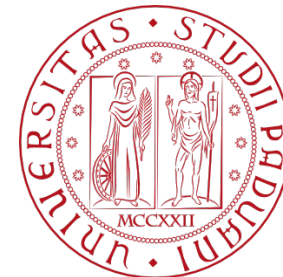


# Programmazione

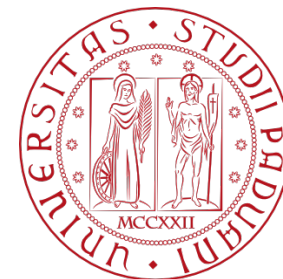
**Giovanni Da San Martino**

Dipartimento of Matematica, Università degli Studi di Padova  
giovanni.dasanmartino@unipd.it  
March-May 2022



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

# Introduzione



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

Il corso ha due docenti:

- Giovanni Da San Martino (responsabile) - [giovanni.dasanmartino@unipd.it](mailto:giovanni.dasanmartino@unipd.it)
- Lamberto Ballan - [lamberto.ballan@unipd.it](mailto:lamberto.ballan@unipd.it)

e 4 assistenti:

- Enrico Cancelli, Andrea Visintin, Francesco Ceccato, Igor Biolcati Rinaldi

- Moodle del corso:  
<https://elearning.unipd.it/math/course/view.php?id=880>
- Useremo un secondo Moodle per gli esercizi di programmazione
  - trovate l'indirizzo nel Moodle sopra (in generale ogni informazione sarà disponibile attraverso il Moodle del corso)

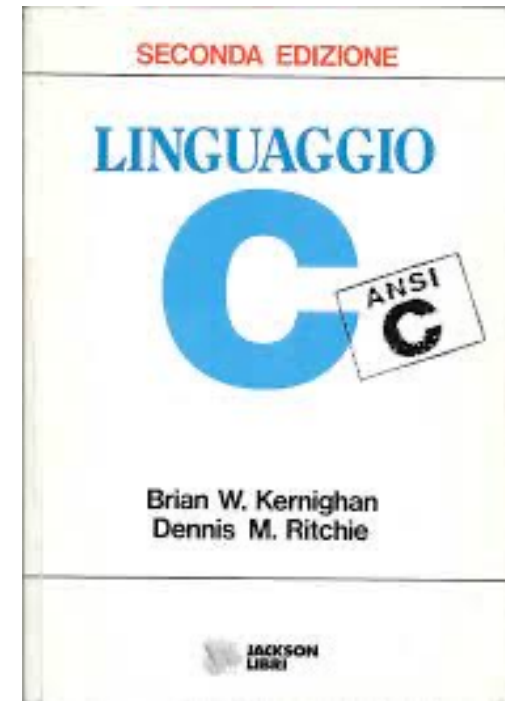
Iscrivetevi ad entrambi!

- Controllare di avere un account per i laboratori funzionante prima di venerdì



- “Il linguaggio C”, Paul Deitel and Harvey Deitel (ottava edizione)

- "The C Programming Language", Brian W. Kernighan Dennis M. Ritchie (disponibile anche in italiano con il titolo "Linguaggio C")



- Insieme [redacted] di istruzioni [redacted] [redacted] per risolvere un problema.
- Il concetto di algoritmo è generale (non ci sono riferimenti al calcolatore)

- Insieme ordinato e finito di istruzioni elementari, chiare e non ambigue, per risolvere un problema.
- Il concetto di algoritmo è generale (non ci sono riferimenti al calcolatore)

- 3 elementi
  1. Il problema da risolvere
  2. La sequenza di istruzioni
  3. L'esecutore delle istruzioni
- 2 attori
  - chi crea le istruzioni
  - chi le esegue





- 3 elementi
  1. Il problema da risolvere
  2. La sequenza di istruzioni
  3. L'esecutore delle istruzioni
- 2 attori
  - chi crea le istruzioni
  - chi le esegue

## PREPARAZIONE

COME PREPARARE GLI SPAGHETTI CACIO E PEPE



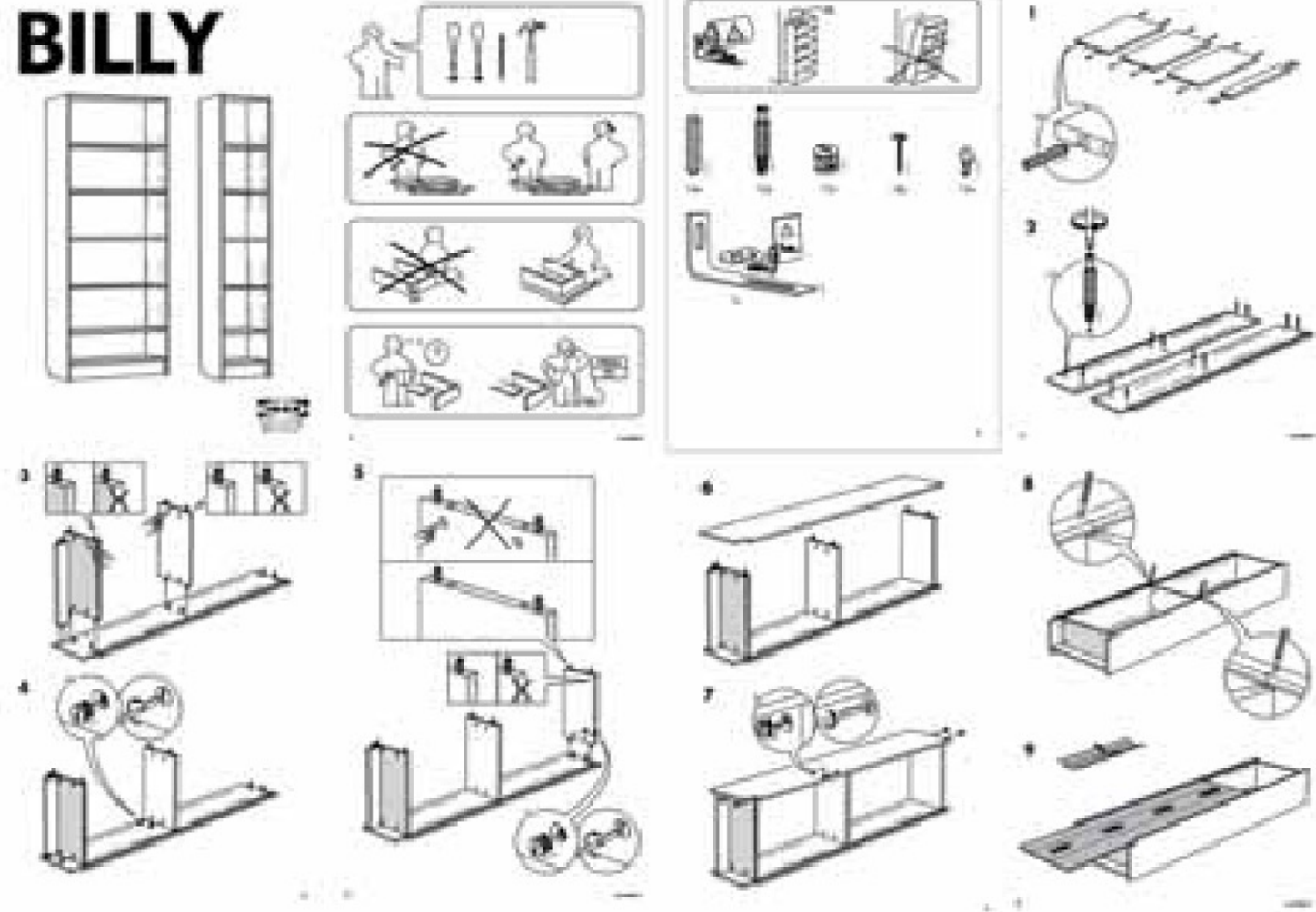
Per preparare gli spaghetti cacio e pepe, come prima cosa occupatevi di grattugiare 200 g di Pecorino. Proseguite mettendo a bollire l'acqua in un tegame (mettetene circa la metà di quanto ne usate di solito per cuocere la pasta, così sarà più ricca di amido) e quando bollerà potete salare a piacere. Una volta salata, potete cuocere gli spaghetti **1**. Nel frattempo, versate i grani di pepe interi su un tagliere **2**, quindi schiacciateli pestando con un pestello per carne o un macinino **3**. In questo modo si

- La risoluzione di un problema spesso implica la risoluzione di una serie di sottoproblemi



- Insieme ordinato e finito di istruzioni elementari, chiare e non ambigue, per risolvere un problema.

## BILLY





- Più difficile di quanto si possa pensare
- Qua sono esseri umani ad eseguire le istruzioni...



- i figli pensano di sapere cosa si debba fare per ottenere il panino,
- ma non riescono a comunicarlo in modo non ambiguo
- Scopo del corso:
  - Non diventare come il bambino!



<https://www.youtube.com/watch?v=Ct-IOOUqmyY>

Programmare significa scrivere algoritmi per il calcolatore

Data la descrizione di un problema da risolvere

1. Si descrive l'algoritmo di risoluzione
2. Si traduce l'algoritmo nel linguaggio di programmazione
  - più adatto oppure in quello imposto dal problema
3. Si fornisce prova che il nostro programma fa quello richiesto
  - Il tipo di prova dipende dal contesto
    - unit tests
    - prove di correttezza

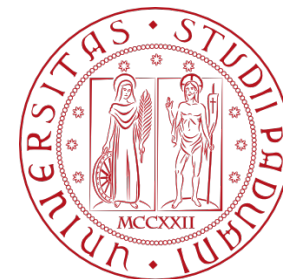
A parità di correttezza, come si può migliorare un programma?

1. Efficienza (tempo e spazio), sia a livello algoritmico che implementativo
  1. Il secondo aspetto è preponderante nel corso, ma tratteremo entrambi
  2. Cenni di complessità
2. Organizzazione: si codifica (per bene) una volta, poi si riusa il codice
  1. Si divide il problema in sottoproblemi e si cerca di riusare le soluzioni già implementate in passato
3. Stile
  1. Il codice deve essere comprensibile ai tuoi colleghi e a te stesso tra 3 mesi.

- Pensare un algoritmo è difficile. Si impara con l'esperienza
  - Ovvero fate gli esercizi!
- Esame: ci saranno due compitini (il primo la settimana dopo Pasqua)
  - Scritti, con domande sui temi del corso e prove di programmazione
  - L'esame sarà effettuato al calcolatore\*



# Compilazione

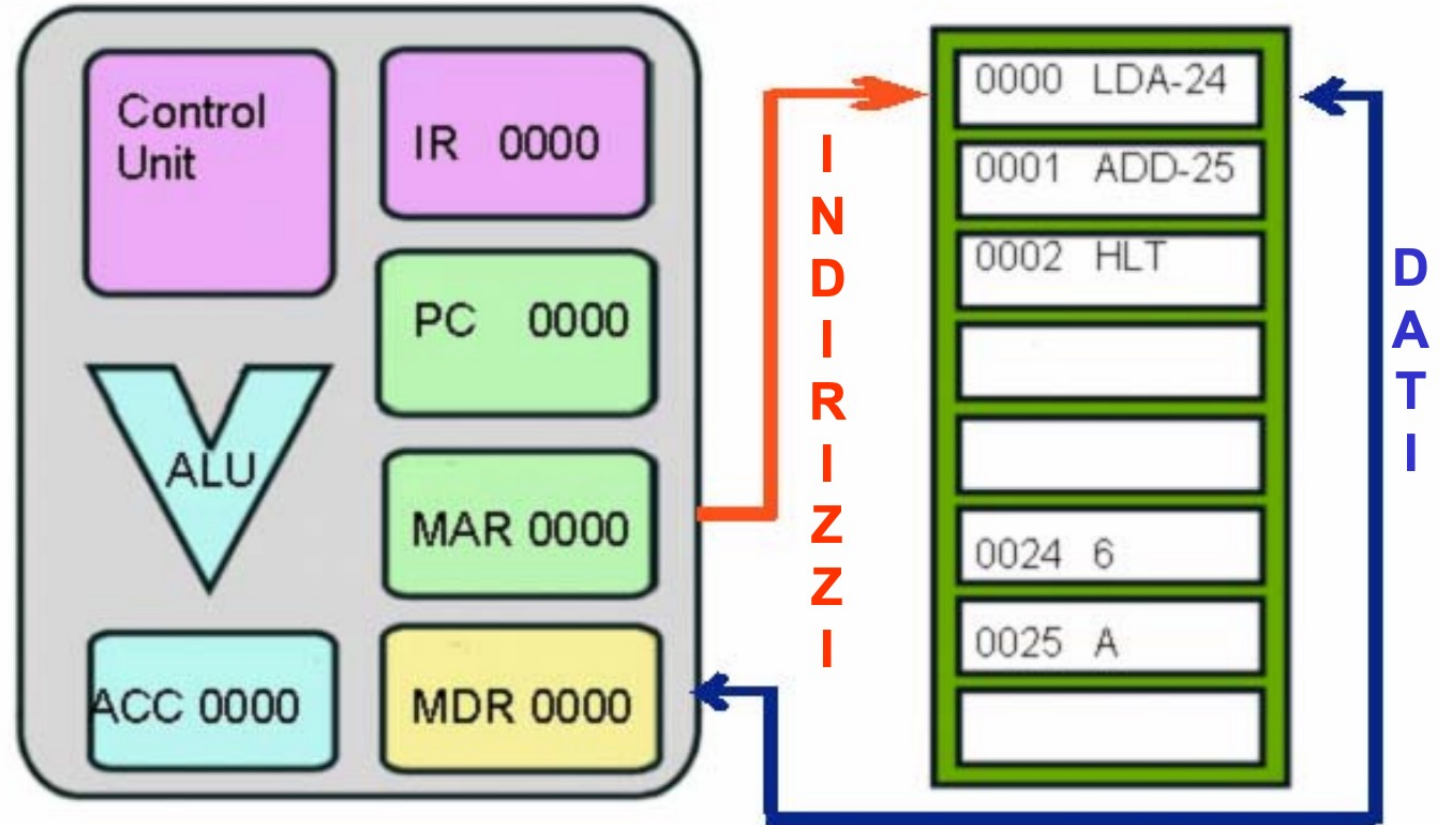


UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

- Programmare significa comunicare un algoritmo al calcolatore
- Quale linguaggio conosce il calcolatore?

- Il Computer è capace di eseguire programmi in linguaggio macchina, tramite il ciclo FDE

- [F] - estrae una istruzione dalla memoria (in codice binario)
- [D] – individua i circuiti corrispondenti all'istruzione
- [E] - attiva tali circuiti



- Il linguaggio della macchina è un linguaggio di basso livello (il livello di dettaglio tende a far perdere la visione d'insieme)
- Se cambia architettura, dobbiamo aggiornare il programma ☹️
- Difficile da ricordare (sequenze di 0 e 1)
  - Soluzione: Il linguaggio assembler è una versione mnemonica del linguaggio macchina, es. ADD invece della sequenza di bit dell'addizione
  - l'assemblatore sostituisce le istruzioni con i corrispondenti codici macchina

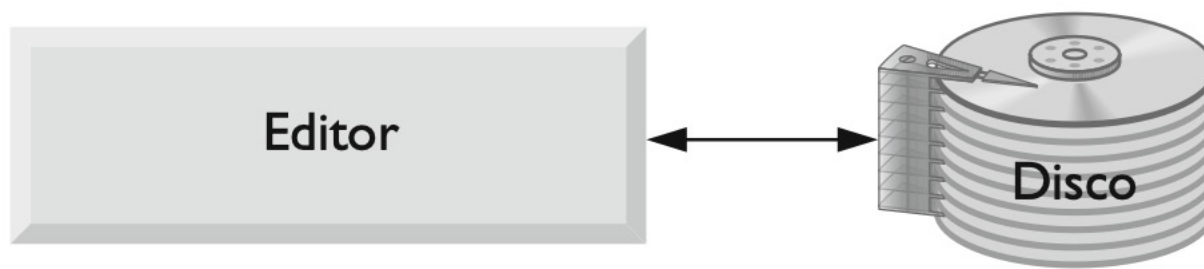
- Il linguaggio della macchina è un linguaggio di basso livello (dipende dall'architettura)
- Si può avere un linguaggio di più alto livello e che ci permetta di evitare di implementare più volte lo stesso programma per architetture diverse?
- Assieme alla specifica del linguaggio, si fornisce uno strumento che traduca i nostri programmi nel linguaggio della macchina ospite: il traduttore



COMPILER

- Interprete: traduce una istruzione di alto livello e la esegue immediatamente
- Compilatore: traduce tutte le istruzioni assieme che vengono poi eseguite tutte assieme direttamente in linguaggio macchina (C, C++)
- Esistono soluzioni intermedie: compilazione in bytecode ed interpretazione (Java)

- Interprete:
  - più lenta l'esecuzione del programma.
  - necessita del traduttore per eseguire il programma
  - se si ha il traduttore ed il codice sorgente, può essere eseguito su ogni computer
- Compilatore:
  - più veloce l'esecuzione (di solito riesce anche a ottimizzare il codice)
  - non necessità del traduttore, ma ogni volta che cambio il programma devo ricompilarlo
  - il codice deve essere compilato per ogni diversa architettura
- Scelte del C: linguaggio compilato; insieme ristretto di comandi di base (ci si affida a librerie di funzioni), il compilatore è "facile" da scrivere, quindi portabilità



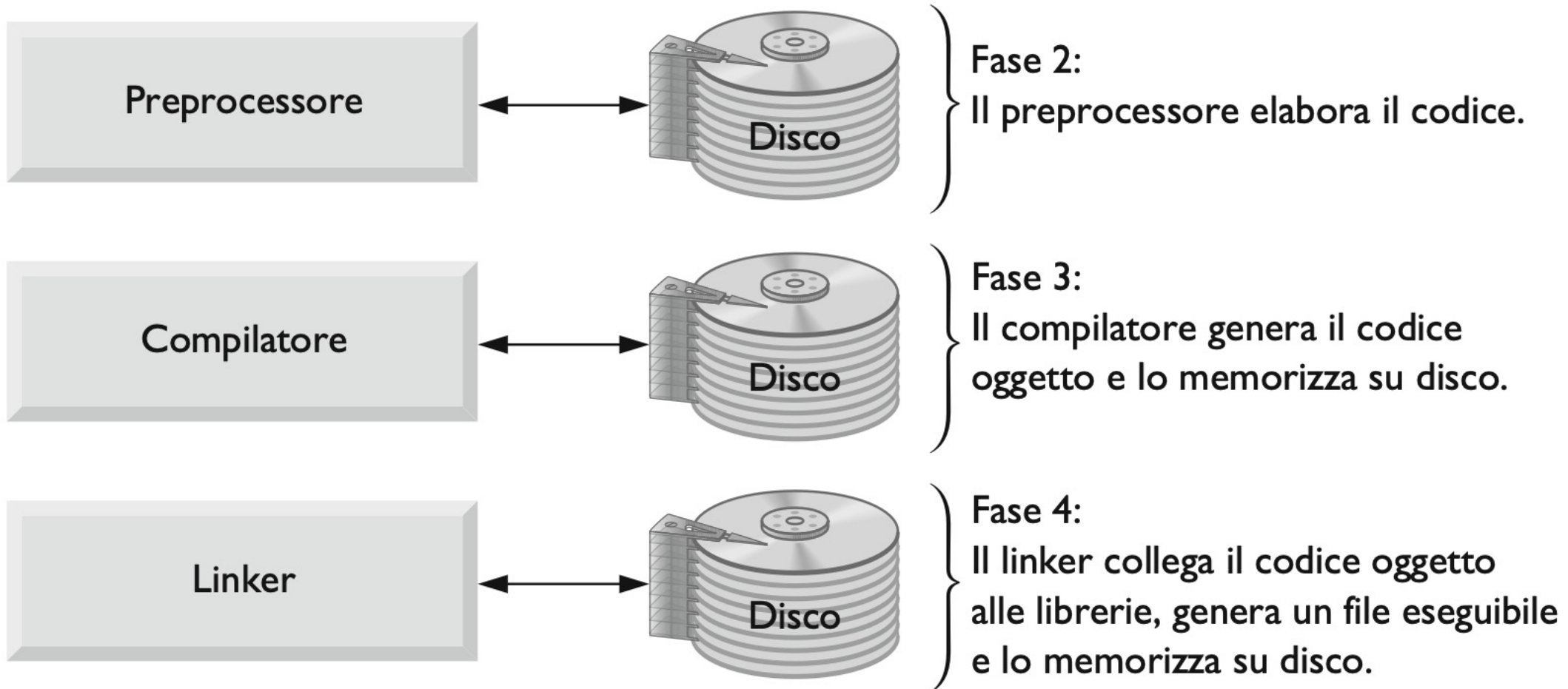
Fase1:

Il programmatore crea il programma con l'editor e lo memorizza su disco.

```
#include <stdio.h>

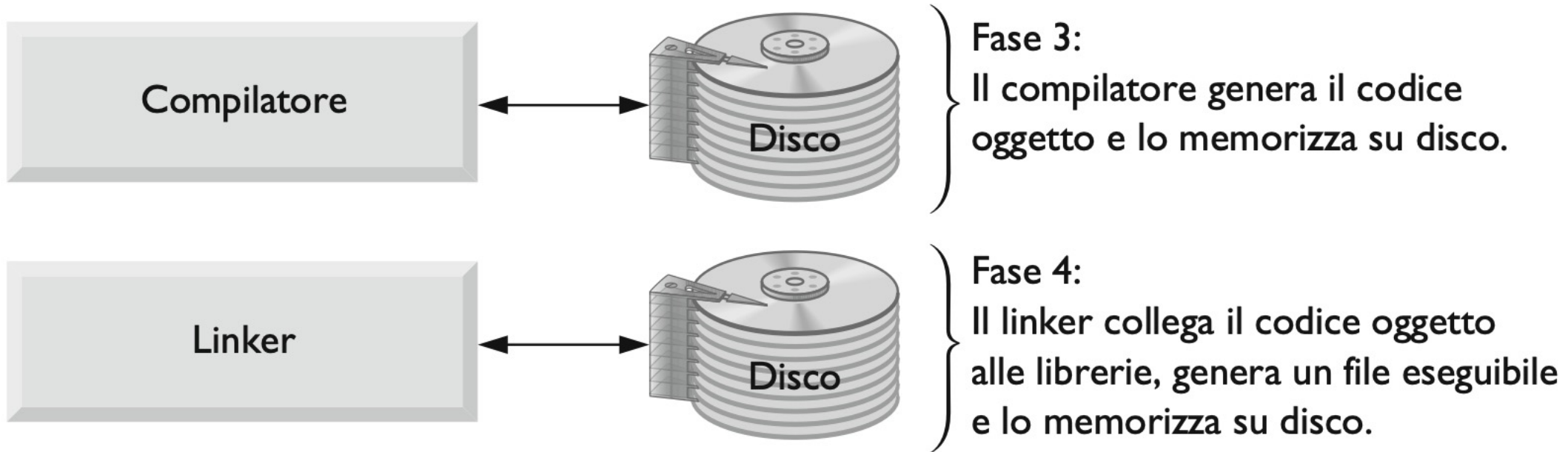
int main (void) {
    printf("Ciao Mondo!\n"); //stampa il messaggio Ciao Mondo!
}
```







- Rimozione dei commenti
- `#include <x>`: il contenuto del file `x` viene copiato all'inizio del nostro file (`x` contiene informazioni su come eseguire comandi aggiuntivi, ad esempio `stdio.h` permette di utilizzare il comando `printf`)
- Espansione delle macro (le vedremo tra qualche lezione)
- Compilazione condizionale (utile se alcune librerie hanno nomi diversi in diversi sistemi operativi)



Un programma è generalmente composto da molti file ed utilizza funzioni già scritte da altri (printf). Per evitare di duplicare il codice di tali funzioni, si caricano in memoria una volta e si collegano al nostro programma (linking)

```
#include <stdio.h>

int main (void) {
    printf("Ciao Mondo!\n"); //stampa il messaggio Ciao Mondo!
}
```

- Il Computer è capace di eseguire programmi in linguaggio macchina
- Bisogna tradurre dal linguaggio ad alto livello a quello macchina!
- gcc -o ciao hello\_world.c
- Il codice tradotto è a questo punto eseguibile: [su linux] ./ciao

- Esiste un unico compilatore? NO
- Il C è nato negli anni 70, ha avuto molto successo, per cui molti compilatori sono stati creati indipendentemente
- ANSI C: una serie di specifiche che standardizzano il comportamento del compilatore