

Tema de casă #1 – RPC Post Office Service

Responsabil: Florin Pop (florin.pop@cs.pub.ro)

Data publicării: 22.10.2012

Termenul de predare: 05.11.2012, ora 23.55.

Obiective

După realizarea acestei teme de casă studentul va fi capabil să:

- realizeze o aplicație client-server bazată pe RPC.
- lucreze cu structuri în specificațiile RPC.
- gestioneze eficient comunicatia de date client-server.

Cunoștințele necesare rezolvării acestei teme de casă sunt:

- Programare în C, tablouri, structuri.
- Noțiunea de socket TCP.
- Noțiuni elementare de geometrie analitică.

Enunțul problemei

Realizați o aplicație de tip serviciu RPC care permite determinarea rutelor unor colete în sistemul de poștă tradițională. Avem coordonatele oficiilor poștale date prin $\langle lat, long \rangle$ (latitudine și longitudine), coordonate cunoscute de o aplicație server. Un client (mobil), pentru care se cunosc coordonatele geografice, are o listă de colete ce vor fi trimise în diverse locații pe glob. Fiecare colet este identificat prin $\langle ID, lat, long \rangle$ (identificator, latitudinea și longitudinea locației de destinație). Traseul unui colet este următorul: *locația clientului* \rightarrow *oficiu poștal sursă* \rightarrow *oficiu poștal destinație* \rightarrow *destinația finală*. Clientul trimite lista de colete serverului, iar acesta va răspunde, pentru fiecare colet în parte, cu traseul urmat și lungimea acestuia.

Client-ul

Clientul primește ca parametru al linie de comanda datele de localizare și un fisier în care sunt trecute M coletele (fișier text cu M linii, fiecare linie conținând 3 numere: ID, lat și long). Exemplu:

```
./poclient lat long filename
```

Clientul va avea o structură pentru fiecare colet, cu datele specifice acestuia. Identificatorul este un număr întreg pe 4 octeți. Latitudinea și Longitudinea sunt numere reale pe 4 octeți, în simplă precizie (IEEE754). Toate latitudinile sunt între -90 și $+90$ grade; toate longitudinile sunt între -180 și $+180$ grade.

Clientul va trimite o interogare server-ului pentru fiecare colet în parte. Răspunsul primit de la server se va afișa pe ecran sub forma: ID_oficiu_sursa ID_oficiu_destinatie distanta, câte o linie pentru fiecare colet, unde distanța va fi dată în kilometri (numar real în simplă precizie).

Propuneți și implementați o modalitate de a minimiza numărul de mesaje schimbate de client și server pentru situația în care clientul are un numar foarte mare de specificații de colete în fisierul de intrare. Metoda propusa și implementarea ei vor fi descrise în fisierul README .

Server-ul

La pornirea server-ului aplicației se va încărca fișierul binar cu numle po.dat care conține următoarele date: un numar N de oficii poștale (întreg pe 4 octeți, $N < 3000000$). Pentru fiecare oficiu poștal, fișierul de intrare va conține două numere reprezentând latitudinea și longitudinea acestuia. Latitudinea și Longitudinea sunt numere reale pe 4 octeti, în simplă precizie (IEEE754). Toate latitudinile sunt între -90 și $+90$ grade; toate longitudinile sunt între -180 și $+180$ grade. Identificatorul oficiului poștal va fi un numar natural de la 1 la N, în ordinea apariției în fisierul de intrare. Serverul poate calcula distanța dintre două oficii poștale $O_1 \langle lat_1, long_1 \rangle$ și $O_2 \langle lat_2, long_2 \rangle$ considerand următoarea formulă:

$$d(O_1, O_2) = 2R_p \arctan\left(\sqrt{\frac{x}{1-x}}\right)$$

unde:

- R_p este raza Pamantului. Vom considera Pamantul o sferă cu raza $R_p = 6380km$.
- $x = \sin^2\left(\frac{lat_1 - lat_2}{2}\right) + \cos(lat_1) \cdot \cos(lat_2) \cdot \sin^2\left(\frac{long_1 - long_2}{2}\right)$

Puteti folosi și formula tradițională cu \arcsin , dar aceasta este instabilă numeric pentru valori mici ale unghiului la centru, între cele puncte (oficii poștale) de pe glob. În programare se va folosi funcția atan2 , care elimină erorile de virgulă mobilă în cazul în care numitorul arctangentei este foarte mic.

Precizări legate de implementarea temei

- Tema se va implementa în limbajul C, folosind pentru compilare utilitarul `rpcgen`;
- Fișierul de definire a programului, funcțiilor și structurilor se va numi `po.x`;
- Functionalitatea clientului se va scrie în fișierul `poclient.c`;
- Functionalitatea server-ului se va scrie în fișierul `poserver.c`;
- Scrieți un fișier `makefile` care, prin comanda `make` compilează aplicația, obținând fișierele binare, iar prin comanda `make clean` șterge toate fișierele generate de comanda `make`;
- Tema de casa va conține un fișier README cu explicații privitoare la modul de definire al structurilor pentru comenzi și rezultat și la modul de implementare a funcțiilor folosite de client și de server;
- Fișierele temei trebuie **OBLIGATORIU** împachetate într-o arhiva de tip '**.zip**', cu numele 'Grupa_NumePrenume_Tema1.zip'. Această arhivă conține și fișierul README.
- Arhiva temei **NU** va conține fișiere binare.