

Compilatori e interpreti

Fondamenti di Informatica A

Percorso di Preparazione agli Studi di Ingegneria

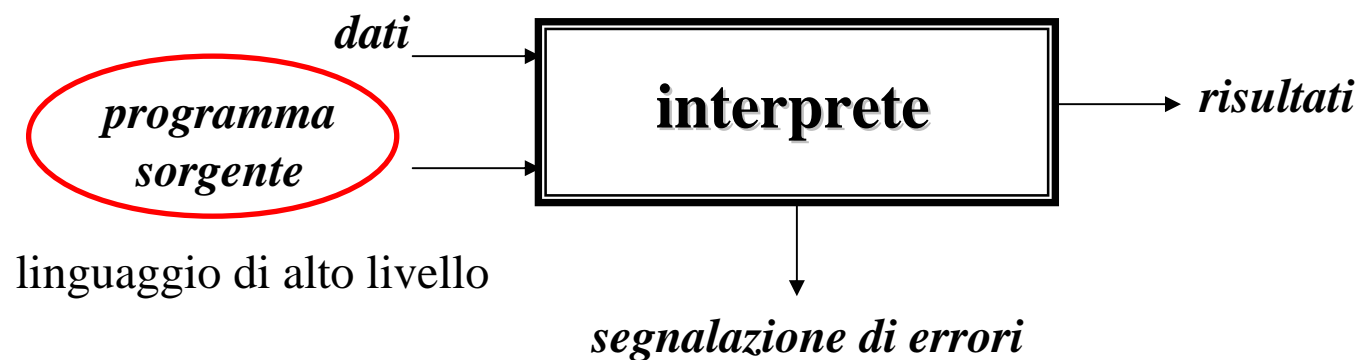
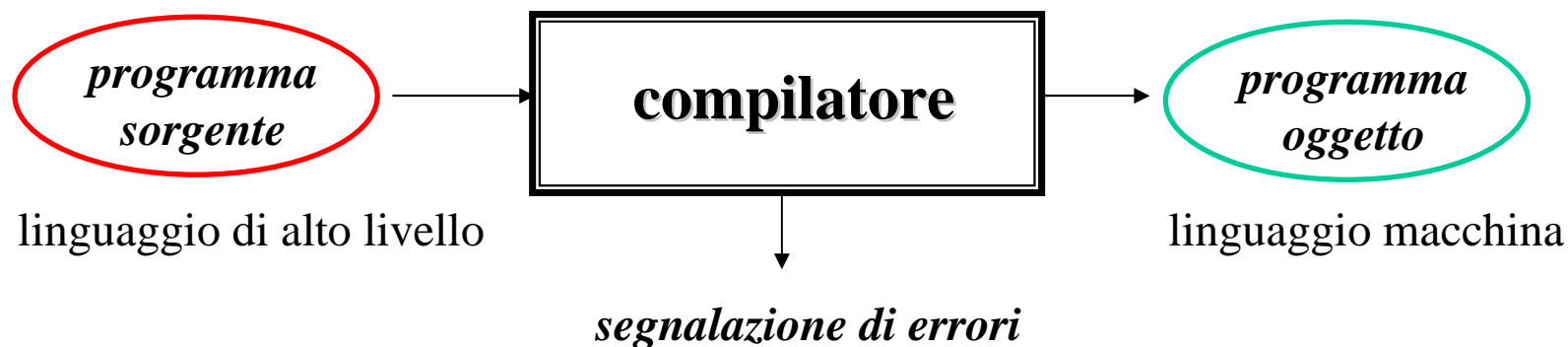
Università degli Studi di Brescia

Docente: Massimiliano Giacomini

Linguaggio di Programmazione (di alto livello)

- Ogni istruzione di un linguaggio di alto livello è in **corrispondenza 1-a-molti** con il linguaggio macchina
- Questi linguaggi permettono la costruzione di sistemi molto complessi
- Hanno avuto un ruolo importante nell'evoluzione dell'informatica
- Sono indipendenti dalla macchina
- Un programma scritto in linguaggio di alto livello per essere eseguito deve essere **tradotto** in linguaggio macchina

Compilatore e interprete



Eseguire un programma scritto in un linguaggio compilato

- Il programma P scritto in linguaggio L viene dato in ingresso a un programma P_{Comp}
- P_{Comp} è il **programma compilatore** del linguaggio L (ad esempio il programma compilatore del C)
- L'esecuzione da parte di un calcolatore di P_{Comp} su P (dove P è il dato di ingresso) produce P_{exe} → **PRIMA FASE**
- L'esecuzione da parte di un calcolatore di P_{exe} su dei dati iniziali produce dei risultati finali → **SECONDA FASE**

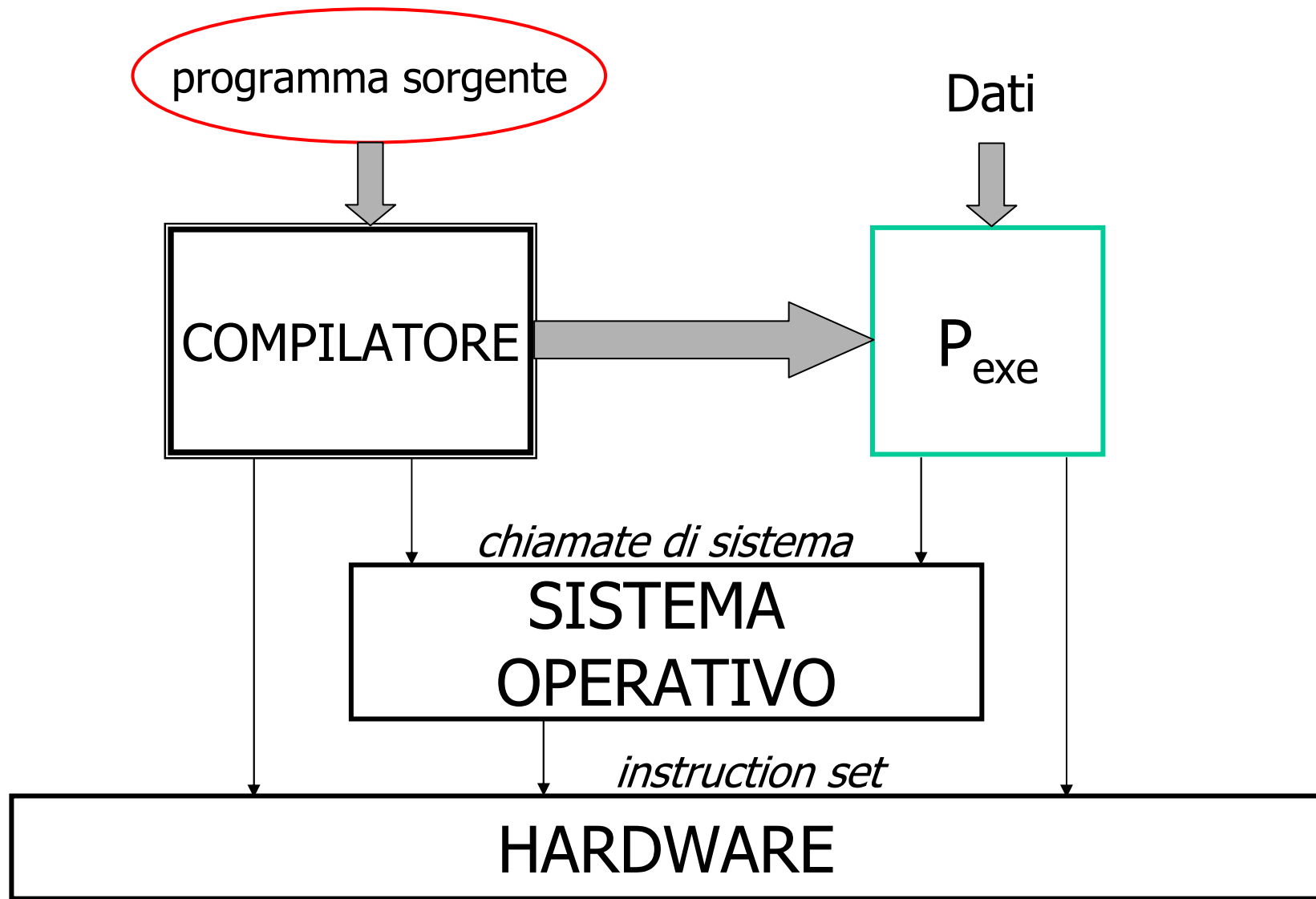
Eseguire un programma scritto in un linguaggio interpretato

- Il programma P scritto in linguaggio L viene dato in ingresso a un programma P_{Int}
- P_{Int} è il **programma interprete** del linguaggio L (ad esempio il programma interprete del Basic)
- L'**esecuzione da parte di un calcolatore di P_{Int}** su P con i dati in ingresso di P produce i risultati finali

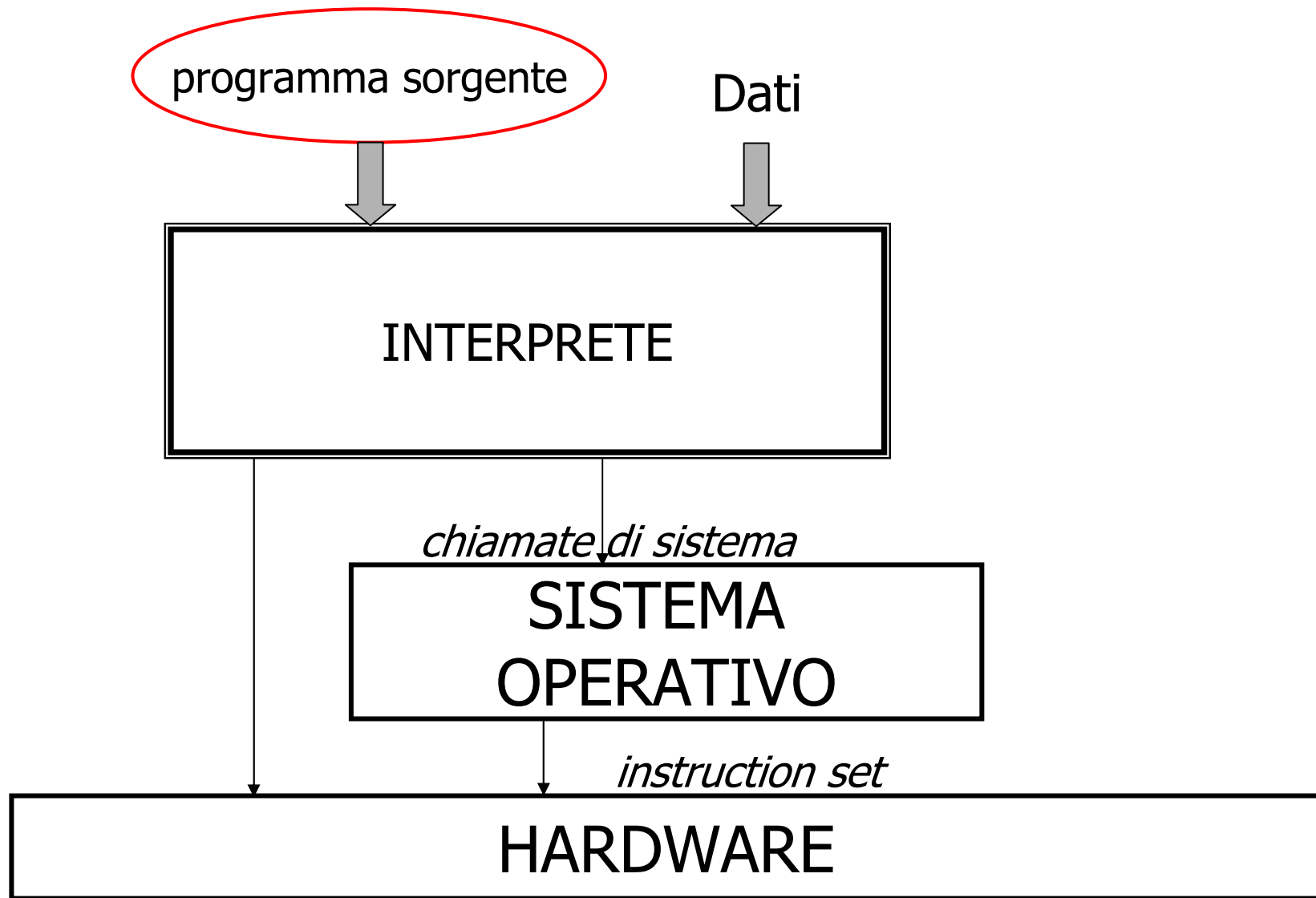
Domanda

- Quali proprietà fondamentali distinguono un compilatore (o un interprete) da un altro?
- ✓ Il linguaggio di alto livello
-
- ✓ L'insieme delle istruzioni macchina (ISA) (semplificando, il tipo di processore)
 - ✓ Il sistema operativo

Compilatore, sistema operativo, hardware



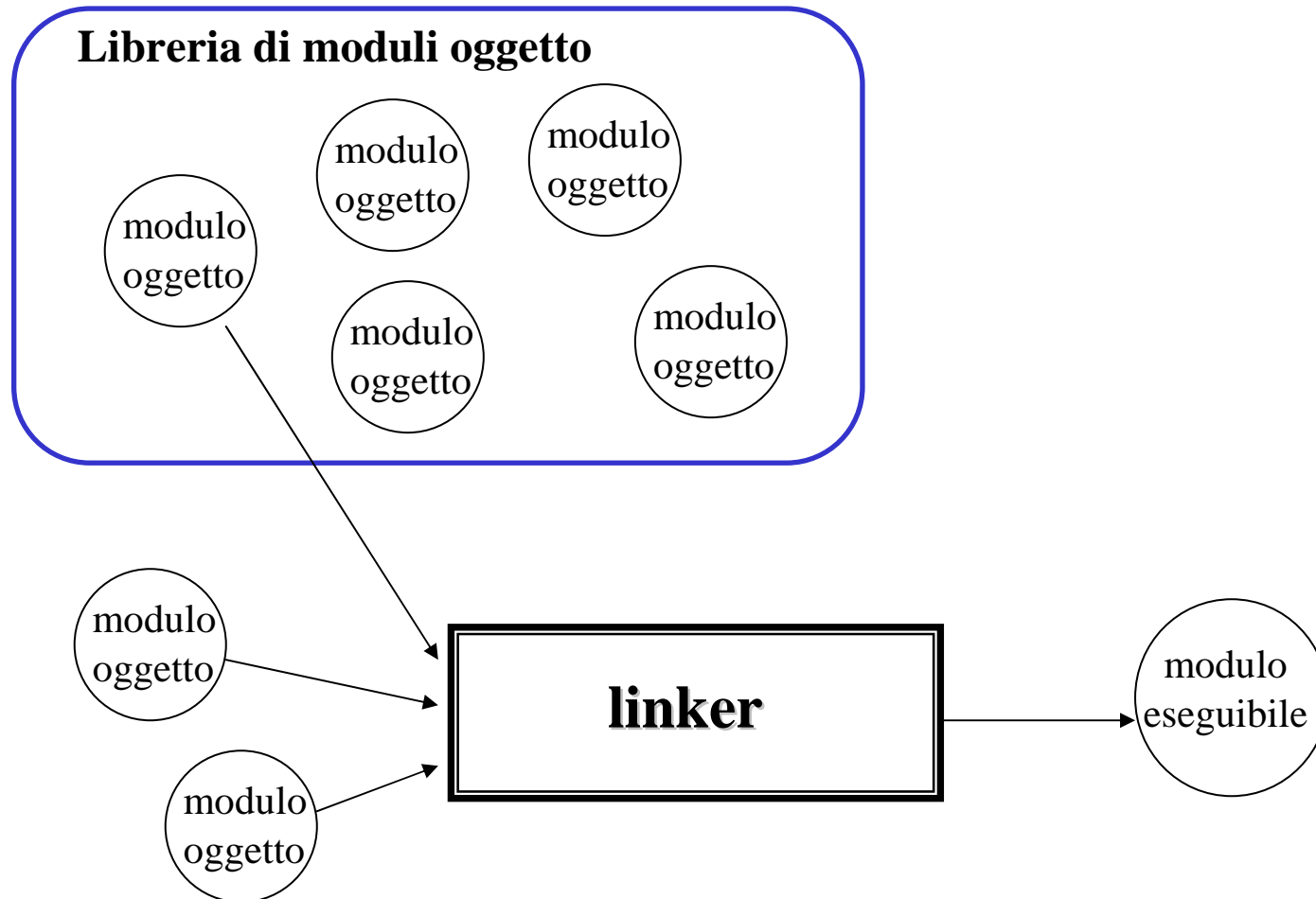
Interprete, sistema operativo, hardware

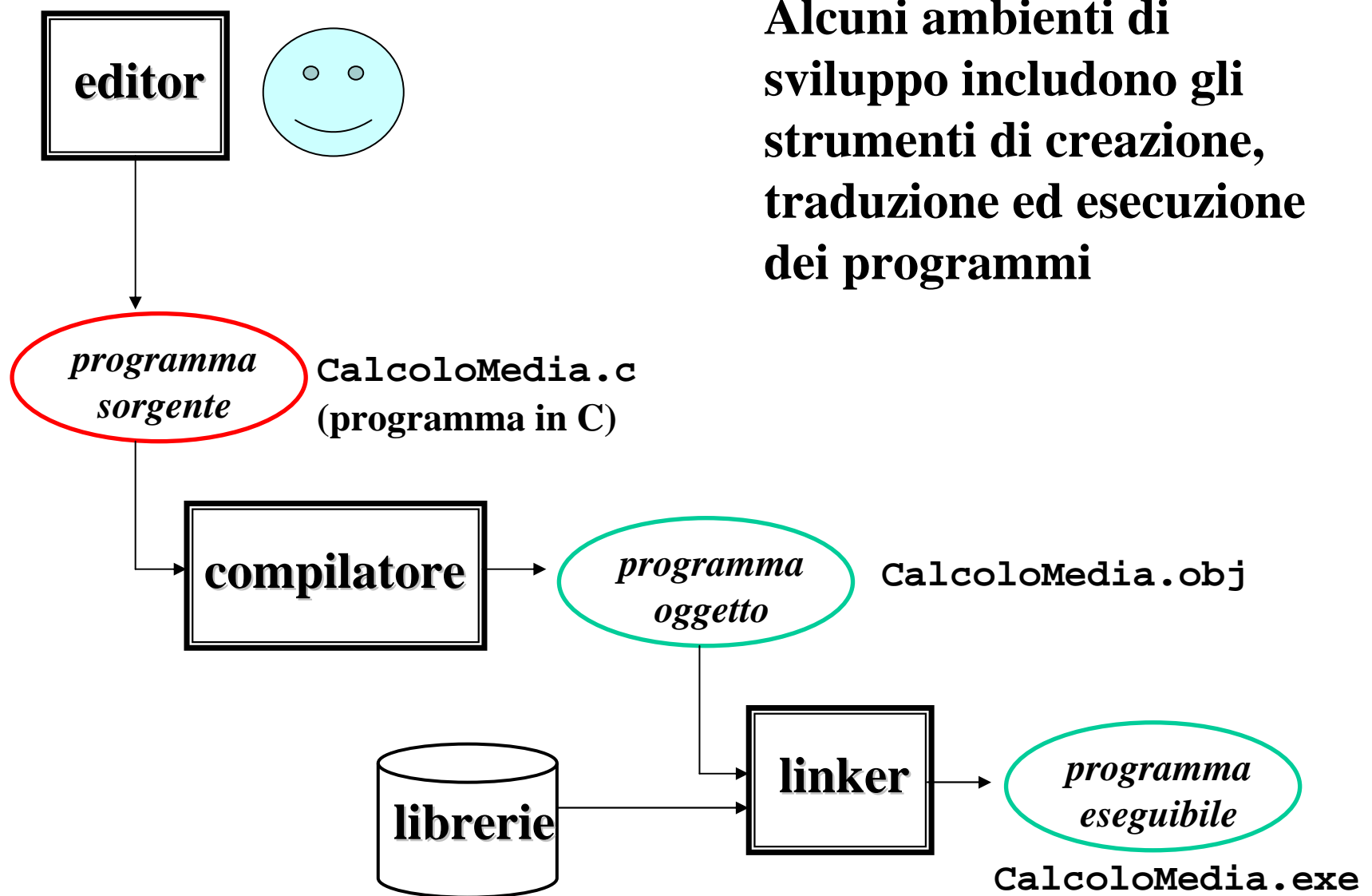


Compilatore e linker

- I compilatori consentono tipicamente la compilazione separata di parti di programmi (*moduli*)
- I diversi moduli possono essere progettati, costruiti e messi a punto separatamente, e archiviati in opportune *librerie*
- Nel momento in cui un programma deve essere eseguito, un programma apposito, detto *linker*, si occupa di ritrovare e collegare opportunamente fra loro i moduli oggetto
- Il risultato del linker è un unico modulo, detto *modulo eseguibile*, pronto per il caricamento in memoria e l'esecuzione

Il ruolo del linker





Alcuni ambienti di sviluppo includono gli strumenti di creazione, traduzione ed esecuzione dei programmi

Confronto fra compilatori e interpreti

- *Velocità* di esecuzione: i programmi compilati hanno in genere prestazioni migliori
- L'esecuzione di un programma interpretato richiede che in memoria sia presente anche l'interprete
- Nella compilazione si possono attuare processi di *ottimizzazione* dell'eseguibile
- *Messa a punto* del programma: gli interpreti permettono di correggere gli errori non appena vengono scoperti, senza bisogno di ricompilare interamente il programma

Confronto fra compilatori e interpreti: portabilità

- Si definisce *portabilità* la possibilità di utilizzare un programma su piattaforme hardware/software diverse da quella dove è stato sviluppato
- Tre modalità:
 - portabilità del file eseguibile: richiede calcolatori con hardware e sistema operativo dello stesso tipo per il quale è stato compilato. Semplice ma restrittivo.
 - portabilità tramite ricompilazione: richiede l'esistenza di un compilatore e di librerie per la nuova piattaforma e lo svolgimento della ricompilazione. Non fattibile dall'utente medio.
 - portabilità tramite interpretazione: richiede l'esistenza di un interprete per la nuova piattaforma. Semplice (purchè l'interprete sia già installato) e non restrittivo.