

ARCHITETTURE E LINGUAGGI PER LA PROGRAMMAZIONE CONCORRENTE

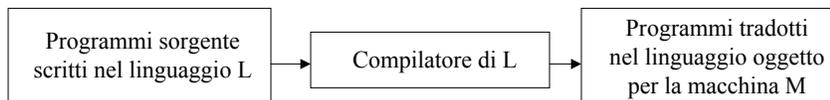
Linguaggi per la programmazione concorrente

- Disponendo di macchine concorrenti (in grado di eseguire più processi sequenziali contemporaneamente) e di un *linguaggio di programmazione con il quale descrivere algoritmi non sequenziali*, è possibile scrivere e far eseguire programmi concorrenti.
- L'elaborazione complessiva può essere descritta come un insieme di *processi sequenziali asincroni interagenti*

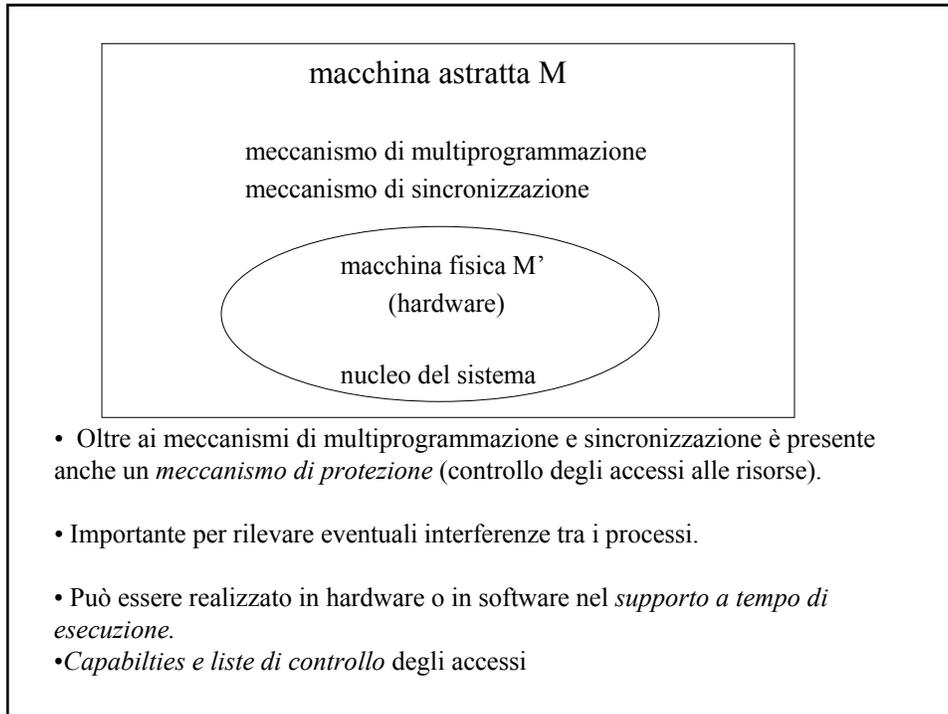
Proprietà di un linguaggio per la programmazione concorrente

- Contenere appositi costrutti con i quali sia possibile dichiarare moduli di programma destinati ad essere eseguiti come *processi sequenziali distinti*.
- Non tutti i processi costituenti un'elaborazione vengono eseguiti *contemporaneamente*. Alcuni processi vengono svolti se, dinamicamente, si verificano particolari condizioni. E' quindi necessario specificare quando un processo deve essere *attivato e terminato*.
- Occorre che siano presenti strumenti linguistici per specificare le *interazioni* che dinamicamente potranno aversi tra i vari processi

Architettura di una macchina concorrente



- Difficilmente M ha una struttura fisica con tante unità di elaborazione quanti sono i processi da svolgere contemporaneamente durante l'esecuzione di un programma concorrente.
- M è una macchina astratta ottenuta con tecniche software (o hardware) basandosi su una macchina fisica M' molto più semplice (con un numero di unità di elaborazione molto minore del numero dei processi)



- Il nucleo corrisponde al supporto a tempo di esecuzione del compilatore di un linguaggio concorrente.
- Nel nucleo sono sempre presenti due funzionalità base:
 - meccanismo di multiprogrammazione
 - meccanismo di sincronizzazione e comunicazione
- Il primo meccanismo è quello preposto alla gestione delle unità di elaborazione della macchina M' (unità reali) consentendo ai vari processi eseguiti sulla macchina astratta **M di condividere** l'uso delle unità reali di elaborazione
- Il secondo meccanismo è quello che estende le potenzialità delle unità reali di elaborazione, rendendo disponibile alle unità virtuali strumenti mediante i quali due o più processi possono sincronizzarsi e comunicare.

Architettura di M:

Due diverse organizzazioni logiche

Gli elaboratori di M sono collegati ad un'unica memoria principale (sistemi multiprocessore)

Gli elaboratori di M sono collegati da una sottorete di comunicazione, senza memoria comune (sistemi distribuiti).

Le due precedenti organizzazioni logiche di M definiscono *due modelli di interazione tra i processi*.

Modello a memoria comune, in cui l'interazione tra i processi avviene su oggetti contenuti nella memoria comune.

Modello a scambio di messaggi, in cui la comunicazione e la sincronizzazione tra processi si basa sullo scambio di messaggi sulla rete che collega i vari elaboratori