



Università degli Studi di Bologna
Scuola di Ingegneria

Corso di Reti di Calcolatori T

Esercitazione 8 (proposta) *Remote Procedure Call (RPC)*

Luca Foschini

Michele Solimando

Anno accademico 2016/2017

Esercitazione 8 1

Specifica

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

- **contare i caratteri**, **le parole** (si usi la definizione di parola data nell'esercitazione 1), **le linee** di un **file di testo** presente sul server remoto e **le occorrenze**, nello stesso file, **di una parola** inviata dal client;
- **contare il numero di file** (presenti nel **direttorio remoto** indicato dal client) **il cui nome inizia col prefisso richiesto dal client** e **di dimensione maggiore o uguale ad una soglia** indicata dal client.

Esercitazione 8 2

Specifica

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.

Esercitazione 8 3

Procedure remote

In particolare, si sviluppino le seguenti procedure:

- La procedura **conta_fileTesto** accetta come parametri d'ingresso **il nome del file** ed **una parola**, e restituisce **una struttura dati** che contiene quattro interi, corrispondenti al numero di occorrenze di caratteri, parole, linee contate ed occorrenze della parola nel file passati, e un intero per segnalare possibili errori, ad esempio, se il file non è presente nel sistema.
- La procedura **conta_filePref** accetta come parametro d'ingresso **il nome del direttorio remoto**, **il prefisso** e **una soglia numerica**, e restituisce **un intero positivo** con il numero di file contati, in caso di successo, -1 altrimenti.

Esercitazione 8 4

Note realizzative

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.

Esercitazione 8 5

REPETITA: XDR - alcuni consigli sulla definizione dei tipi di dati

I dati al **primo livello** (cioè quelli passati direttamente alle funzioni) possono essere passati **SOLO per valore** e **NON si possono passare** tipi di dato complessi (ad esempio gli array). Ad esempio:

```
string ECHO(string s);      Sì ☺  
char[] ECHO(char arrayCaratteri[12]);    No ☹
```

I dati al **secondo livello** (cioè definiti all'interno di altre strutture dati) possono invece usare anche strutture dati complesse (ad esempio array) e puntatori.

```
struct Input{char arrayCaratteri[12];}; } Sì ☺  
... Input ECHO(Input i);
```

Le **matrici** vanno però sempre definite **PER PASSI**:

```
struct Matrix{char arrayCaratteri[12][12];}; No ☹  
struct Riga{char arrayCaratteri[12];}; } Sì ☺  
struct Matrix{Riga riga[12];};
```

Esercitazione 8 6

REPETITA: XDR - alcuni consigli sulla definizione dei tipi di dati

Come definire una stringa all'interno di una struttura di un file XDR

```
struct Input
{
    string direttorio <50>;
    int x;
};
```

Una stringa va definita indicando la dimensione come massimo di default '<>' o massimo indicato esplicitamente '<50>', usando le **parentesi angolari**

Esercitazione 8 7



Proposta di estensione: Multiple Get



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 **client attivo** (**il client effettua la connect**), la seconda con **server attivo** (**il server effettua la connect**). 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**).

Esercitazione 8 8



Trasferimento più direttori: Client Attivo



La procedura remota 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, **effettua la chiamata RPC** e riceve l'**enpoint di ascolto**, quindi **stabilisce una connessione** con il server remoto e riceve i file salvandoli sul direttorio locale.

Il **Server** implementa la **procedura RPC richiesta** ed è realizzato come **server concorrente e parallelo**. Per ogni nuova richiesta ricevuta il processo padre, dopo **aver accettato la richiesta RPC**, attiva un processo figlio a cui affida la **creazione della socket di ascolto** e il completamento del servizio richiesto; quindi il padre restituisce **la lista dei file e il proprio endpoint**.

Esercitazione 8 9



Trasferimento più direttori: Server Attivo



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; infine, **effettua la accept** e i trasferimenti di file necessari.

Il **Server** implementa la procedura RPC richiesto ed è realizzato come **server concorrente e parallelo**; per ogni nuova richiesta ricevuta il processo padre, dopo aver **accettato la richiesta RPC**, attiva un processo figlio a cui affida la **creazione della socket** su cui eseguire la **connect** e il completamento del servizio richiesto; quindi **restituisce la lista dei file**.

Esercitazione 8 10

Consegna

Chi vuole può inviare lo svolgimento ai docenti, con lo strumento specificato sul sito del corso.