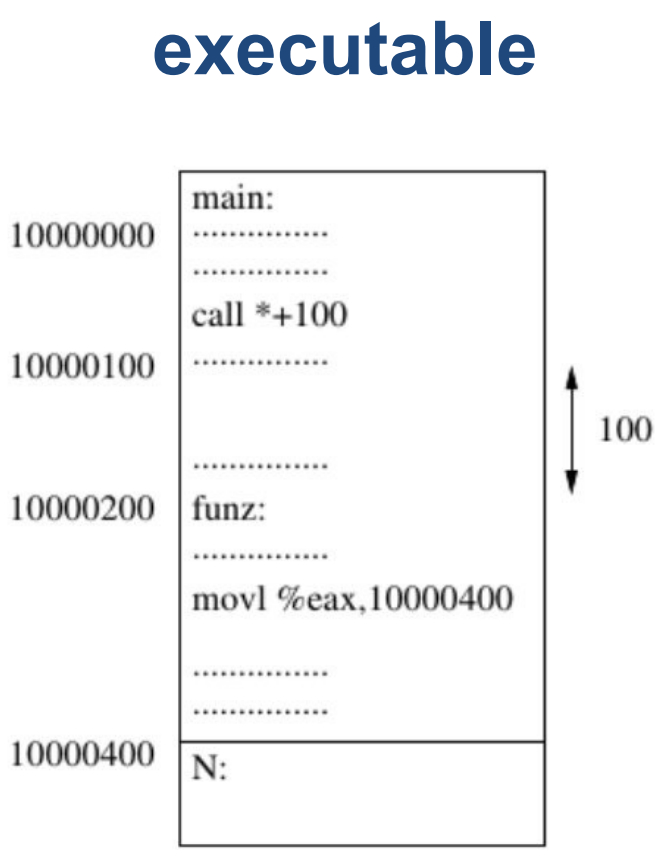
Il linker

- Il linker è un programma che genera un eseguibile integrando file oggetto che sono stati compilati separatamente.
- I file oggetto (generati dal compilatore) sono parti di programma che contengono:
  - codice macchina
  - Informazioni utili al linker, sottoforma di tabelle (di simboli).

- Ulteriori informazioni sono disponibili al questo link:

https://it.wikipedia.org/wiki/Linking


symbol table
intrn N
extern funz
text segment
main:
.....
.....
.....
call 00000000
.....
.....
data segment
N:
module A
symbol table
intrn funz
extern N
text segment
funz:
.....
.....
mov %eax,00000000
.....
.....
data segment
module B


executable
10000000
main:
.....
.....
call *+100
10000100
.....
.....
10000200
funz:
.....
movl %eax,10000400
.....
.....
10000400
N:
100

Modularizzazione

- L'uso disciplinato di alcuni meccanismi del linguaggio C++ consente una corretta strutturazione di un programma in moduli.

- Tra i principali meccanismi vi sono:
  - la compilazione separata,
  - l'inclusione testuale,
  - le dichiarazioni **`extern`**.
  - l'uso dei prototipi di funzioni.

Specifica e Implementazione

- E’ buona norma tenere separata la specifica di un modulo dalla sua implementazione:
  - un programma utente di un modulo A deve conoscerne solo la specifica, ma non i dettagli della sua implementazione.
- A tal fine si scrive un file di intestazione o header file (estensione .h) contenente le dichiarazioni che costituiscono l’interfaccia di A, ed un file separato per l’implementazione di A.

main. cpp

```
// Utilizzatore del modulo A
#include “A.h”
```

A.h

```
// Interfaccia di A
```

A.cpp

```
// Implementazione del modulo A
#include “A.h”
```

- fintanto che l’interfaccia resta inalterata, l'implementazione può essere modificata senza dover ricompilare il modulo utente (ma naturalmente occorre ricollegare i moduli oggetto).

- La specifica, contenuta nel file di intestazione, può essere riguardata come una sorta di contratto sottoscritto tra l'implementatore e l'utente del modulo in questione

- Più programmatori che lavorano ad un stesso progetto una volta accordatisi sulla specifica dei vari moduli, possono procedere in maniera indipendente all'implementazione dei rispettivi moduli.

Specifica ed Implementazione: Esempio

- Programma per la gestione di una segreteria studenti:
  - Ci sarà bisogno di un elenco per gestire le informazioni anagrafiche degli studenti: nome, cognome, #matricola, reddito, ecc.
- Il programma deve:
  - Leggere prendere in input un certo numero di studenti N
  - Fornire in uscita i valori del massimo e del minimo reddito
  - Visualizzare l'elenco di tutti gli studenti
  - Visualizzare l'elenco di tutti gli studenti al di sotto di un certo reddito

Una possibile struttura del programma è la seguente:

main.c

```
// Utilizzatore del modulo studenti
#include “studente.h”
```

studente.h

```
// INTERFACCIA del modulo
// studente
  const int  MAX_STRING = 100;
  const int  MAX_STUDENTS = 100;

  struct Studente {
        char nome[MAX_STRING],
             cognome[MAX_STRING];
        int matr, red;
        void input();
        void output();

  void inputStudenti(int &n, Studente s[]);
  void outputStudenti(int n, Studente s[]);
  int ric_max_red(int n, Studente s[]);
   void aggiungiStudente(int &n,
        Studente v[], Studente s);
```

studente.c

```
// IMPLEMENTAZIONE del modulo
// studenti
#include “studente.h”
```

Studente.cpp

```
#include “studente.h”

void inputStudenti(int &n, Studente s[])
{
   int i;  // indice di scorrimento del vettore

   cout<<"Quanti elementi vuoi inserire? ";
   cin>>n;

   //assegnaz. di valore agli elementi dell'array
   cout<<"Inserisci "<<n<<" studenti"<<endl;

   for (i=0;i<n;i++)
   s[i].input();
 }

void outputStudenti(int n, Studente s[])
{
   int i;  // indice di scorrimento del vettore

   cout<<"Elenco studenti";

   for (i=0;i<n;i++) {
      cout<<"studente # "<<i + 1<<endl;
      s[i].output();
   }
}
```

```
void Studente::input() {
  cout<<"inserire cognome: ";
  cin>>cognome;
  cout<<"inserire nome: ";
  cin>>nome;
  cout<<"inserire matricola: ";
  cin>>matr;
  cout<<"inserire il reddito: ";
  cin>>red;
}

void Studente::output() {
  cout<<endl<<"cognome: "<<cognome;
  cout<<endl<<"nome: "<<nome;
  cout<<endl<<"matricola: "<<matr;
  cout<<endl<<"reddito: "<<red;
  cout<<endl;
}
```

Variabili extern

- All'interno di un modulo è possibile anche usare variabili che sono definite all'interno di altri moduli.
- Per fare ciò è necessario dichiararla usando la parola riservata **`extern`**;
- In questo modo si indica al compilatore che la variabile è definita in un altro modulo che sarà poi linkato assieme.

A.cpp

```
// DICHIARAZIONE DI N
// come variabile esterna
extern int N;
```

B.cpp

```
// DEFINIZIONE DI N

int N;
```

NOTA

La variabile **`N`** definita nel modulo B.cpp è accessibile se e solo se NON è stata definita come variabile **`static`**.

Librerie software

- Le tecniche di sviluppo modulare consentono lo sviluppo di librerie di moduli software, comprese le librerie del linguaggio.
- Il produttore di una libreria distribuisce:
  - Un documento contenente le specifiche della libreria
  - i file di intestazione (che devono essere inclusi dall'utilizzatore nel codice sorgente) dei moduli che fanno parte della libreria;
  - i moduli di libreria in formato oggetto (già compilati), che l'utilizzatore deve collegare assieme ai propri moduli oggetto.

Application programming interface (API)

- È un interfaccia tra una libreria software (il server) e gli utenti (client)
- È l'insieme di procedure e protocolli che consentono l'accesso ai “servizi” forniti da una determinata libreria
- La documentazione di una API è costituita dalla descrizione (specifica e semantica) dell'insieme di procedure, variabili e strutture dati che la compongono.
- Esempi: windows, google maps, facebook