

giugno 2002

## Tema di informatica

La Zeta & C. è un'azienda specializzata nel controllo di macchinari tramite l'uso di piccoli elaboratori, ognuno responsabile di una parte del macchinario da automatizzare. Gli elaboratori sono connessi tra loro tramite una rete locale e ogni macchina è identificata univocamente all'interno della rete tramite un numero di nodo. Il sistema operativo utilizzato è stato progettato dall'azienda stessa ed è caratterizzato da uno strato software di rete che mette a disposizione di ogni elaboratore una serie di primitive di comunicazione che consentono di scambiare informazioni tra tutte le macchine. In particolare, da ciascuna macchina è possibile inviare a qualunque altra un pacchetto dati tramite la primitiva:

`Send([in]To, [in]Type, [in]Packet)`

dove `To` è il numero di nodo del destinatario, `Type` è la classe del pacchetto e `Packet` è una sequenza di 1024 byte che trasporta l'informazione scambiata. La primitiva `Send` è non bloccante e non affidabile; un errore sulla rete che non consenta di portare a termine una trasmissione non viene quindi segnalato al mittente. Analogamente, la primitiva:

`Receive([in]Type, [out]From, [out]Packet)`

permette a una macchina di ricevere una sequenza di byte (`Packet`) relativa ad un pacchetto di classe `Type`. Le classi `Type` non sono fissate a priori e dovranno, pertanto, essere definite dal sistemista e utilizzate per realizzare astrazioni di comunicazione di livello più alto. La primitiva `Receive` è non bloccante: i pacchetti in arrivo vengono temporaneamente memorizzati dal sistema operativo in un buffer che può contenere fino a  $N$  pacchetti ( $N$  fissato dal progettista). Eventuali pacchetti oltre l' $N$ -esimo vengono automaticamente scartati.

La rete può non mantenere l'ordine dei pacchetti inviati: detto  $A$  un pacchetto inviato nell'istante  $t_m$  e giunto nell'istante  $t_p$ , e detto  $B$  un pacchetto inviato nell'istante  $t_n > t_m$  e giunto nell'istante  $t_q$ , si può solo dire che  $t_p \neq t_q$ .

Per la Zeta & C. la trasmissione non affidabile di 1024 bytes per volta non è sufficiente poiché intende realizzare controlli industriali avanzati in cui c'è la necessità di trasferire grandi quantità di dati critici in maniera assolutamente affidabile (serie di misure di sensori, verifica del controllo di una parte del macchinario, ecc.). Pertanto si richiede al candidato di:

- progettare uno strato software che permetta una trasmissione affidabile dei pacchetti definendo un protocollo che garantisca che i pacchetti arrivino correttamente dall'altra parte, ritrasmettendo i pacchetti che si ipotizza non siano arrivati;
- progettare un ulteriore strato che utilizzi i servizi forniti dallo strato di cui al punto precedente e che consenta di realizzare l'astrazione dello stream (flusso continuo) affidabile di dati.

Poiché il software progettato deve lavorare in un ambiente real-time è necessario che la soluzione adottata sia semplice in modo da introdurre un overhead minimo.

Per la progettazione si segua la seguente traccia:

1. Definire i requisiti e l'architettura logico-funzionale degli strati che permettono la trasmissione e descrivere brevemente i modelli di carattere generale adottati.
2. Descrivere i singoli componenti software necessari alla costruzione del sistema, specificando le funzionalità offerte agli altri componenti.
3. Definire le metodologie di test e di collaudo degli strati software progettati.
4. Codificare gli strati precedentemente descritti tramite linguaggi di alto livello come C, C++, Java possibilmente utilizzando un approccio Object-Oriented o comunque tale da assicurare alla soluzione le necessarie modularità, manutenibilità e estendibilità incrementale.