

Reti di Calcolatori T

Prova d'esame

23/01/2019 - Compito 2

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 **RentAcaR**, per la gestione delle prenotazioni di autovetture condivise. I servizi di **RentAcaR** mantengono, per ogni prenotazione, le seguenti informazioni: la **targa del veicolo** (unico all'interno del sistema), la **patente di guida** dell'utente (una stringa di 5 interi), il **tipo di veicolo prenotato** (una stringa che può assumere uno dei seguenti valori: **camper**, **auto**), le **immagini del veicolo**. I file con le immagini relative ad un veicolo sono mantenuti in una cartella denominata '<targa>_img' situata nel direttorio corrente dove vengono lanciati server e client. In particolare, si richiede di realizzare le seguenti funzionalità:

1. **download delle foto delle vetture in base al numero di targa**: questa operazione richiede il numero di targa e scarica le immagini presenti nella cartella relativa al veicolo;
2. **aggiornamento della patente di guida**: nel caso in cui cambi il guidatore, questa operazione richiede il numero di targa del veicolo e il numero di patente del nuovo guidatore, quindi aggiorna la struttura dati;
3. **eliminazione di una prenotazione**: questa operazione richiede il numero di targa di un veicolo quindi elimina la prenotazione e il file immagine relativi, aggiornando la struttura dati e il file system;
4. **visualizzazione di tutte le prenotazioni di un certo tipo di veicolo** (camper, auto), **immatricolato successivamente al 2011**: questa operazione richiede il tipo di veicolo e stampa a video la lista delle prenotazioni richieste. Le targhe vengono generate in successione cambiando le prime due lettere (es. **AAxxxXX**, **ABxxxXX**, **ACxxxXX**, ..., **EDxxxXX**), consideriamo targhe del 2011 quelle che iniziano con le lettere **'ED'**.

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.

Targa del veicolo	Patente di guida	Tipo del veicolo	Folder immagini
AN745NL	00003	auto	AN745NL_img/
FE457GF	50006	camper	FE457GF_img/
L	0	L	L
...
NU547PL	40063	auto	NU547PL_img/
LR897AH	56832	camper	LR897AH_img/
MD506DW	00100	camper	MD506DW_img/

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

Parte Java

Sviluppare un'applicazione C/S basata su **socket stream** che realizzi le operazioni remote per:

- **eliminare una prenotazione;**
- **scaricare le foto di un veicolo richiesto:**

utilizzando un'unica connessione per ogni sessione cliente: questo vincolo è da intendersi come **prioritario e fondamentale**.

Più in dettaglio:

- Il **cliente** è organizzato come un **processo filtro ciclico che consuma l'input fino a fine file** e, per ogni iterazione del ciclo, chiede all'utente quale tipo di operazione vuole effettuare e realizza le interazioni col server utilizzando **una sola connessione per la intera sessione**; alla ricezione di fine file, libera opportunamente le risorse e termina.
Per ogni richiesta ricevuta dall'utente, il client prima invia il tipo di servizio al server, poi gestisce gli invii e le ricezioni necessarie alla realizzazione dello specifico servizio richiesto.
Nel caso della funzionalità di **eliminazione di una prenotazione**, per ogni richiesta, il client chiede all'utente il numero di targa di un veicolo e riceve l'esito dell'operazione dal server.
Nel caso di **download delle foto**, per ogni richiesta, il client chiede all'utente il numero di targa di un veicolo ed esegue la multiple-get delle foto relative al veicolo, salvandole in locale.
- Il **server** è organizzato come un **unico processo che gestisce in modo parallelo** l'interazione coi clienti, generando un figlio per tutta la sessione di richieste da quel client. Per ogni richiesta, il processo figlio che serve la sessione con una prima lettura discrimina il tipo di funzionalità richiesto, poi gestisce opportunamente l'operazione e si pone in attesa di nuove richieste dallo stesso client; alla lettura della fine sessione, il figlio termina.
Nel caso della funzionalità di **eliminazione di una prenotazione**, il server riceve il numero di targa di un veicolo, elimina il record dalla tabella e le foto del veicolo dal file system, quindi invia l'esito dell'operazione al cliente.
Nel caso di **download delle foto**, il server riceve il numero di targa di un veicolo e quindi invia nome e contenuto di tutti i file delle immagini richieste usando l'unica connessione della sessione.

Parte C

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

- **visualizzare tutte le prenotazioni di un certo tipo di veicolo (camper o auto), immatricolato successivamente al 2011,**
- **aggiornare il numero di patente.**

Il progetto prevede che il server metta a disposizione due procedure (parte di una interfaccia specificata in XDR e contenuta nel file **RPC_xFile.x**) invocabili in remoto dal client:

- la procedura **visualizza_prenotazioni** accetta come parametro d'ingresso il **tipo di veicolo**, e restituisce la lista (array) delle prenotazioni del tipo richiesto, e di veicoli la cui immatricolazione è avvenuta successivamente all'anno 2011, cioè aventi come prime due lettere della targa un valore maggiore di **'ED'**. Restituire al massimo i primi 6 risultati utili.
- la procedura **aggiorna_licenza** accetta come parametro d'ingresso il numero di targa e il nuovo numero di patente, aggiorna il record in tabella e restituisce l'esito dell'operazione.

Si progettino inoltre i sorgenti:

- **RPC_Server** (contenuta nel file **RPC_Server.c**), che implementa le procedure del server invocabili in remoto;
- **RPC_Client** (contenuta nel file **RPC_Client.c**), il processo filtro che realizza l'interazione con l'utente, propone ciclicamente i servizi che utilizzano le due procedure remote, e stampa a video i risultati, fino alla fine dello stream di input.