Programmazione I

Lezione 3

Programmazione ed architetture

- La **macchina** comprende un solo tipo di linguaggio, di tipo binario: l'attivazione o meno dei suoi circuiti

  (linguaggio macchina)

- Per renderlo leggibile, al linguaggio macchina è associato un linguaggio di **basso livello** con corrispondenza (sostanzialmente) 1 a 1

  (assembly)

- Le istruzioni sono conservate in memoria e gestite con
  - Fetch
  - Decode
  - Execute

Esempio “ibrido”: Register Machines

- Esperimento “bottom up”: il modello di calcolo astratto delle Register Machines
- Struttura delle Register Machines
- Primitive:
  - INC (r)
  - DEC (r)
  - JNZ (r, etichetta)

  Esempio: creare JZ, J, Reset e Somma usando le operazioni primitive (alla lavagna)
- Discussione: macro e riutilizzo di codice assembly
- Discussione: istruzioni, circuiti, macro, RISC e CISC

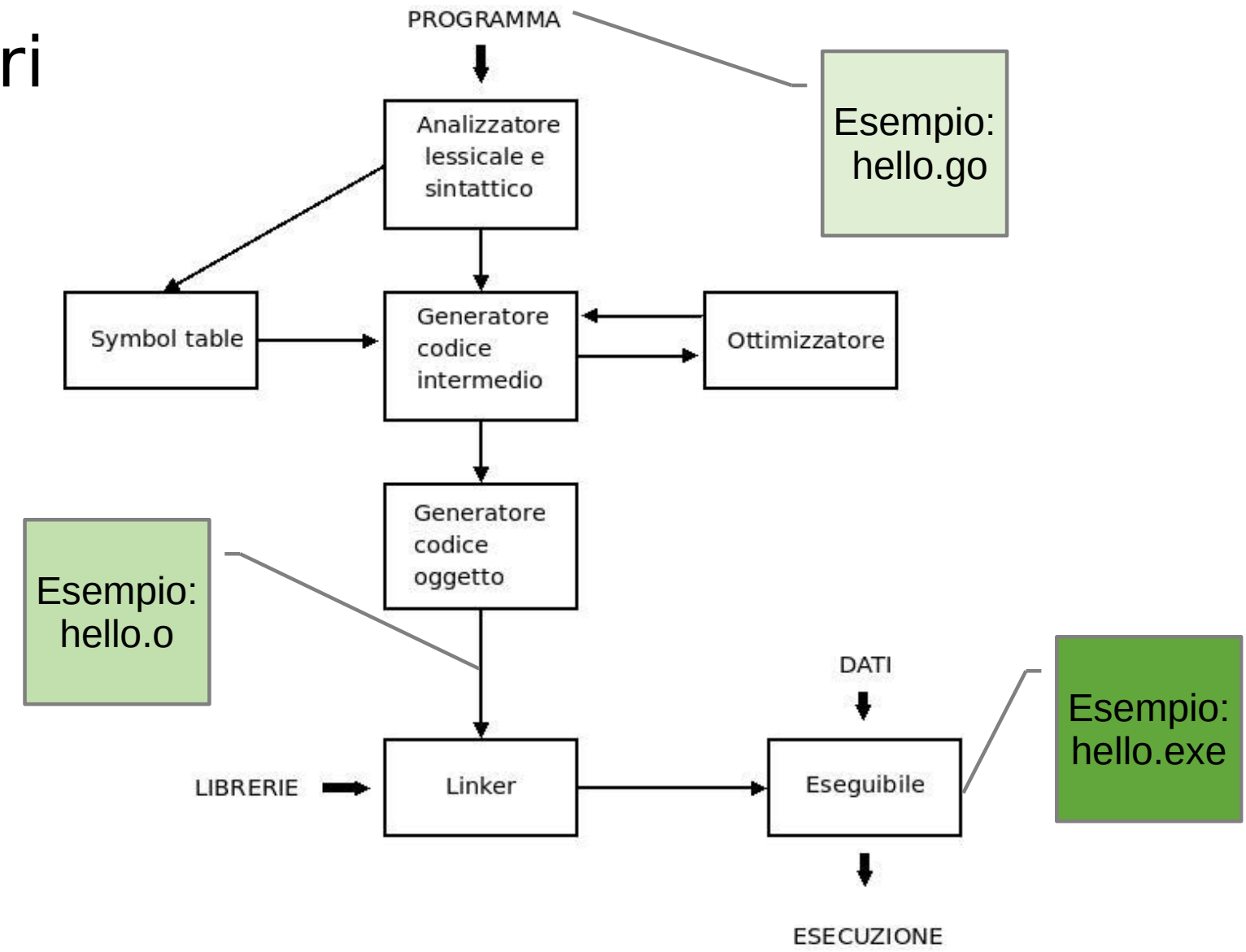
Recap: programmazione ed architetture

- Organizzare la complessità di un problema reale utilizzando istruzioni assembly è **teoricamente possibile**, ma **praticamente impercorribile**.

- Procedimento “top down”

- Ci affidiamo a linguaggi ad **alto livello**, che poi sono tradotti in modo automatico in linguaggi a più basso livello

Compilatori



Dal sorgente all’eseguibile

Per ottenere un’applicazione:

- Editing codice sorgente (programmazione vera e propria)
- Creazione di codice oggetto **(compile time)**
- Creazione eseguibile **(linking time)**
- Esecuzione **(run time)**

Esempio: hello.go

Esempio: visualizzazione assembly hello.o

Esempio: esecuzione hello.exe