

9° Esercitazione (da svolgere):

Remote Procedure Call: inizializzazione strutture dati sul server

Si richiede la progettazione e la realizzazione del **servizio di prenotazione delle postazioni di una sala lettura** basato su RPC.

Le postazioni sono organizzate in file e colonne, e si prevedono tre tipi di prenotazioni: Bibliotecari (B), Professori (P), e Dottorandi (D). Si realizzino le operazioni per:

1. **prenotare una postazione**: questa operazione richiede all'utente il tipo di prenotazione, la fila e la colonna, quindi aggiorna la struttura dati contenente le prenotazioni;
2. **visualizzare lo stato della sala**: questa operazione visualizza l'attuale stato di occupazione della sala indicando, per ogni postazione, il tipo di prenotazione.

La struttura dati con lo stato della sala potrebbe essere la seguente tabella; ogni postazione libera viene etichettata con la lettera 'L', mentre le postazioni occupate sono opportunamente etichettate con 'B' o 'P' o 'D' a seconda del tipo di prenotazione.

	Col. 1	Col. 2	Col. 3	Col. 4	Col. 5	Col. 6	Col. 7
Fila 1	L	L	L	L	L	L	L
Fila 2	D	L	L	L	D	D	B
Fila 3	L	P	L	L	D	D	L
Fila 4	L	L	L	L	D	D	L
Fila 5	L	L	L	L	L	L	L
Fila 6	L	L	P	L	P	P	L
Fila 7	L	L	L	L	P	P	L

Utilizzando RPC sviluppare un'applicazione C/S per effettuare le operazioni remote proposte.

La procedura `prenota_postazione` accetta come parametro d'ingresso la struttura dati *Input* contenente il *tipo* di prenotazione ('B' o 'P' o 'D'), la *fila* e la *colonna* che si vogliono prenotare, e restituisce l'esito dell'operazione - 0 in caso di successo, -1 altrimenti, ad esempio, se posto che si vuole occupare è già prenotato.

La procedura `visualizza_stato` non ha alcun parametro di ingresso (ha come parametro di ingresso un puntatore a *void*), e restituisce la struttura dati *Sala* contenente lo stato di occupazione attuale della sala.

Più in dettaglio:

- il **client** realizza l'interazione con l'utente proponendo ciclicamente i servizi che utilizzano le due procedure remote e stampa a video gli esiti delle chiamate, fino alla fine del file di input da tastiera;
- il **server** implementa le procedure invocabili in remoto restituendo l'esito delle operazioni come indicato sopra.

Inizializzazione strutture dati

Abbiamo un problema di **inizializzazione della struttura dati** che mantiene lo stato di occupazione della sala sul server. La struttura deve essere inizializzata alla partenza del server impostando tutte le postazioni allo stato di default libero (stato 'L').

Nella generazione automatica del codice, lo stub del programma server (il main) viene prodotto da RPCGEN. Bisogna quindi considerare soluzioni per attuare l'operazione di inizializzazione, anche *eventualmente modificando i file generati in modo automatico*.

In particolare, si suggeriscono le seguenti possibilità:

- inserire una *procedura locale di inizializzazione nel file utente* contenente le procedure remote. L'invocazione di una qualsiasi procedura remota invocata dal client, provoca l'invocazione di tale procedura se e solo se l'inizializzazione non è già stata invocata (introduzione di una variabile di controllo **static** nel file utente).
- aggiungere all'interfaccia del server una *procedura remota di inizializzazione*. Tale procedura remota deve essere invocata dal client prima di qualsiasi altra procedura, e che deve essere eseguita dal server al più una volta.
- inserire il *codice di inizializzazione direttamente all'interno del main (stub del server)*;
- inserire una *procedura locale di inizializzazione nel file utente* contenente le procedure remote. Tale procedura deve essere invocata all'interno del main **prima** dell'invocazione della procedura `svc_run()` che mette il server in ascolto di nuove richieste. Per poter fare questo, è necessario dichiarare la procedura di inizializzazione come **extern** all'interno del file dello stub contenente stub e main (lato server).

Proposta di estensione

Si vuole estendere l'applicazione C/S realizzata in modo da gestire altre due operazioni per:

- **liberare una postazione**: questa operazione richiede all'utente la fila e la colonna della postazione da liberare, quindi aggiorna la struttura dati contenente le prenotazioni;
- **visualizzare le postazioni libere e occupate**: questa operazione visualizza le posizioni libere e quelle occupate oscurando le informazioni relative al tipo di prenotazione.

Per ogni richiesta di **liberazione di postazione**, il server riceve la fila e la colonna, aggiorna opportunamente la struttura dati, e invia la risposta al client: 0 in caso di successo, -1 in caso di problemi.

Per ogni richiesta di **visualizzazione delle postazioni libere e occupate**, il server riceve la richiesta e invia la struttura dati sostituendo tutte le occorrenze di prenotazione, cioè tutti i caratteri 'B', 'P', e 'D', con una generica etichetta occupato ('O').

Inoltre, si provino ad utilizzare **tutte le diverse modalità di inizializzazione** della struttura dati proposte.

Consegna

Potete inviare via email l'estensione ai docenti, per discutere la soluzione ed eventualmente pubblicarla sul sito del corso