# PSO proiect

# RPC- Remote procedure call

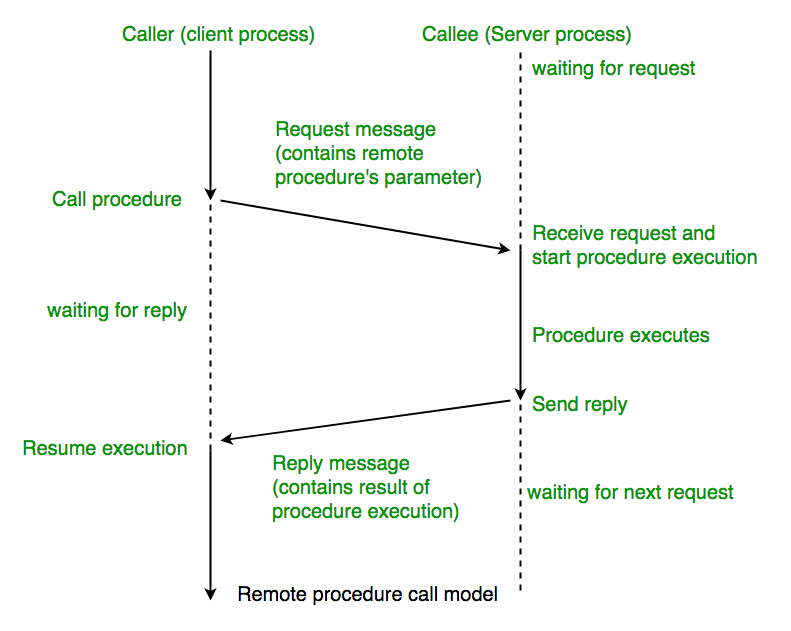
Documentație

Sd. sg. Simulescu Vlad,

Sd. sg. Crihană Dragoș,

C 113C

* **definirea conceptului-** o aplicație RPC clasică va consta dintr-un client și un server. Aplicația client va executa prin intermediul serverului o procedură, serverul fiind cel care o execută și cel care trimite înapoi către client rezultatul.



* **descrierea proiectului-** acest proiect se va folosi pentru implementarea aplicației RPC de mai multe procese implementate pe VM-uri diferite: un proces administrator și alte procese client. Procesele client se vor ajuta de procesul server pentru realizarea anumitor funcții. Proiectul are drept scop posibilitatea de a crea un program care să se folosească de biblioteca creată de noi pentru a îndeplinii anumite funcționalități în acel program.
* **Serverul-** rulează pe un VM diferit de cel pe care rulează codul clientului. Acesta va avea la dispoziție un tabel de funcții pe care le poate îndeplinii. La pornirea procesului server se va deschide un socket pentru comunicare, care va aștepta request-uri de la clienți. Se va crea un thread principal care are drept scop ascultarea de cereri și distriburiea lor către thread-uri secundare spre executare. Tot la pornirea procesului pe lângă crearea unui thread principal vor mai fi create un anumit număr prestabilit de thread-uri secundare care vor avea ca scop executarea efectivă a funcțiilor. Fiecarui thread secundar i se va asigna o cerere primită de la un client. În momentul în care numărul de cereri depășește thread-urile libere, numărul acestora se va suplimenta. Acesta se va ocupa de următoarele:
  + thread-ul așteaptă primirea unei cerințe( sub formă de pachet)
  + deserializarea pachetului primit de la client
  + interpretarea cererii: se va verifica în primul rând dacă cererea formată dintr-o funcție și parametrii acesteia există în tabela cu funcții pe care serverul le poate executa și dacă parametrii primiți sunt corespunzători; în cazul în care se găsesc greșeli la aceasta, se va serializa un mesaj de eroare într-un pachet, acesta fiind trimis clientului
  + în cazul în care funcția există și parametrii primiți sunt valizi se va continua prin executarea ei și generarea unui rezultat
  + rezultatul va fi serializat într-un pachet și transmis către client
  + thread-ul se eliberează și așteaptă asignarea unei cerințe noi
* **Clientul-** codul clientului va avea posibilitatea apelării unei funcții prin intermediul unei biblioteci, aceasta ocupânduse de:
  + deschiderea unui socket pentru comunicarea cu serverul
  + serializarea funcției într-un pachet ce va fi transmis serverului
  + trimiterea pachetului către server
  + așteptarea pachetului ce conține răspunsul
  + deserializarea pachetului ce conține răspunsul
  + interpretarea răspunsului primit
* Funții pe care serverul să le poată îndeplinii:
  + Funcții de criptare și decriptare
  + Hash
  + Funcții matematice
  + Funcții pentru managementul fișierelor
  + Funcții pentru managementul proceselor