MODELLO A MEMORIA COMUNE

Aspetti caratterizzanti

• Ogni applicazione *viene strutturata* come un insieme di componenti, suddiviso in due sottoinsiemi disgiunti: *processi* (componenti attivi) e *risorse* (componenti passivi).

• **Risorsa**. Qualunque oggetto, fisico o logico, di cui un processo necessita per portare a termine il suo compito.

• Le risorse sono raggruppate in *classi*; una classe identifica l'insieme di *tutte e sole* le operazioni che un processo può eseguire per operare su risorse di quella classe.

• Il termine risorsa si identifica con quello di *struttura dati* allocata nella memoria comune.

• **Risorsa privata** di un processo P (o **locale** a P).
P è il *solo* processo che può eseguire operazioni sulla risorsa.

• **Risorsa comune (o globale)** è una risorsa su cui *più processi* possono operare.

• In un modello a memoria comune i processi interagiscono *esclusivamente* operando su risorse comuni (competizione e cooperazione).

• **Meccanismo di controllo degli accessi**: è necessario definire quali processi ed in quali istanti possono *correttamente* accedere alla risorsa.

• **Allocatore o gestore di una risorsa R:** entità che ha il compito di definire in ogni istante t l'insieme *SR(t)* dei processi che possono accedere ad R in quell'istante.

Risorsa allocata staticamente.

•Il gestore di R definisce l'insieme *SR all'istante To* (istante iniziale dell'elaborazione) senza modificarlo durante l'elaborazione.

• Il gestore della risorsa è il *programmatore* che in base alle *scope rules* del linguaggio stabilisce quale processo possa vedere e quindi elaborare la risorsa.

• Il compito di controllare gli accessi è svolto dal *compilatore* che, secondo le regole di visibilità, assicura che *soltanto* i processi ai quali la risorsa è stata allocata possano accedervi.

Risorsa allocata dinamicamente

• L'insieme SR è variabile nel tempo.
a)Risorsa dedicata. Sr(t) contiene *al più un processo*
b)Risorsa condivisa. Sr(t) contiene *più processi* contemporaneamente

• Il gestore della risorsa opera a *tempo di esecuzione*. Nel modello a memoria comune il gestore è una *risorsa* (nel modello a scambio di messaggi è un *processo*).

•Sr(t) è inizialmente vuoto. Per operare sulla risorsa il processo *deve chiedere il permesso al gestore.*
Il gestore può accettare, ritardare o rifiutare la richiesta .

• *Schema logico* seguito dal processo: richiesta della risorsa, uso e rilascio del diritto di accedere.

	risorsa dedicata	risorsa condivisa
risorsa allocata staticamente	privata	comune ai processi cui la risorsa è allocata
risorsa allocata dinamicamente	comune	comune

• Una risorsa allocata staticamente e dedicata ad un solo processo è una *risorsa privata*.
• In tutti gli altri casi le risorse sono *comun*i, o perché condivise tra più processi o perché dedicate, ma *dinamicamente*, a processi diversi in tempi diversi.

• Ogni gestore viene programmato come una risorsa astratta.

•E' costituito da:
- •*una struttura dati* usata per mantenere aggiornato lo stato della risorsa gestita;
- •*alcune procedure* che costituiscono le operazioni utilizzate dai processi per richiedere o rilasciare il diritto di operare sulla risorsa gestita

• La struttura dati e le procedure sono programmate in modo da soddisfare la *strategia di allocazione della risorsa*.

• Il gestore è una *risorsa condivisa e allocata staticamente*

• **Cooperazione tra processi**
Sono necessari *costrutti* che consentano la *sincronizzazione* degli accessi ad una risorsa comune tramite la quale i processi si scambiano informazioni.
Tali costrutti verranno usati nella *programmazione delle procedure* di accesso alla struttura dati della risorsa.

Competizione tra processi
Si presenta solo nel caso di risorse *condivise.*
Sono necessari costrutti per programmare le procedure di accesso in modo che il loro comportamento coincida con quello di *operazioni primitive*(da eseguirsi cioè in *mutua esclusione*)

• Per presentare *alcuni meccanismi linguistici* usati nella programmazione di interazioni tra processi si seguirà il seguente schema:

- Presentazione del meccanismo

- Esempi di uso del meccanismo

- Traduzione del meccanismo linguistico in termini di un *meccanismo primitivo* di sincronizzazione fornito dal supporto a tempo di esecuzione (nucleo)

• Il meccanismo primitivo di riferimento è quello **semaforico** con le due operazioni **wait** e **signal** (Dijkstra).