

Modello a scambio di messaggi

Definizioni

- **Modello a scambio di messaggi:** ogni processo può accedere esclusivamente alle risorse allocate nella propria memoria (virtuale) locale; ogni risorsa è accessibile ad un solo processo, eventuali altri utilizzatori chiederanno a questo processo di operare con la risorsa e restituire i risultati.
- **Gestore di risorsa:** processo server.
- **Modalità di interazione:** scambio di messaggi tra processi, attraverso un canale.
- **Canale:** collegamento logico mediante il quale due processi comunicano.
 - *L'astrazione canale è fornita dal nucleo;*
 - *Gli strumenti linguistici di alto livello per specificare e utilizzare i canali deve essere fornita dal linguaggio di programmazione.*

Parametri di un canale

- **Tipologia di canale:** direzione del flusso dati che il canale può trasferire:
 - *Monodirezionali*
 - *Bidirezionali* (es. Client/Server)
- **Designazione del canale:** nome del canale e dei processi sorgente e destinatario della comunicazione:
 - *Link* (canale simmetrico 1:1)
 - *Port* (canale asimmetrico 1:N)
 - *Mailbox* (canale asimmetrico N:M)
- **Tipo di sincronizzazione tra i processi comunicanti**
 - Comunicazione *asincrona*
 - Il mittente continua a eseguire subito dopo l'invio
 - Non ci sono informazioni per sapere lo stato del mittente e la send non è un punto di sincronizzazione
 - Richiede un *buffer illimitato* (o eccezione al riempimento)
 - Se si vuole mantenere la semantica, il supporto a tempo di esecuzione deve bloccare la *send* in caso di coda piena.
 - Comunicazione *sincrona*
 - Il primo che fa send o receive si sospende finché l'altro non ha eseguito l'operazione duale;
 - Non serve un buffer, perché un messaggio viene inviato solo dopo che il precedente è stato ricevuto;
 - Ogni messaggio dà informazioni sullo stato attuale del mittente;
 - Esempio: send e receive dell'acknowledge
 - Comunicazione con *sincronizzazione estesa*
 - Il mittente attende al risposta al proprio messaggio (eseguo operazione e attendo risultato)
 - Analogia semantica con la chiamata di procedura
 - Modello C/S
 - Punto di sincronizzazione

Meccanismo di ricezione ideale

- Consente al server di verificare lo stato (disponibilità di messaggi) su più canali;
- Abilita la ricezione di un messaggio da un qualsiasi canale pronto;
- Sospende il processo se tutti i canali sono vuoti.

Comando con guardia

Stati di una guardia

- Guardia fallita: espressione booleana con valore false
- Guardia ritardata: espressione booleana vera, ma receive bloccata
- Guardia valida: espressione booleana vera e receive pronta

Strutture a comando alternativo

- Se ci sono una o più guardie valide, viene scelta una ed eseguita sia la receive che l'istruzione, *poi il comando termina*;
- Se ci sono guardie ritardate e nessuna valida, il processo si sospende in attesa di un messaggio che porti una guardia in stato valido;
- Se sono tutte fallite, terminazione.

Struttura a comando ripetitivo

- Se ci sono una o più guardie valide, viene scelta una ed eseguita sia la receive che l'istruzione, *poi il comando ricomincia*;
- Se ci sono guardie ritardate e nessuna valida, il processo si sospende in attesa di un messaggio che porti una guardia in stato valido;
- Se sono tutte fallite, terminazione.

Realizzazione primitive

- Prima si realizzano quelle asincrone utilizzando gli strumenti del nucleo: sono più primitive e quelle sincrone si possono tradurre in termini di primitive asincrone
- **Primitive asincrone:**
 - Buffer illimitato sulla porta = messaggi in lista concatenata;
- **Primitive sincrone:**
 - Tramite semafori
 - Tramite primitive asincrone
 - Tramite primitive del nucleo

Architetture distribuite

- **Sistemi operativi distribuiti (DOS):** insieme di nodi omogenei dotati dello stesso sistema operativo (stesso nucleo);
 - Scopo: gestire tutte le risorse nascondendo all'utente la distribuzione in rete;
- **Sistemi operativi di rete (NOS):** insieme di nodi eterogenei, con sistemi operativi diversi e autonomi;
 - Ogni nodo fornisce servizi a clienti remoti (uso di *socket*), la trasparenza è ottenuta mediante middleware.

Comunicazione con sincronizzazione estesa

Tipologie

- **Chiamata di procedura remota:** un processo richiede un servizio e rimane sospeso fino al completamento del risultato;
 - Rendez-vous esteso = i processi rimangono sincronizzati durante l'esecuzione del servizio;
 - Analogia semantica con una chiamata di funzione (eseguita però remotamente)
 - Per ogni operazione richiedibile, il server dichiara una procedura e per ogni richiesta si crea un nuovo thread che la esegue; la sincronizzazione non è automatica, bisogna usare semafori per la concorrenza tra i thread.
- **Rendez vous:** meccanismo analogo a chiamata di procedura remota, ma l'operazione è un insieme di istruzioni in un punto qualsiasi del processo servitore, che utilizza l'istruzione di *accept* per sospendersi fino all'arrivo di una richiesta, dopodiché esegue l'insieme di istruzione e manda il risultato al chiamante.
 - Non ci sono thread, quindi la sincronizzazione è automatica
 - Primitiva *accept* come *receive*, si inserisce in una struttura ripetitiva di comandi con guardia:
 - Se non ci sono richieste, sospensione
 - Se le richieste sono multiple per un servizio, vanno in coda
 - Ad ogni servizio possono corrispondere più *accept* in funzione del punto di elaborazione
 - Schema asimmetrico 1:N
 - Se la scelta di servire o meno dipende anche dall'input, si spezza il tutto in due fasi: prima invio i parametri, poi la richiesta.
- **Linguaggio ADA:** adotta come metodo il rendez-vous, con introduzione dei *protected type* (simili ai *monitor*) e dell'istruzione *requeue*.
 - Comunicazione asimmetrica di tipo rendez-vous
 - Rendez vous avviene quando entrambi esprimono la volontà di eseguire l'operazione (*entry*)
 - Una parte che specifica le operazioni *entry* e una parte la realizza.

Corrispondenza tra monitor e processi servitori

Modello a memoria comune	Modello a scambio di messaggi
Risorsa: istanza di monitor	Risorsa: struttura dati locale a un processo server
Accesso tramite funzione del monitor	Accesso tramite porta del server
Tipo di parametri della funzione	Tipo della porta
Tipo restituito dalla funzione	Tipo della porta di risposta
Funzione del monitor	Ramo (comando con guardia) della struttura a controllo ripetitivo
Condizione di sincronizzazione di una funzione	Espressione logica della guardia
Chiamata di funzione	Invio di una richiesta sulla porta del server e ricezione del risultato
Esecuzione in mutua esclusione tra le chiamate del monitor	Scelta di uno dei rami con guardia valida
Corpo della funzione	Istruzione del ramo corrispondente della guardia