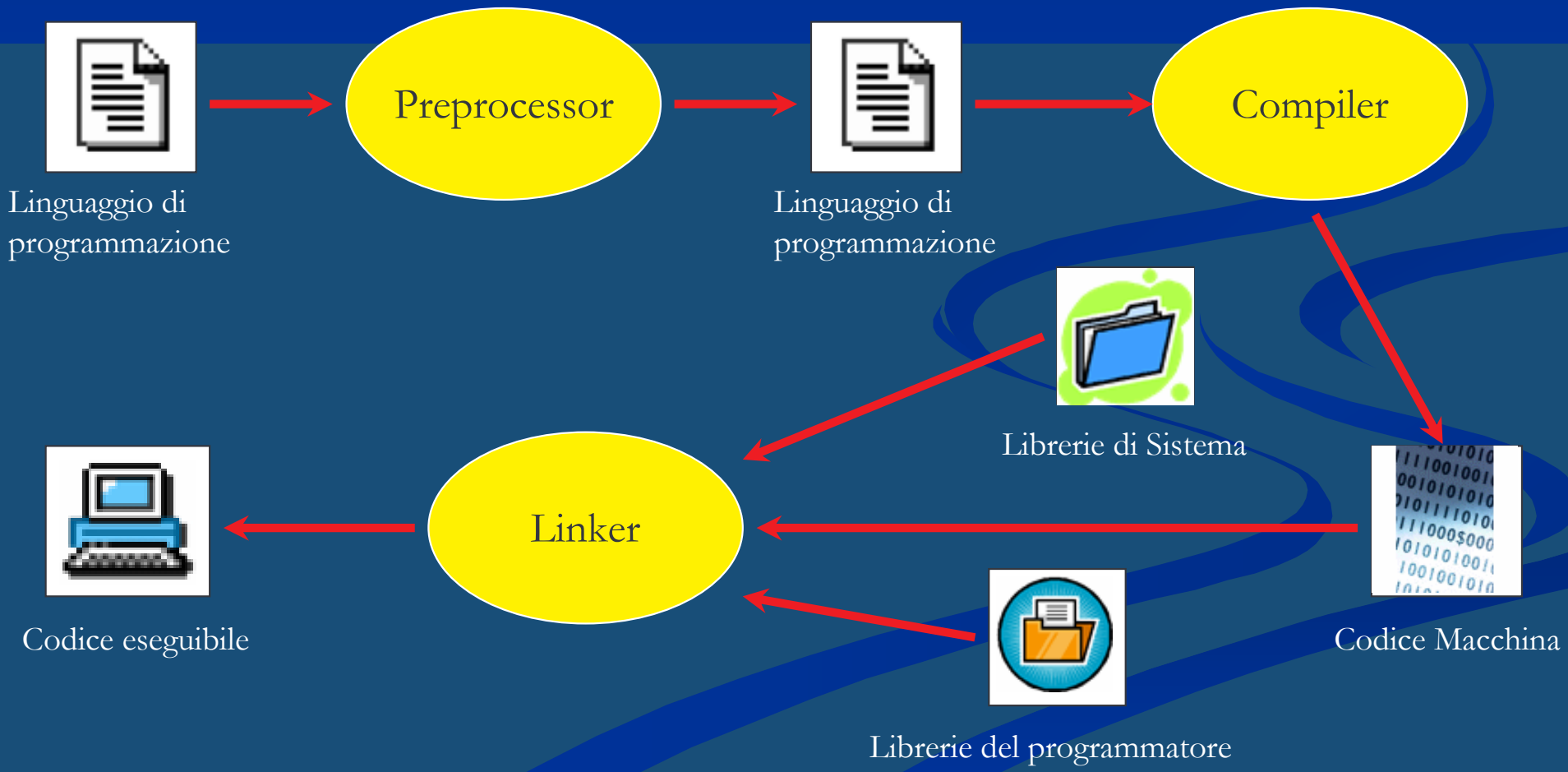
Dal linguaggio all'eseguibile

La generazione di un programma eseguibile a partire dal codice scritto in linguaggio di programmazione è un processo suddiviso in diversi passi



Il Preprocessore

Un ***preprocessor*** è un programma (o una porzione di programma) che effettua sostituzioni testuali sul **codice sorgente** di un programma.

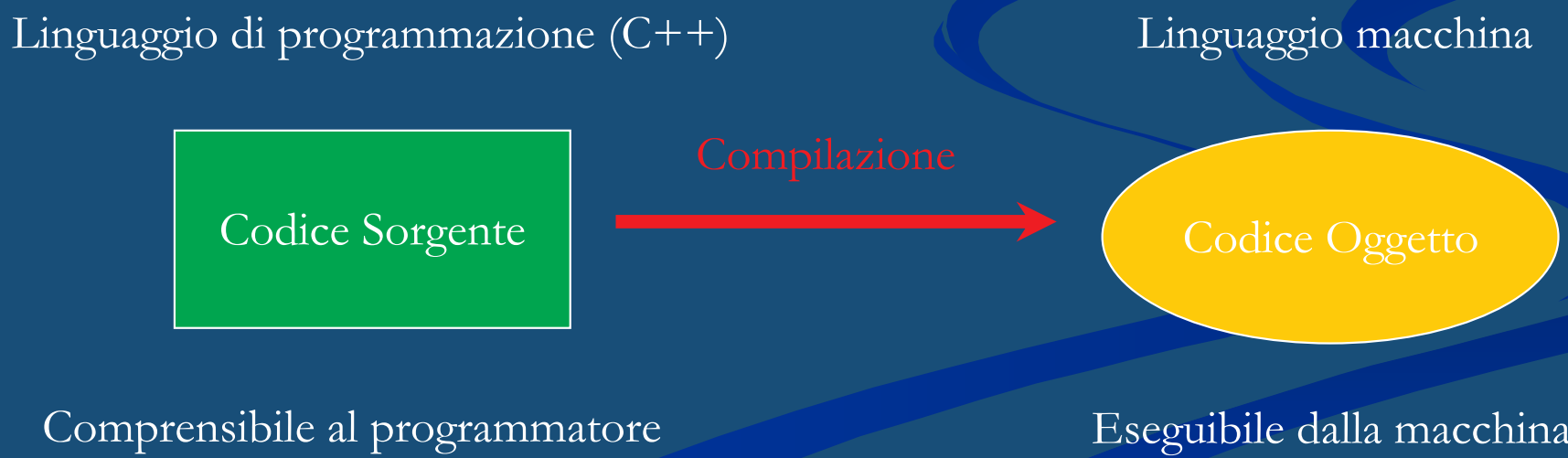
I più comuni tipi di sostituzioni sono:

- l'espansione di macro
- l'inclusione di altri file
- la selezione condizionale.

Le istruzioni inserite nel codice per essere elaborate dal preprocessore sono solitamente dette *direttive*: nel preprocessore del C/C++, le direttive sono righe che iniziano con il carattere "#".

Il Compilatore

Un ***compiler*** è un programma traduttore, impiegato per produrre **codice oggetto** (in linguaggio macchina) a partire da **codice sorgente** scritto in un dato linguaggio di programmazione (nel nostro caso il C++) di livello più alto. Questo processo si chiama **compilazione**.



Codice Sorgente

Il **codice sorgente** (spesso abbreviato **sorgente**) è un insieme di istruzioni e dati, utilizzati per implementare un algoritmo in **codice macchina**, ossia per costruire un programma eseguibile per computer.

Codice Oggetto

Il **codice oggetto** (o file oggetto) è la traduzione del **codice sorgente** in linguaggio macchina (binario), comprensibile solo dall'elelaboratore.

I codici oggetto si trovano normalmente raggruppati in ***librerie*** che contengono una serie di funzioni strettamente imparentate tra di loro, ad esempio una serie di operazioni matematiche.

Il codice oggetto è composto normalmente da codice eseguibile, più una serie di informazioni che permettono al **linker** di unirlo, se richiesto, con altri codici oggetto per generare un programma funzionante.

Il compilatore: schema di funzionamento

- **Analisi Lessicale:** il programma viene suddiviso in stringhe note, come ad esempio parole chiave del linguaggio (*while*, *for*), costanti e nomi di variabili, i cosiddetti *token*.
- **Analisi Sintattica e Semantica: i**n questa fase il compilatore crea un albero sintattico partendo dai *token*. L'albero è costruito in base a una grammatica che specifica le possibili sequenze di *token* accettate del linguaggio.
- **Generazione del Codice: d**all'albero sintattico e dalla tabella dei simboli vengono prese le informazioni per generare il codice eseguibile.

In generale, i compilatori sono in grado di riconoscere alcune classi di errori presenti nel programma, e in alcuni casi di suggerire in che modo correggerli.

Errori di Compilazione

Il compilatore può segnalare due tipi di errori: gli *error* ed i *warning*.

La presenza anche di un solo *error* in compilazione impedisce sempre l'invocazione del linker.

I *warning*, al contrario, non arrestano il processo e viene pertanto prodotto comunque un file eseguibile. Essi riguardano situazioni di ambiguità che il compilatore può risolvere basandosi sui propri standard, ma che è opportuno segnalare al programmatore, in quanto essi potrebbero essere la manifestazione di situazioni non desiderate, quali, ad esempio, errori di logica: alcuni messaggi di warning possono essere tranquillamente ignorati a ragion veduta.

Il Linker

Un ***linker*** è un programma che prende uno o più moduli contenti codici oggetto, generati dal compilatore, e li assembla in un singolo programma eseguibile.

I moduli contengono codice macchina e informazioni utili al linker, che consistono principalmente in definizioni di *simboli*:

- *Simboli Definiti o Esportati*: si tratta di funzioni o variabili che sono presenti nel codice oggetto e che possono essere disponibili per altri moduli.
- *Simboli Non Definiti o Importati*: sono funzioni o variabili che sono chiamate o referenziate da questo modulo, ma non sono definite internamente (perché contenute in altri moduli o nelle *librerie di sistema*).

Errori di Linking

Anche se il codice che compone il nostro programma non presenta errori (ed infatti il compilatore ha prodotto i vari moduli di codice oggetto) il processo di produzione delle eseguibile può essere interrotto da errori di linking:

- *Undefined Symbol* - il linker non riesce a *risolvere* tutti i *simboli non definiti*. In pratica non riesce a trovare il modulo che contiene la definizione di una variabile o l'implementazione di una funzione che sono state utilizzate nel codice.
- *Ambiguous Symbol* - il linker non riesce a stabilire quale tra i *simboli esportati*, presenti nei vari moduli che sta analizzando, corrisponde ad un particolare *simbolo importato*.