

## ***KERNET (Network Transaction Service)***

Nell'ambito industriale, il settore informatico è sicuramente il più soggetto a costanti progressi e innovazioni: tale fenomeno comporta un palese vantaggio economico, fornendo sempre nuove tecnologie a costi progressivamente più contenuti. Tuttavia questa stessa frenetica "escalation" tecnologica rende rapidamente obsoleti prodotti software concettualmente validi e al passo coi tempi, in particolar modo nell'ambito di sistemi gestionali distribuiti.

Negli ultimi anni abbiamo assistito a un progressivo orientamento del mercato verso elaborazioni distribuite e sistemi aperti, basati sull'approccio Client/Server. Sfruttando i benefici di tale concezione i sistemi aperti consentono soluzioni multivendor ad elevata interoperabilità tra diverse applicazioni e ad elevata portabilità tra piattaforme differenti; inoltre i sistemi aperti offrono la libertà di scegliere e di aggiornare le piattaforme ospiti in base alle reali necessità e di ripartire le elaborazioni su siti diversi.

Pertanto numerose aziende, che verso la fine degli anni 80 hanno investito pesantemente nella realizzazione di applicativi, si trovano oggi con un prodotto progettato per piattaforme ormai superate, ma che mantengono la loro totale validità applicativa. Per evitare di dover investire nuovamente nella realizzazione di un nuovo prodotto o nella modifica quasi integrale del precedente, i produttori di tali applicativi possono orientarsi verso alternative migliori: una di queste è **KERNET**.

## ***Kernet è ...***

**Kernet** (KERnel for NETwork) è un tool per la realizzazione in house delle funzioni di un TP Monitor integrate nella struttura applicativa.

Questo prodotto si orienta a piattaforme Client/Server costituite da:

- sistemi Unix o MS Windows NT per la componente server
- sistemi MS Windows, Unix o MS Windows NT per la componente client

e viene rilasciato sotto forma di librerie che implementano i seguenti livelli funzionali:

- **Transaction Dispatching & Delivery**, con il supporto di un protocollo dedicato di sessione (KNP) implementato sui socket TCP/IP;
- **Resource Allocation**, per il controllo e l'allocazione delle risorse di sistema e la sincronizzazione dei task applicativi a livello server;
- **Security**, con il supporto di funzioni di crittografia ;
- **Network Presentation**, per la gestione di strutture dati e di rete indipendente dalle piattaforme operative dei componenti client e server.

Quando il software **KERNET** viene integrato in una struttura applicativa preesistente, risulta abbastanza facile implementare una struttura a crescita scalare, che partendo da semplici funzioni di connessione evolve sino a coprire funzioni più complesse di gestione, monitoring e amministrazione della struttura applicativa.

Il software **KERNET** è stato progettato per reti di tipo TCP/IP e piattaforme di sistema Unix e MS Windows; tuttavia è possibile estenderlo ad altre piattaforme che supportano interfacce socket/BSD e Posix-like. L'intrinseca validità di **KERNET** sta proprio nel fatto che la maggior parte delle piccole e medie aziende adottano oggi reti di tipo TCP/IP con server Unix o Windows-NT e client Windows, ma in qualsiasi momento è possibile adattarlo a esigenze più particolari e specifiche.

## ***Kernet dove ...***

KerNet è stato utilizzato come supporto progettuale e run-time nei seguenti progetti:

- porting del pacchetto applicativo gestionale client/server **Strega** di **Infoter S.p.A.** da ambienti proprietario a piattaforme aperte Unix e Windows NT su reti TCP/IP;
- progetto **Stream** per l'automazione di una Struttura Tecnica Delegata nel Corporate Banking per conto di **CredemTel S.p.A.**;
- progetto **LongEye** per conto di **Società Autostrade S.p.A.**: gestione di stazioni remote per rilevamento veicoli fermi su tratti di galleria; gestione e trasporto dello stream video generato dalle telecamere di sorveglianza;
- progetto **SAMOT**, in collaborazione con **Project Automation S.p.A.**, per la realizzazione di una infrastruttura architeturale per progetti di gestione traffico e semafori;
- progetto **KremDir** di **S.I.A.E.**, per la raccolta e la distribuzione di dati contabili e per la gestione del software dissemination.

## **Specifiche Tecniche del Prodotto**

### **KerNet Network Transaction Service**

Il protocollo KNP (KerNet Protocol) e' stato progettato per integrare il protocollo TCP a livello 5, e servire di base alla implementazione di architetture distribuite secondo il modello Client/Server, definendo un insieme di modalita' di dialogo tra processi client e server operanti sui nodi di una rete TCP/IP. In particolare il protocollo definisce:

- la **sessione**, come momento di gestione della comunicazione utente, fornendo gli elementi per identificare univocamente una sessione nell'ambito della vita di una rete;
- la **transazione**, come elemento atomico della comunicazione utente, fornendo gli elementi per identificare univocamente una transazione nell'ambito di una sessione;
- gestione della comunicazione a **messaggi**, e non a flusso continuo (record-boundary);
- gestione della sessione in termini di **sequenze protette** (bracketing) e di **data-token**;
- modalita' di trasferimento per **sequenze di messaggi**, permettendo una **segmentazione** dei messaggi applicativi di grandi dimensioni;
- gestione automatica dei **numeri di sequenza e di segmentazione** dei messaggi.

#### Processi Client-Kernet

Le primitive client del software KerNet permettono di accedere ai servizi applicativi dei Server-KerNet, realizzando il primo passo dell'accesso client/server. Nel loro complesso le primitive client permettono:

- creazione/chiusura di sessioni
- allocazione/rilascio delle risorse
- inizio/termine di transazioni
- invio di piu' messaggi e segmenti
- ricezione di messaggi e segmenti
- chiusura forzata di transazioni

Qualunque programma che opera in ambiente Windows, sia esso realizzato in C/C++, Visual Basic o altro, può diventare un client KNP.

#### Processi Server-Kernet

Lato server il software KerNet permette la facile realizzazione di servizi in linguaggio C (e C++) in una architettura multitasking (ovvero multiprocess e multithread per gli ambienti Unix/Posix e multithread per l'ambiente Windows-NT). Un servizio applicativo, inteso come unione di più tipi di transazioni applicative, viene realizzato per ogni processo client da un task server dedicato (processo o thread), con il concorso del codice applicativo specifico per i vari tipi di transazione. Su ricezione del messaggio di inizio di una transazione, il software KerNet schedula automaticamente il relativo codice applicativo nell'ambito del task server. Successivamente la transazione procede secondo lo schema di colloquio applicativo con il concorso del software KerNet.

Per definire le transazioni di rete di un dato servizio l'utente dispone di un piccolo compilatore (KNGEN) che permette la generazione automatica delle tabelle necessaria a personalizzare il software KerNet per lo specifico servizio applicativo.

L'interfaccia di programmazione del software KerNet é stata realizzata in modo da garantire la completa trasparenza rispetto all'ambiente ospite (Unix o Windows-NT).

## Interazioni tra task e processi in ambiente server

Il software KerNet permette di gestire le interazioni tra task e processi mediante una interfaccia virtuale indipendente dalla piattaforma che supporta tre tipi di interazione inter-process e/o inter-thread:

- set di mutex
- shared memory
- attended I/O channel

### *Mutex*

La gestione di pool comuni di risorse richiede la sincronizzazione di piú task dello stesso applicativo o processi legati a differenti applicativi su oggetti di sincronismo che permettono la protezione delle singole risorse del pool. Il mutex é definito in ambiente KerNet come un insieme di semafori binari detti mutex-item, che in un dato istante possono essere aperti o chiusi. Un task può richiedere la sincronizzazione su un mutex-item per ottenere l'accesso ad una risorsa critica, con la garanzia che una volta ottenuto l'accesso nessun altro task concorrente possa operare sulla risorsa. Al termine dell'utilizzo della risorsa il task rilascia il mutex-item relativo, permettendo in tal modo la schedulazione eventuale di un altro task che ha richiesto l'accesso alla risorsa ed é in attesa. Le primitive di accesso ad un mutex permettono: creazione, rimozione, apertura, chiusura, richiesta di accesso e rilascio di un mutex.

### *Shared Memory*

Il software KerNet supporta la gestione di pool di memorie condivise (shared-memory) tra piú task/processi con le primitive di inizializzazione, creazione, rimozione, apertura, chiusura, protezione, rilascio, verifica di una shared memory, calcolo del valore assoluto di un indirizzo relativo e viceversa.

Le shared-memory KerNet sono gestite con un insieme di shared-memory di sistema. L'insieme delle shared-memory di sistema viene gestito internamente con il supporto di una struttura che definisce la shared-memory KerNet.

### *Attended I/O Channel*

La gestione degli eventi di rete da parte di un task server KerNet può richiedere l'attivazione di uno o piú processi di elaborazione sotto il controllo di programmi applicativi dedicati. In tal caso il task server ha la necessità di comunicare con i processi di elaborazione creati sia in termini di messaggi che di sincronismo. L'utilizzo di canali di I/O realizzati con coppie di pipe half-duplex dedicate allo scambio con lo specifico processo di elaborazione risponde pienamente ad entrambi gli scopi. Il software KerNet supporta la gestione di canali di I/O bidirezionali dedicati e serviti (attended) da un processo di elaborazione specifico che opera sotto il controllo di un programma applicativo che scambia messaggi e sincronismi con il task padre per mezzo dei file di sistema 'stdin' (standard input) e 'stdout' (standard output). Il livello di servizio KerNet sugli attended-I/O-channel viene realizzato con le primitive di: creazione, rimozione, invio di messaggi/sincronismi, ricezione di messaggi/sincronismi su un attended-I/O-channel.