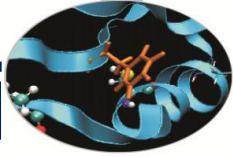


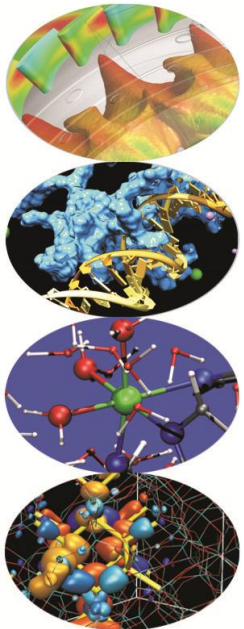
Compilatori e Livelli di Compilazione

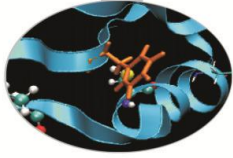


Introduction to Fortran 90

Elda Rossi, *CINECA*

Marzo 2015





compilatore

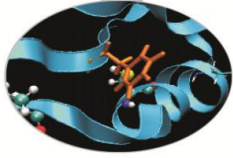
- Un **compilatore** è un programma che traduce una serie di istruzioni scritte in un determinato linguaggio di programmazione (codice sorgente) in istruzioni di un altro linguaggio (codice oggetto).
- nel 1957 il team Fortran presso l'IBM, guidato da John Backus, fu accreditato come primo inventore di un compilatore completo.



linker

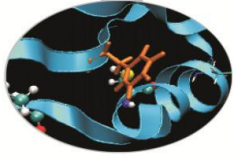
- Il **linking** è il procedimento di integrazione dei vari moduli a cui un programma fa riferimento (i quali possono essere sottoprogrammi o librerie), per creare una singola unità eseguibile.
- Il *linker* (o *link editor* o *loader*) è un programma che effettua il collegamento tra il programma oggetto, cioè la traduzione del codice sorgente in linguaggio macchina, e le librerie del linguaggio necessarie per l'esecuzione del programma.





Il compilatore

- Traduce il codice sorgente in codice macchina
- Rifiuta codici sintatticamente errati
- Segnala (alcuni) potenziali problemi semantici
- Può tentare di “ottimizzare” il codice
 - ottimizzazioni indipendenti dal linguaggio
 - ottimizzazioni dipendenti dal linguaggio
 - ottimizzazioni dipendenti dalla CPU
 - ottimizzazioni dell’uso della memoria e della cache
 - suggerimenti al processore su cosa probabilmente farà il codice



Il compilatore

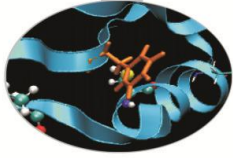
Il programmatore ha a disposizione un'ampia scelta di compilatori sviluppati per diversi sistemi operativi quali Linux/Unix, Windows, Macintosh.

Oltre ai compilatori gratuiti derivanti dal progetto GNU, esistono diversi compilatori commerciali.

I più conosciuti sono probabilmente quelli prodotti dall'Intel e dalla Portland.

Una certa rilevanza di mercato hanno anche i compilatori commercializzati dalla Absoft, dalla HP e da IBM.

CINECA: FERMI (xlf), GALILEO (Ifort)



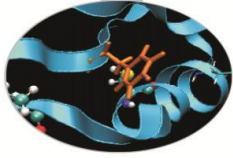
Il compilatore

Esegue trasformazioni del codice come:

- Dead and redundant code removal
- Common subexpression elimination (CSE)
- Strength reduction
- Inlining
- Index reordering
- Loop pipelining/unrolling, merging
- Cache blocking

Lo scopo è massimizzare le prestazioni

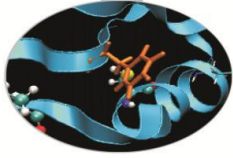
Opzioni di Ottimizzazione



I compilatori prevedono numerose **opzioni di ottimizzazione** che vengono **specificate in fase di compilazione**, unitamente alle altre. Esistono livelli di ottimizzazione **generali** e **specifici**, legati all'hardware sottostante.

Alcune opzioni possono risultare onerose dal punto di vista del tempo richiesto per la compilazione. Ottimizzazioni spinte possono non rispettare la semantica del programma. Ottimizzazioni elevate modificano la sequenza delle istruzioni e possono creare difficoltà nella fase di debugging.

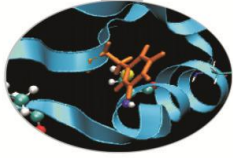
Per questo motivo, è consigliabile utilizzare livelli bassi di ottimizzazione quando si è ancora in fase di sviluppo del codice.



Opzioni generiche

Le opzioni generiche non sono legate ad un particolare hardware ma permettono comunque di ottenere eseguibili molto performanti.

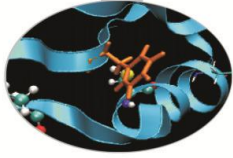
La più comune è indicata dalla sintassi **-On** dove n può assumere i valori interi compresi fra 0 e 4, dove ogni valore è cumulativo, ovvero include le caratteristiche dei livelli precedenti, migliorandone le funzionalità, anche se la compilazione risulterà più lenta.



Livelli di ottimizzazione

Un compilatore presenta livelli incrementali di ottimizzazione, ad esempio:

- O0: nessuna ottimizzazione, il codice è tradotto letteralmente
- O1, -O2: ottimizzazioni locali, compromesso tra la velocità di compilazione, ottimizzazione e dimensioni dell'eseguibile (livello di default)
- O3: ottimizzazioni memory-intensive, può alterare la semantica del programma
- O4: ottimizzazioni aggressive

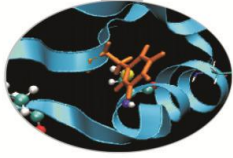


Scelta delle opzioni

- Le opzioni sono dettagliatamente descritte nella User's Guide del compilatore specifico

- **User's Guide**

<https://gcc.gnu.org/onlinedocs/gfortran.pdf>



Per iniziare ...

```
> gfortran -o prova program.f90 sub1.f90 ...
```

```
→ prova
```

```
> ./prova
```