

messaggi-a2

Dato un sistema di supporto per message passing asincrono dotato delle chiamate `asend` e `arecv` scrivere un meccanismo a messaggi doppi `a2send` `a2recv` che operi nel modo spiegato qui di seguito. La `a2send` consente di spedire un messaggio in modo asincrono mentre la `a2recv` restituisce solo coppie di messaggi provenienti dallo stesso mittente. Quindi la `a2recv` deve sempre attendere due messaggi che rispettino le richieste del ricevente (mittente giusto, o ogni mittente se `**`) provenienti dallo stesso mittente prima di potersi sbloccare. Il problema è risolvibile senza fare uso di un processo server.

```

// Mi "creo" una struttura dati che associ ad ogni processo una
// coda di messaggi ricevuti
Map buffer: <process-id> -> Queue
Queue senders = new Queue();

a2send(Msg m) {
    asend(m);
}

// riceve una coppia di messaggi da un id specifico
// nota che mi sono "inventato" la possibilità di ritornare una
// coppia di valori (fra parentesi)
(Msg,Msg) a2receive(Pid id)
{
    while (buffer[id].size < 2) {
        // resto in questo ciclo fino a quando non ci sono
        // almeno due messaggi
        msg = areceive(*);
        buffer[msg.sender].add(msg);
        if (buffer[msg.sender].size % 2 == 0)
            // il sender e' pronto a ricevere una coppia di messaggi
            // lo aggiungo
            senders.add(msg.sender);
    }
    // rimuovo la prima occorrenza di id dalla coda
    senders.remove(id);

    // Rimuovo effettivamente due messaggi e li ritorno.
    return (buffer[id].remove(), buffer[id].remove());
}

// Questo riceve da tutti
(Msg,Msg) a2receive(*)
{
    while (senders.size == 0) {
        // resto in questo ciclo fino a quando non arriva una coppia
        // di messaggi
        msg = areceive(*);
        buffer[msg.sender].add(msg);

        if (buffer[msg.sender].size % 2 == 0)
            // il sender e' pronto a ricevere una coppia di messaggi
            // lo aggiungo
            senders.add(msg.sender);
    }
    id = senders.remove();
    return (buffer[id].remove(), buffer[id].remove());
}
}

```