

ASSEMBLATORE

L'assemblatore traduce un programma scritto in linguaggio assembly nel corrispondente programma in linguaggio macchina eseguibile.

E' quindi costituito da un programma che legge tutte le istruzioni assembly che costituiscono il programma, traduce in linguaggio macchina i codici operativi, i dati e le label, controllandone la correttezza sintattica, e genera un file eseguibile.

Livello implementato tramite **compilazione** e non interpretazione



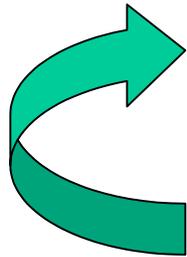
Formato istruzione assembly

<i>Label :</i>	<i>cod. operat.</i>	<i>Operando/i</i>	<i>; commento</i>
----------------	---------------------	-------------------	-------------------

Esempio di source code

```
MOV SI, 0
MOV CX, 7
CICLO: CMP VET[SI], BX
        JE BINGO
        INC SI
        LOOP CICLO
BINGO: RET
```

**Backward
reference**



Forward reference



Label : CICLO, BINGO

ILC = Instruction Location Counter

contiene il valore numerico associato alle label

Problema delle forward reference !

ASSEMBLATORE A DUE PASSI

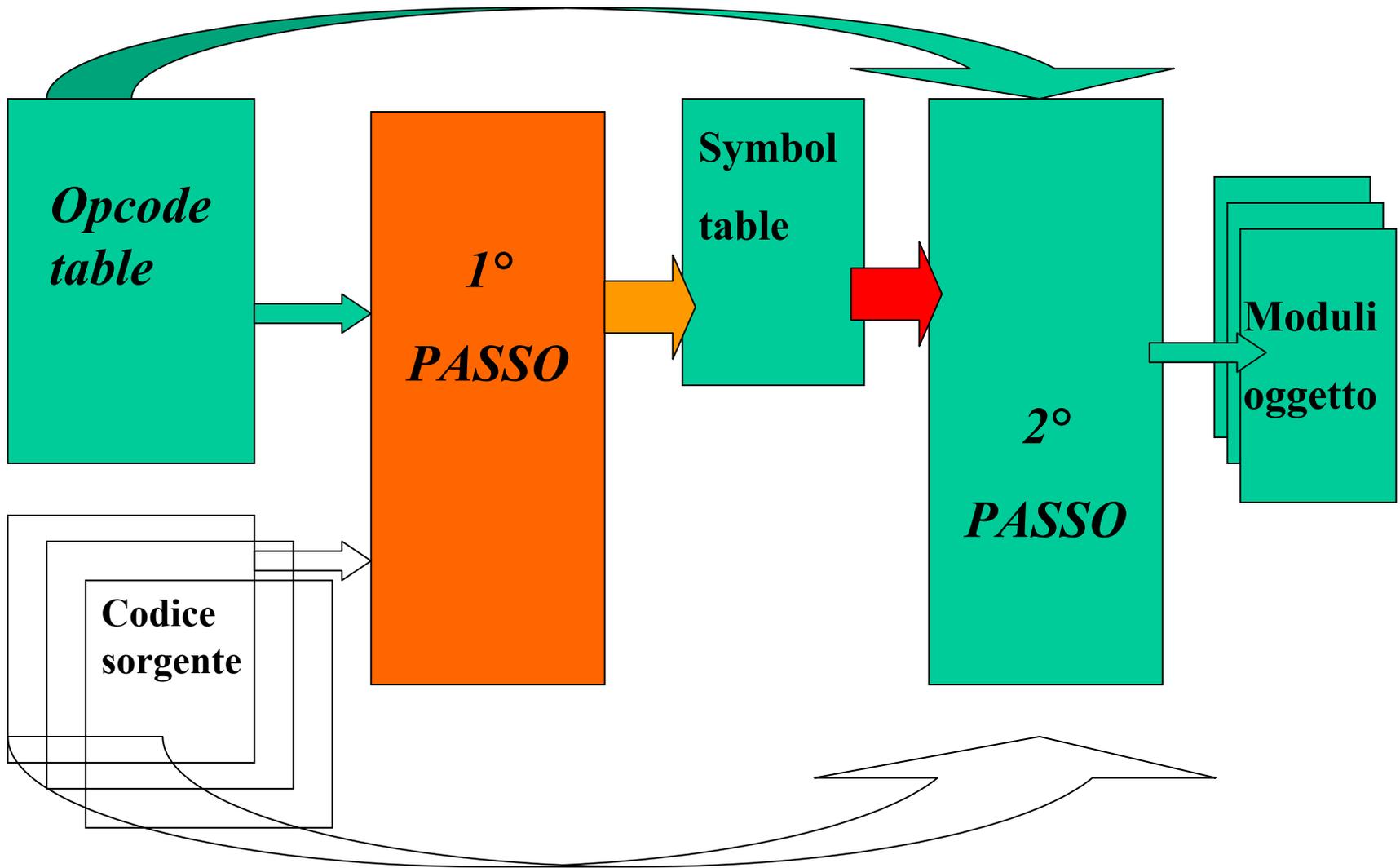
- 1° Passo* : - individuazione di tutti i nomi (le label) che compaiono come riferimento simbolico di dati o di istruzioni
- creazione di una Symbol Table che contiene le label con la loro posizione relativa all'interno del programma
- 2° Passo* : - traduzione dei codici mnemonici delle istruzioni, degli operandi e delle label, mediante la consultazione della symbol table costruita nel 1° passo.

Pseudoistruzioni

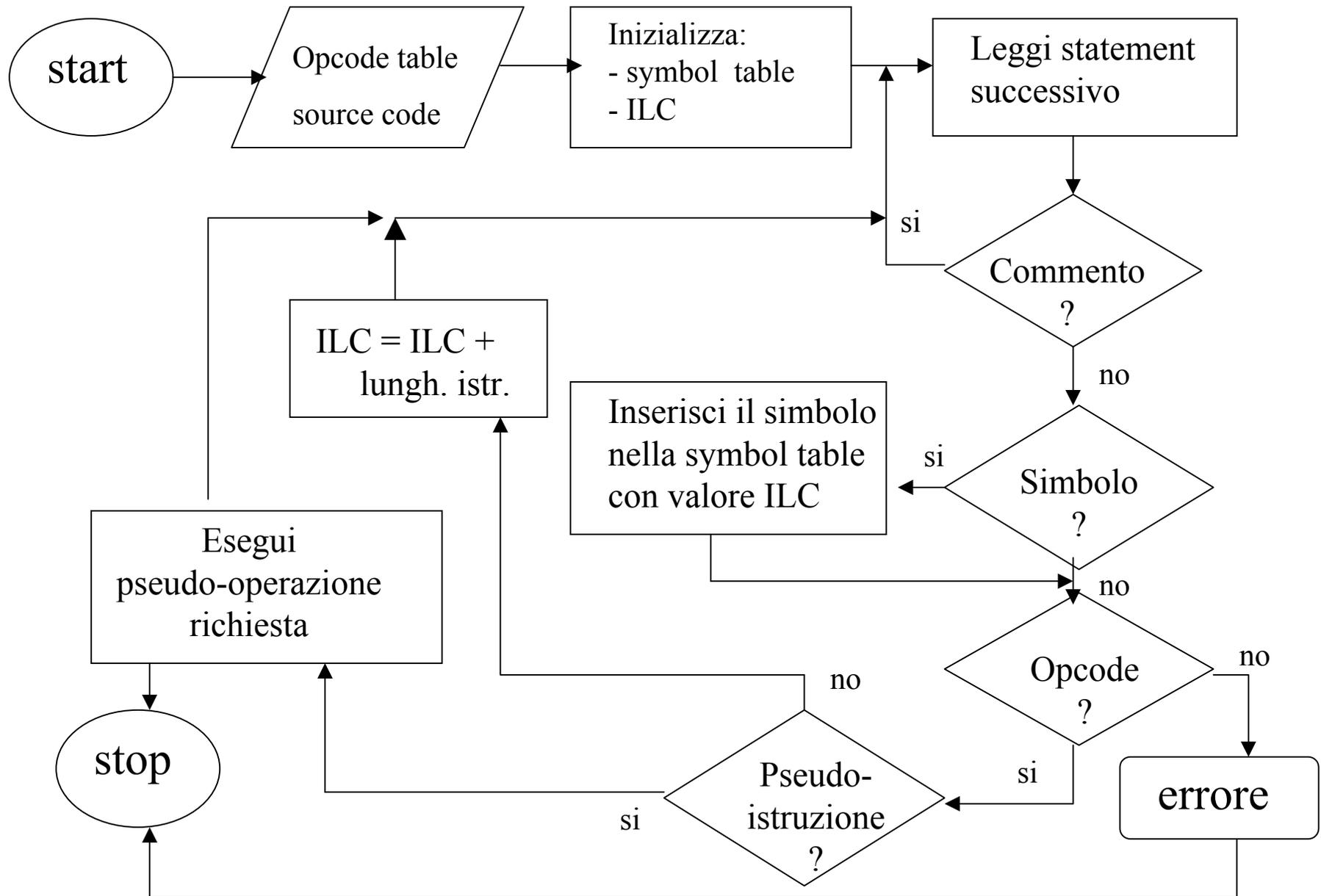
- Il linguaggio assembler consente al programmatore di specificare informazioni indispensabili per la traduzione del programma sorgente in programma oggetto
- *Pseudo-istruzioni* : Istruzioni il cui codice operativo rappresenta un *comando* (direttiva) all'assembler.

Le pseudo-istruzioni infatti non compaiono nel programma oggetto alla fine della fase di traduzione

ASSEMBLATORE a due passi



1° passo dell'assemblatore



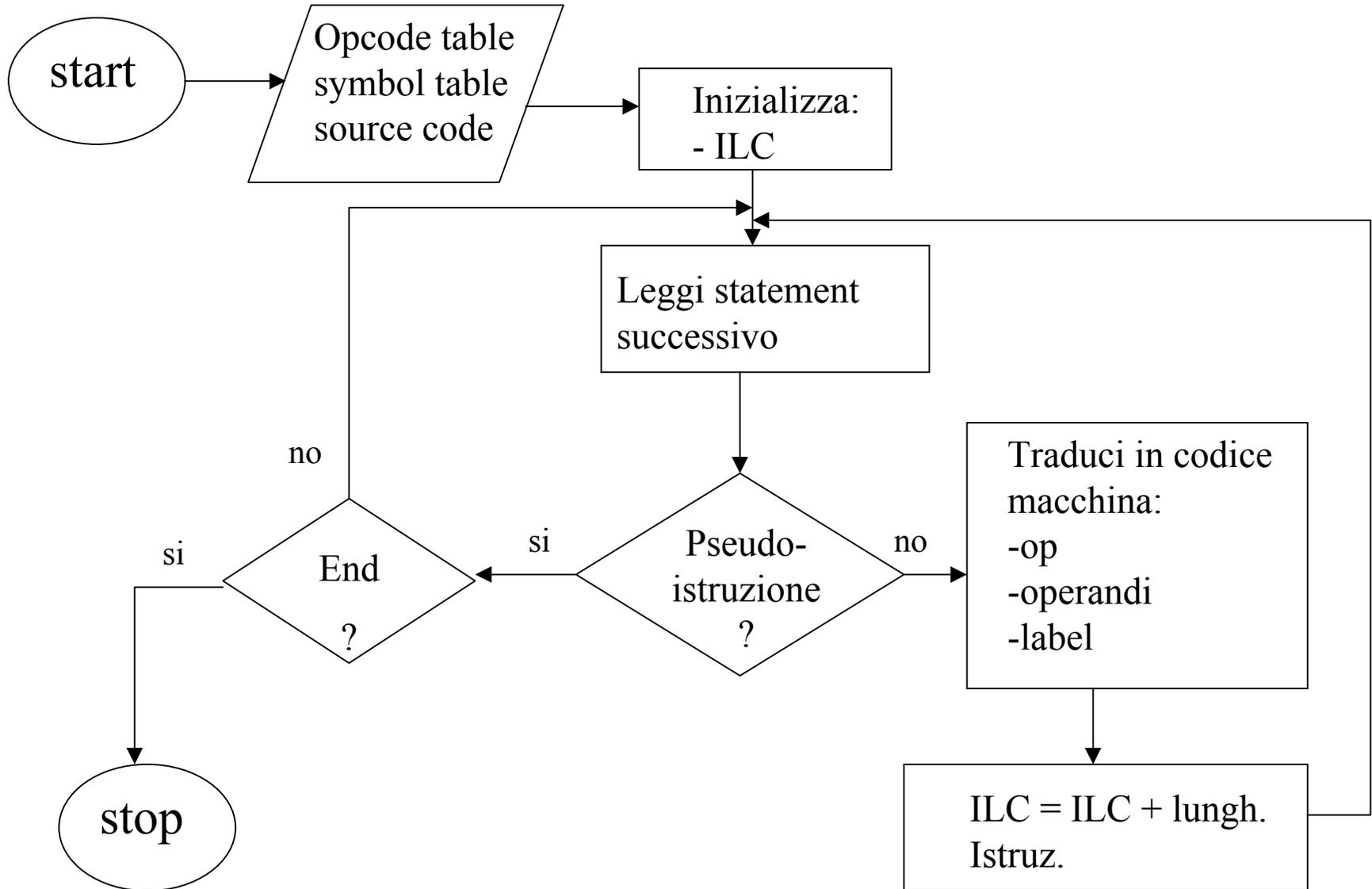
Symbol table

<i>Simbolo</i>	<i>valore ILC</i>	<i>altre informaz.</i>
.....
CICLO	124	- lunghezza del data field associato al simbolo,
BINGO	128	- bit di rilocazione,
		- se il simbolo è accessibile al di fuori della procedura
.....

Opcode table

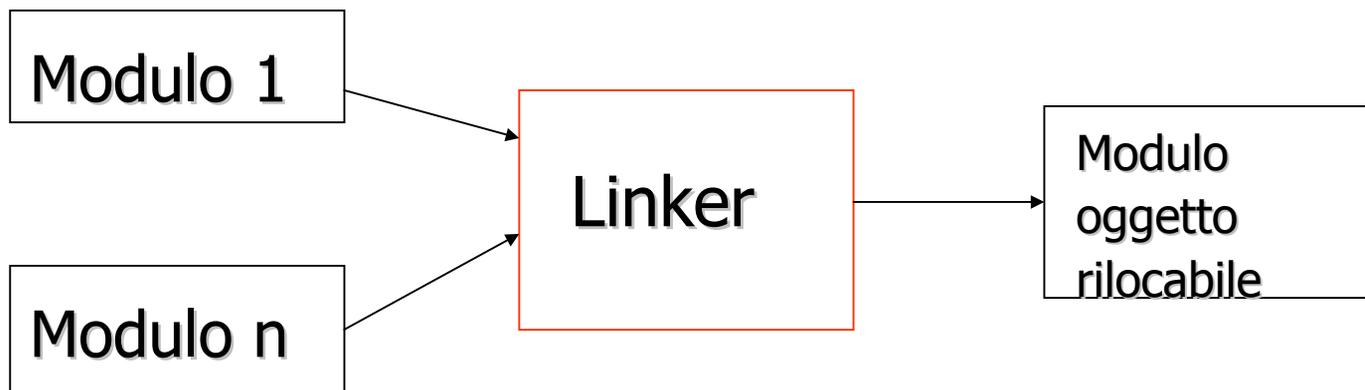
<i>Codice</i>	<i>opcode hex.</i>	<i>lungh. in byte</i>	<i>classe.....</i>
MOV	5A	3	1
ADD	4A	2	2
JE	1C	2	3
SUB	19	2	2
MUL	D2	2	2
.....

2° passo dell'assemblatore

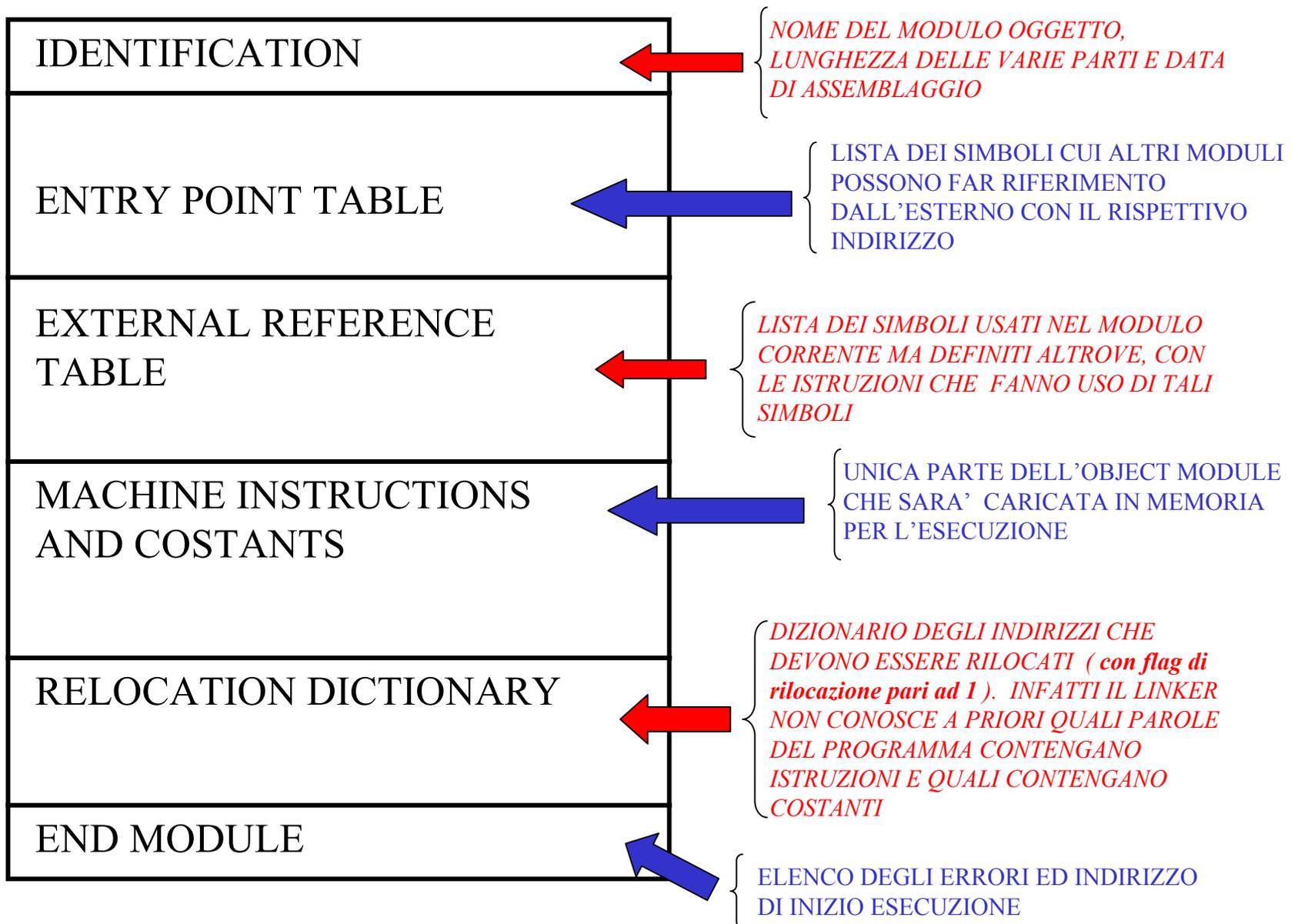


Processo di collegamento

- *Programmi: insieme di procedure (moduli) tradotti separatamente dall'assemblatore (o compilatore). Ogni modulo oggetto ha il suo spazio di indirizzamento separato*
- **Linker** : *programma che esegue la funzione di "collegamento " dei moduli oggetto in modo da formare un unico modulo eseguibile*

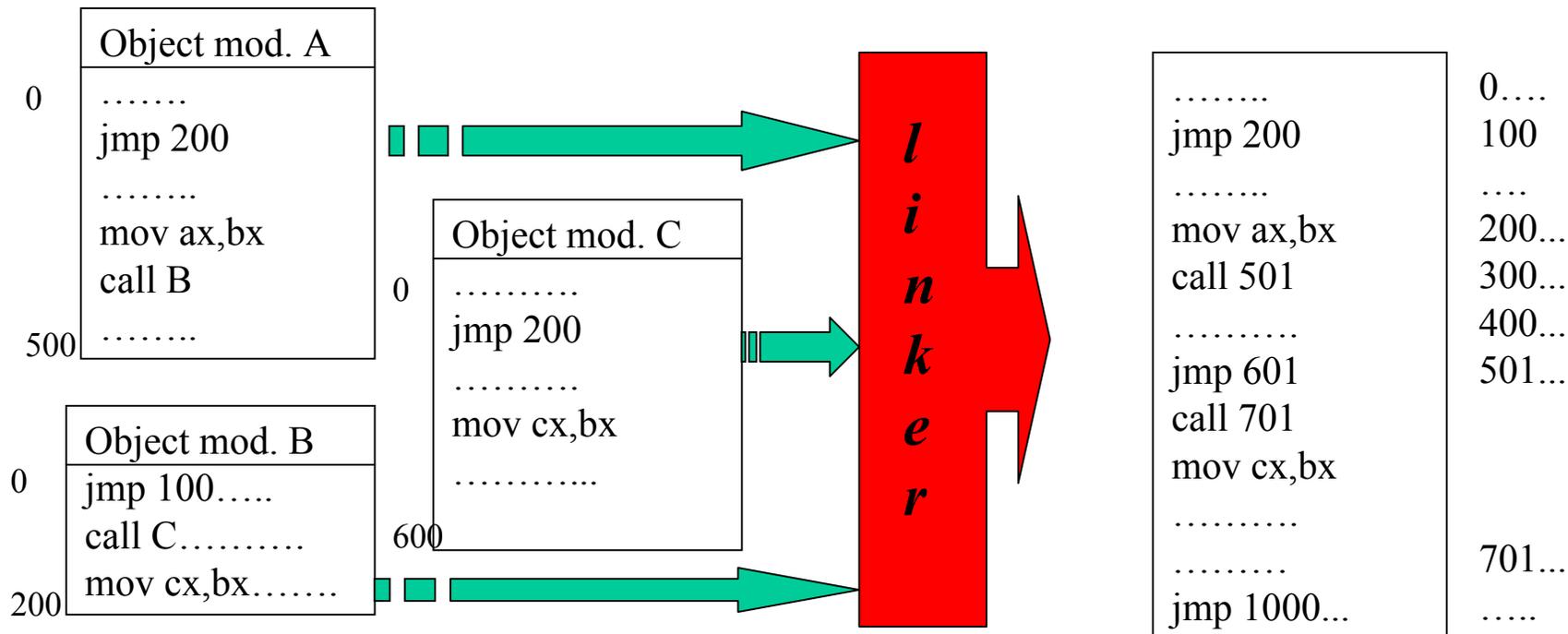


Struttura modulo oggetto



Linker

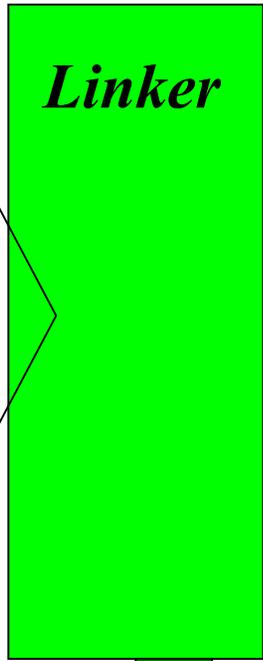
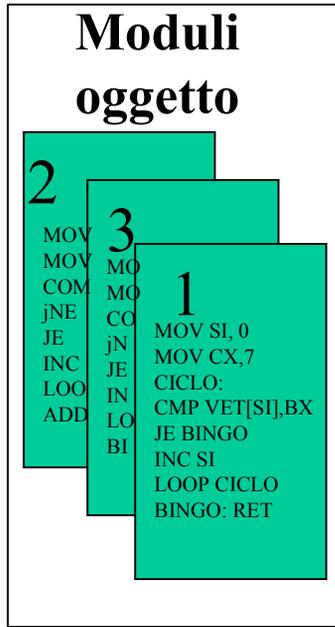
- *Il linker:*
 - *crea una tabella con tutti i moduli oggetto e la loro lunghezza*
 - *assegna un indirizzo di inizio (di load) ad ogni modulo oggetto*
 - *riloca gli spazi di indirizzamento dei vari moduli (**Rilocazione**)* : in tutte le istruzioni che contengono un indirizzo di memoria somma una “**costante di rilocazione**” uguale all’indirizzo di inizio del modulo in cui la istruzione è contenuta
 - *assegna nelle istruzioni di chiamata a procedure esterne l’indirizzo di partenza di queste procedure (**External Reference**),*



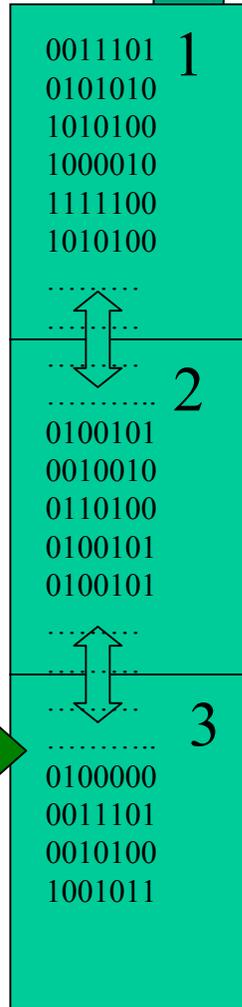
Processo di caricamento

L'OUTPUT DEL LINKER VIENE LETTO DA UN PROGRAMMA "CARICATORE" (**LOADER**), LA CUI FUNZIONE E' QUELLA DI REGISTRARE NELLA MEMORIA CENTRALE IL PROGRAMMA PRONTO PER L'ESECUZIONE, A PARTIRE DA UN DETERMINATO INDIRIZZO STABILITO DAL Sistema Operativo.

- **Loader** : *programma che registra nella memoria centrale il programma binario eseguibile prodotto dal linker, determinando il legame tra indirizzi logici e fisici (binding).*
- *I Meccanismi per facilitare la mappatura degli indirizzi logici in indirizzi fisici sono:*
 - *la Segmentazione o la Paginazione;*
 - *il collegamento dinamico (dynamic binding), tecnica con cui alcune procedure vengono collegate solo quando sono chiamate in esecuzione.*



**Modulo
oggetto
rilocabile**



**Modulo assoluto
o
eseguibile**

