8° Esercitazione (da svolgere):

Remote Procedure Call

Utilizzando RPC sviluppare un'applicazione C/S che consenta di effettuare le operazioni remote per:

- **contare le occorrenze di un carattere** in un file presente sul server remoto;

- **contare il numero file** presenti in un direttorio sul server remoto (nomi di direttorio inclusi).
  Come utile estensione, provare a realizzare il **conteggio dei file dell'intero sottoalbero remoto** con radice il direttorio indicato dal client. In questo caso, l'operazione procede ricorsivamente al conteggio dei file presenti in ogni sottodirettorio, fino ad arrivare ai direttori "foglia" che non contengono ulteriori sottodirettori, quindi restituisce il numero di file contati.

La procedura **`conta_occorrenze`** accetta come parametri d'ingresso il nome del file e il carattere, e restituisce un intero positivo con il numero di occorrenze contate, in caso di successo, -1 altrimenti.
La procedura **`conta_file`** accetta come parametro d'ingresso il nome del direttorio remoto, e restituisce un intero positivo con il numero di file contati, in caso di successo, -1 altrimenti.

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.

NOTA
Tenere bene presente le diverse modalità di rappresentazione delle stringhe:

```
string   s;
char *   s;
char     s[dim];
```

e ricordare i tipi consentiti dalle RPC: si rivedano le **definizioni XDR** dei tipi di dato.

Proposta di estensione

Get Multiple

Si vuole sviluppare un'applicazione C/S basata su RPC e su socket con connessione per il trasferimento di tutti i file di un direttorio remoto dal server al client (multiple get). In particolare, si vogliono realizzare due modalità di trasferimento, la prima con server attivo (*il server effettua l'accept*), la seconda con client attivo (*il client effettua l'accept*). Si dovranno realizzare un **client** e un **server**; l'utente, per ogni trasferimento, decide quale delle due modalità utilizzare.

Per entrambe le modalità, si prevede un'*interazione iniziale sincrona (realizzata con una richiesta RPC sull'oggetto remoto server)* per trasferire la **lista dei file** da inviare e **l'endpoint** (host e porta) **di ascolto**; quindi, una seconda fase di *trasferimento dei file* dal server al client *(realizzata con socket connesse)*.

In particolare, per la *prima modalità*, il metodo remoto accetta come argomento di ingresso il **nome del direttorio** e restituisce una struttura dati con l'**endpoint di ascolto del server** e la **lista con i nomi e la lunghezza di tutti i file** da trasferire.

Il **client** richiede ciclicamente all'utente il nome del direttorio da trasferire ed effettua la chiamata RPC, quindi stabilisce una connessione con il server remoto e riceve i file salvandoli sul direttorio locale.

Il **server** implementa il metodo RPC richiesto ed è realizzato come server *concorrente* e *parallelo*; per ogni nuova richiesta ricevuta il processo padre, dopo aver accettato la richiesta RPC, crea la socket di ascolto, attiva un processo figlio a cui affida il completamento del servizio richiesto e restituisce la struttura dati con la lista dei file e il proprio endpoint.

Per la *seconda modalità*, il metodo remoto accetta come argomento di ingresso il **nome del direttorio** e l'**endpoint di ascolto del client** e restituisce la **lista con i nomi e la lunghezza di tutti i file** da trasferire.

Il **client** richiede ciclicamente all'utente il nome del direttorio da trasferire, crea la socket di ascolto, effettua la chiamata RPC e riceve la lista dei file da trasferire; quindi, effettua la accept e i trasferimenti di file necessari.

Il **server** implementa il metodo RPC richiesto ed è realizzato come server *concorrente* e *parallelo*; per ogni nuova richiesta ricevuta il processo padre, dopo aver accettato la richiesta RPC, crea la socket per la connessione col client, attiva un processo figlio a cui affida il completamento del servizio richiesto e restituisce la struttura dati con la lista dei file.

Consegna

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