

Reti di Calcolatori T

Prova d'esame

14/12/2015 - Compito 1

Tempo a disposizione: 2h

Cognome:
Nome:
Matricola:

È obbligatorio inserire Cognome, Nome, e numero di Matricola all'inizio di ogni file sorgente, pena la non valutazione del compito, che viene stampato in modo automatico solo in caso siano presenti gli elementi detti sopra.

Si devono consegnare tutti i file sorgente e tutti gli eseguibili prodotti singolarmente (per favore, solo quelli relativi ai file sorgente consegnati!!!).

La prova intende valutare le capacità progettuali e di programmazione sia in ambiente Java che in ambiente C, pertanto è consigliato sviluppare entrambe le soluzioni richieste al meglio.

In entrambi gli esercizi, sia in Java che in C, si effettuino gli opportuni controlli sui parametri della richiesta e si gestiscano le eccezioni, tenendo presente i criteri secondo cui si possa ripristinare il funzionamento del programma oppure si debba forzarne la terminazione.

Leggete con attenzione le specifiche del problema prima di impegnarvi "a testa bassa" nello sviluppo delle singole parti. Naturalmente, ci aspettiamo che i componenti da consegnare siano stati provati e siano funzionanti.

Si richiede il progetto della gestione dei servizi PrenotazioniCampeggio, per la gestione delle prenotazioni di un campeggio. I servizi di PrenotazioniCampeggio mantengono, per ogni prenotazione, le seguenti informazioni: l'identificatore della prenotazione (unico all'interno del sistema), il numero di persone, la tipologia di prenotazione (una stringa che può assumere uno dei seguenti valori: 'mezza piazzola', 'piazzola' o 'piazzola deluxe'), il tipo di veicolo prenotato (una stringa che può assumere uno dei seguenti valori: niente - se non c'è veicolo, camper, moto, auto), la targa del veicolo e un file immagine con la foto della piazzola assegnata. I file con gli immagini sono mantenuti nel directorio corrente, dove vengono lanciati server e client.

In particolare, si richiede di realizzare le seguenti funzionalità:

1. **download delle foto delle piazzole relative ad una tipologia di prenotazione:** questa operazione richiede il tipo di prenotazione e scarica tutti i file immagine corrispondenti alle prenotazioni di tale tipo;
2. **aggiornamento del numero persone:** questa operazione richiede l'identificatore della prenotazione e il numero di persone da aggiornare, quindi aggiorna la struttura dati;
3. **eliminazione di una prenotazione:** questa operazione richiede un identificatore di prenotazione quindi elimina prenotazione e file immagine relativi all'identificatore richiesto;
4. **visualizzazione di tutte le prenotazioni di un certo tipo, senza veicolo e con un numero di persone maggiore di una soglia:** questa operazione richiede il numero di persone e il tipo di prenotazione, e stampa a video la lista delle prenotazioni del tipo richiesto per cui come tipo di veicolo è stato indicato 'niente' e il numero di persone è maggiore della soglia richiesta.

Si progetti con particolare attenzione la **struttura dati** che mantiene lo stato, fino ad un massimo di N prenotazioni (L, per libero a default), da implementare opportunamente nei diversi ambienti richiesti, Java e C.

Identificatore	Numero persone	Tipologia prenotazione	Tipo veicolo	Targa veicolo	File immagine
HGFD89	3	mezza piazzola	niente	L	mezza_piazz1.jpg
WERT26	5	piazzola deluxe	camper	AA567AA	piazz_deluxe2.jpg
L	L	L	L		L
...
QLJC33	4	piazzola	auto	BB444BB	piazz3.jpg

Si considerino e si segnalino le possibilità di interferenze fra le operazioni, evitandole dove necessario.

Parte Java

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

- **eliminare una prenotazione;**
- **visualizzare tutte le prenotazioni senza veicolo, di un certo tipo e con un numero di persone maggiore di una soglia.**

Il progetto RMI si basa su un'interfaccia (contenuta nel file *RMI_interfaceFile.java*) in cui vengono definiti i metodi invocabili in remoto dal client:

- Il metodo **elimina_prenotazione** accetta come parametro d'ingresso l'identificatore della prenotazione, e restituisce un intero con l'esito dell'operazione.
- Il metodo **visualizza_prenotazioni** accetta come parametri d'ingresso il numero di persone e il tipo di prenotazione, e restituisce la lista (array) dei dati delle presentazioni (non i contenuti dei file immagine) del tipo richiesto, per cui come tipo di veicolo è stato indicato 'niente' e per cui il numero di persone è maggiore della soglia richiesta.

Si progettino inoltre le classi:

- **RMI_Server** (contenuta nel file *RMI_Server.java*), che implementa i metodi del server invocabili in remoto;
- **RMI_Client** (contenuta nel file *RMI_Client.java*), il processo che realizza l'interazione con l'utente proponendo ciclicamente i servizi che utilizzano i due metodi remoti, e stampa a video i risultati, fino alla fine del file di input da tastiera.

Parte C

Progettare un servitore multiservizio (usando obbligatoriamente una select) che consenta di effettuare le operazioni remote per:

- **download delle foto delle piazzole relative ad una tipologia di prenotazione,**
 - **aggiornare il numero di persone,**
- utilizzando un'unica connessione per ogni sessione cliente.

Più in dettaglio:

- Il **client_stream** è organizzato come un **processo filtro ciclico fino alla fine del file di input** e realizza la funzionalità di **download delle foto delle piazzole relative ad una tipologia di prenotazione** utilizzando **socket stream** e un'unica socket.
Per ogni richiesta, il client chiede all'utente e invia al server il tipo di prenotazione, quindi riceve i nomi e i file delle immagini e li salva sul file system locale.
- Il **client_datagram** è organizzato come un **processo filtro ciclico fino alla fine del file di input** e realizza la funzionalità di **aggiornamento del numero di persone** utilizzando **socket datagram**.
Per ogni richiesta, il client chiede all'utente e invia al server l'identificatore della prenotazione e il numero di persone da aggiornare; quindi riceve l'esito dell'operazione stampandolo a video.
- Il **server** principale unico discrimina le richieste utilizzando la primitiva select: il **server gestisce in modo parallelo** la funzionalità di download delle foto; mentre la funzionalità di aggiornamento del numero di persone è **realizzata in modo seriale**.
Per ogni richiesta di **download delle foto delle piazzole relative ad una tipologia di prenotazione**, il figlio riceve il tipo di prenotazione, quindi invia nome e contenuto di tutti i file delle immagini relativi alle prenotazioni richieste usando l'unica connessione della sessione.
Per ogni richiesta di **aggiornamento del numero di persone** il server riceve l'identificatore della prenotazione e il numero di persone da aggiornare, quindi aggiorna la struttura dati e restituisce un intero positivo che indica l'esito dell'operazione: 0 in caso di successo, -1 in caso di problemi.