POLITECNICO DI MILANO





I Socket

Laboratorio Software 2008-2009
M. Grotto R. Farina

Sommario

1. **Applicazioni Distribuite**
   - Introduzione
   - Interfacce e protocolli
2. **I Socket**
   - Descrizione
   - Stile di comunicazione
   - Namespace e protocollo
   - Include e system call
   - Creazione e chiusura
   - Dettagli sulle funzioni
   - Connessione
   - Socket locali
   - Internet domain socket
   - Socket pairs

Applicazioni distribuite

Introduzione

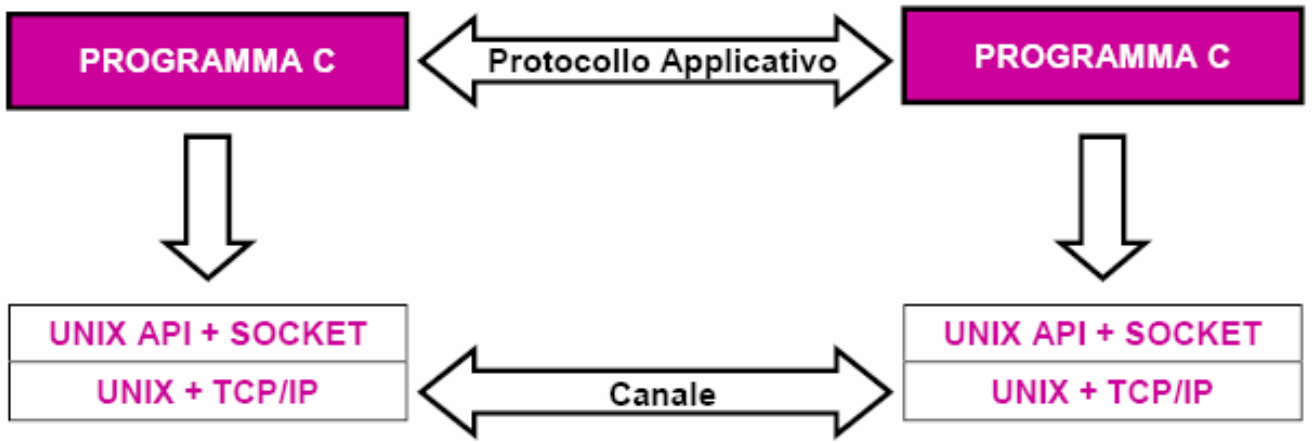
- Applicazione distribuita:
  - E' costituita da un insieme di programmi che vengono generalmente eseguiti su macchine diverse
  - I vari programmi cooperano attraverso una rete di calcolatori
- Le applicazioni distribuite richiedono la capacità di comunicare attraverso una rete
- Per la comunicazione i programmi di una applicazione distribuita si appoggiano a un insieme di servizi tipici forniti:
  - Dal sistema operativo
  - Dal software di rete

Applicazioni distribuite

Interfacce e protocolli

- Il programmatore utilizza funzioni di libreria
  - L'insieme delle funzioni di base prende il nome di Application Program Interface o API
  - un socket è un'astrazione software progettata per poter utilizzare delle API standard e condivise per la trasmissione e la ricezione di dati attraverso una rete

- La configurazione di riferimento di una applicazione distribuita basata su TCP/IP e socket è la seguente:



I Socket

Descrizione

- Mezzo di comunicazione BIDIREZIONALE tra processi residenti sulla stessa macchina o su macchine differenti
- I dati trasmessi sono suddivisi in pacchetti
- I socket sono rappresentati da file descriptor
  - È possibile operare con le primitive di I/O classiche
- Tre parametri caratteristici
  - Stile di comunicazione
  - Namespace
  - Protocollo

I Socket

Stile di comunicazione

- Stabilisce come vengono trattati i dati trasmessi
- Connection
  - Garantisce la ricezione di TUTTI i pacchetti nell'ordine esatto di spedizione
- Datagram
  - Pacchetti persi o riordinati a causa della rete
  - Ogni pacchetto va etichettato con l'indirizzo del destinatario

I Socket

Namespace e protocollo

- Namespace
  - Che forma hanno gli indirizzi
    - Ogni indirizzo identifica un'estremità di un socket
  - Nomi locali = nomi di file
  - Nomi Internet = indirizzo IP + porta
    - La porta identifica il particolare socket
- Protocollo
  - Modalità di trasmissione
  - TCP/IP: il più famoso per Internet
  - AppleTalk
  - UNIX local communication protocol

I Socket

Include e system call

- `#include <sys/types.h>`
- `#include <sys/socket.h>`
- `socket();` crea un socket
- `close();` distrugge un socket
- `connect();` crea una connessione tra due socket
- `bind();` assegna un indirizzo ad un socket server
- `listen();` configura un socket per accettare connessioni
- `accept();` accetta una connessione e crea un nuovo socket per la comunicazione
- `send();` per spedire
- `recv();` per ricevere

I Socket

Creazione e chiusura

- Si usa `socket();` per la creazione
  - Namespace
    - **`PF_LOCAL`**, **`PF_UNIX`**, **`PF_INET`** (`PF` = protocol families)
  - Stile di comunicazione
    - **`SOCK_STREAM`**, **`SOCK_DGRAM`**
  - Terzo parametro: protocollo
    - Meccanismo di basso livello per trasmissione e ricezione
    - Dipendente dalla coppia namespace-stile: assegnare 0 è la scelta migliore
  - Restituisce un file descriptor
- `close();` per chiudere il socket

I Socket

Dettagli sulle funzioni

- `bind();`
  - Socket file descriptor
  - Puntatore ad una struttura per l'indirizzo del socket
  - Lunghezza in bytes della struttura
- Quando un indirizzo viene collegato ad un socket di tipo connection tramite `bind();` è necessario invocare `listen();` per indicare che è un server
  - File descriptor e dimensione della coda
- `accept();` per accettare una connessione
  - File descriptor e puntatore ad una struttura `sockaddr` riempita con i dati del client
  - Crea un nuovo socket e restituisce il file descriptor

I Socket

Connessione

- Il server
  - Crea un socket
  - Invoca `bind();` per assegnargli un indirizzo
  - Invoca `listen();` per consentire connessioni al socket
  - Invoca `accept();` per accettare connessioni
  - Quando una connessione è accettata viene creato un secondo socket per il trasferimento dati, lasciando il primo in ascolto
- Il client invoca `connect();` passando l'indirizzo del socket al quale effettuare la connessione
  - Socket da usare e indirizzo a cui connettersi

I Socket

Socket locali

- Namespace locale: `PF_LOCAL` o `PF_UNIX`
- Formato degli indirizzi (`struct sockaddr_un`)
  - Il campo `sun_family` impostato a `AF_LOCAL`
  - Il campo `sun_path` specifica il percorso del file da usare
    - Lunghezza massima di 108 bytes
    - Il processo deve avere permessi di scrittura nella directory per poter creare nuovi file
    - Per la connessione al socket un processo deve avere i permessi di lettura sul file

I Socket

Socket locali

- Si usa la macro `SUN_LEN` per calcolare la lunghezza in bytes della struttura `sockaddr`
- Solo per processi locali
  - Impossibile l'uso anche se computer differenti condividono lo stesso filesystem
- Si invoca `unlink();` sul file descriptor quando si è finito di usare il socket

Esempio

`socket-server.c`

`socket-client.c`

I Socket

Internet domain socket

- Namespace Internet: `PF_INET`
- Formato degli indirizzi (struct sockaddr_in)
  - `sin_family` impostato a AF_INET
  - `sin_addr` memorizza l'indirizzo IP della macchina come intero a 32 bit
    - `gethostbyname();` per convertire indirizzi IP nella notazione dotted o nomi in interi a 32 bit

      Restituisce un puntatore ad una struttura `hostent` il cui campo `h_addr` contiene l'indirizzo IP
  - `sin_port` memorizza il numero di porta
    - Distingue i diversi socket su una macchina
    - Funzione `htons();` per convertire il numero di porta in network byte order

I Socket

Socket pairs

- Pipe limitate dal fatto che la comunicazione è unidirezionale e solo tra processi in relazione
- `socketpair();` crea una coppia di socket connessi
  - Comunicazione bidirezionale tra processi in relazione
  - Primi tre parametri identici a quelli per l'invocazione di socket (dominio, stile, protocollo)
    - `PF_LOCAL` come dominio
  - Parametro addizionale: array di interi di dimensione 2
    - File descriptors dei socket
    - Simile alle pipe