

# Tema de casă #1 – RPC+DB

Responsabili de temă: Florin POP ([florin.pop@upb.ro](mailto:florin.pop@upb.ro)), Ion-Dorinel FILIP ([dorinel.filip@upb.ro](mailto:dorinel.filip@upb.ro))

Termenul de predare: 22.11.2020

## Obiective

Obiectivele temei de casă, care contribuie la realizarea competențelor profesionale cu privire la protocolul RPC sunt următoarele:

- realizarea unei aplicații client-server bazată pe RPC cu cereri/răspunsuri multiple prin care se poate gestiona o bază de date stocată la distanță;
- scrierea de specificații pentru o aplicație simplă de administrare a unei baze de date;
- gestionarea eficientă a încărcării unui server de aplicații;
- utilizarea de structuri de date pentru aplicații ce folosesc protocolul RPC.

Cunoștințele necesare rezolvării acestei teme de casă:

- Programare în C/C++, structuri, funcții, lucrul cu fișiere binare/text.
- Noțiuni privind folosirea compilatorului RPC.

## Enunțul problemei

Realizați o aplicație de gestiune a unei baze de date, aflate pe un sever de aplicații care are la bază protocolul RPC. Aplicația poartă numele RPC+DB și are două componente: un client și un server.

*Server*-ul va păstra în memorie o bază de date. Înregistrările care sunt stocate în baza de date reprezintă datele monitorizate de diverși senzori pe anumite intervale de timp, de forma:

```
struct sensor_data{  
    int data_id;           // identificatorul datelor  
    int no_values;         // numărul de valori  
    float *values;         // valorile monitorizate  
}
```

Server-ul va expune diverse operații de lucru cu baza de date. Un set minimal de operații este:

- add/update – adăugarea/actualizarea unui înregistrări noi (dacă deja există o înregistrare cu identificatorul specificat, se va întoarce o eroare);
- del – ștergerea unei înregistrări specificate prin identificator (dacă aceasta nu există se va genera un cod de eroare);
- update – înlocuirea unei înregistrări specificate prin identificator cu date noi (dacă nu există înregistrarea se va adăuga ca înregistrare nouă);
- read – citirea unei înregistrări specificate prin identificator (dacă aceasta nu există se va genera un cod de eroare);
- get\_stat – obținerea de statistici pentru o înregistrare (min, max, medie și mediană);
- get\_stat\_all – statistici generale pe toată baza de date.

Server-ul oferă două comenzi, load și store, pentru încărcarea bazei de date dintr-un fișier în memorie și respectiv pentru scrierea acesteia în fișier.

*Clientul* are la dispoziție un fișier cu comenzi care reprezintă operații asupra bazei de date, ca în următorul exemplu orientativ:

```
login user
load
add 1110 4 1.0 2.5 0.5 3.5
del 2256
update 3345 3 7.0 2.3 9.1
read 2134
get_stat 2134
get_stat_all
store
logout
```

Prima comandă realizează o autentificare a clientului la server, prin care acesta primește o cheie de sesiune generată de server (un număr natural de tipul `unsigned long`). Server-ul păstrează o listă cu perechi (`user`, `session_key`) pentru gestiunea clienților. Comanda `login` poate eșua (fiind respinsă de server) dacă există deja un utilizator cu același user autentificat în sistem.

Fiecare client are propria bază de date pe care efectuează operații. După autentificare, clientul poate încărca de pe disk baza de date existentă (creată în sesiunile anterioare) prin comanda `load` sau poate crea o bază nouă în memorie dacă nu se execută comanda `load`. Dacă la execuția comenzii `load` nu se găsește baza de date, se va crea una nouă. Dacă comanda `load` se execută după efectuarea altor comenzi, aceasta va eșua. Prin comanda `store` se va suprascrive baza de date de pe disk cu cea existentă în memorie (fără a șterge memoria). Dacă după comanda `store` se mai execută alte comenzi, efectul acestora se va regăsi doar în memorie, până la o nouă execuție a comenzii `store`. Numele bazei de date poate fi de forma: `user.rpcdb`.

La terminarea execuției, prin comanda `logout`, se va închide sesiunea prin ștergerea perechii (`user`, `session_key`) păstrate de server și prin eliberarea memoriei folosite pentru baza de date.

## Cerințele pentru realizarea aplicației RPC+DB

1. Definiți interfața care expune tipurile de date și metodele aplicației;
2. Descrieți specificațiile de funcționare (prin comentarii în interfață) pentru toate tipurile de date și funcțiile definite în interfață;
3. Implementați aplicația server care păstrează baza de date și care este gestionată prin operațiile descrise în interfață;
4. Implementați un model de client de test pentru aplicația server care va citi comenzile de executat dintr-un fișier text primit ca parametru al comenzii și le va executa în ordine;
5. Oferiți cel puțin 3 fișiere de test pentru baza de date.

## Mențiuni suplimentare

- Tema se va implementa în C/C++;
- Specificațiile de funcționare sunt descrise de cel care realizează implementare aplicației, nu sunt impuse de către cei care vor evalua aplicația RPC+DB;
- Evaluarea se va face pe baze fișierelor de test (de bază) puse la dispoziție și a unor alte fișiere de test propuse de evaluator pe baze testelor de bază;
- Testele de bază vor conține toate operațiile definite în specificații (cel puțin o dată);
- Vă puteți baza pe specificațiile oficiale [https://docs.oracle.com/cd/E18752\\_01/html/816-1435/rpcproto-24229.html](https://docs.oracle.com/cd/E18752_01/html/816-1435/rpcproto-24229.html) și pe aplicația dezvoltată în cadrul laboratorului.