

Table of Contents



Implementing RPC Mechanism

Stubs

Basic RPC Operation

What is RPC mechanism?

The steps in making a RPC

Examples of RPC

Design Issues in RPC

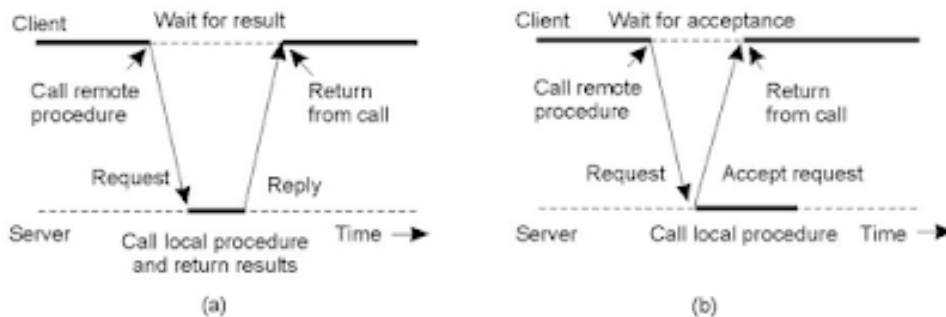
Related posts:

Implementing Rpc Mechanism

To achieve semantic transparency, implementation of the RPC mechanism is based on the concepts of stubs.

Stubs

- It provides a normal/local procedure call abstraction by concealing the underlying RPC mechanism.
- A separate stub procedure is associated with both the side on client and server processes.
- To hide the distance and functional underlying communication network, an RPC communication package Known as RPC Runtime is used on both the sides.
- Thus the implementation of RPC involves the five elements of the program:-
 1. The client
 2. The client stub
 3. The RPC Runtime
 4. The server stub
 5. The server



Principle between client and server program

Basic RPC Operation

The RPC Mechanism is based on the observation that a procedural call is well known for transfer of control and data within a program running on a single machine.

On invoking a remote procedure, the calling process is suspended.

If any parameter is passed to the remote machine where the procedure will execute.

On completion, the results are passed back from server to client and resuming execution as if it had called a local procedure.

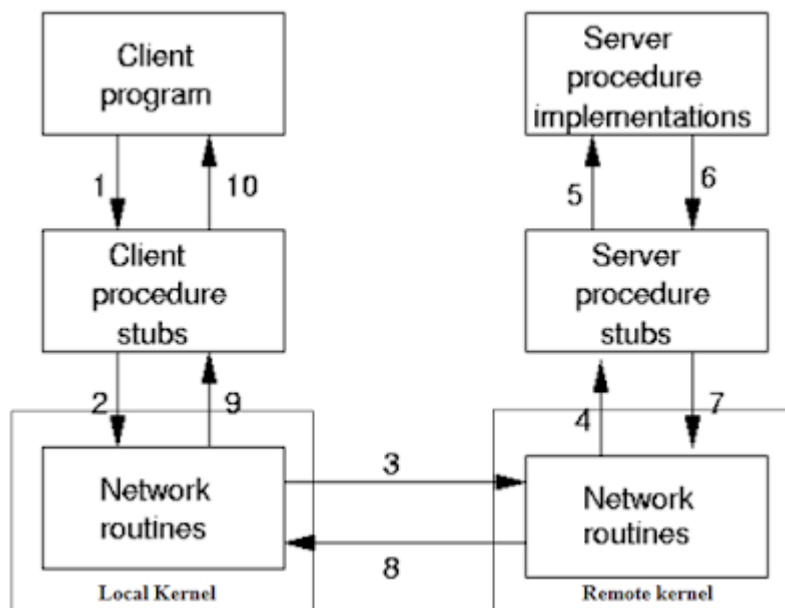
RPC mechanism is based on the concept of stub procedures.

The server writer writes the server and links it with the server-side stubs; the client writes his program and links it with the client-side stub.

The stubs are responsible for managing all details of the remote communication between client and server.

What is RPC mechanism?

They enable client to communication with server by calling procedures in a similar way to the conventional use of procedure on the local procedure call, but the call procedure is executed in a different process and usually a different computer.



Steps of RPC

The steps in making a RPC

- Client procedure calls the client stub in a normal way.
- Client stub builds a message and traps to the kernel.
- Kernel sends the message to remote kernel.
- Remote kernel gives the message to server stub.
- Server stub unpacks parameters and calls the server.
- Server computes results and returns it to server stub.

- Server stub packs results in a message to client and traps to kernel.
- Remote kernel sends message to client stub.
- Client kernel gives message to client stub.
- Client stub unpacks results and returns to client.

Examples of RPC

- Distributed Computing Environment.
- Open Software Foundation.
- Middleware between existing network operating system and distributed application.
- Initially design for Unix-Win NT.

Design Issues in RPC

- RPC mechanism is based on the concept of stub procedures.
- The server writer writes the server and links it with the server-side stubs; the client writes his program and links it with the client-side stub.
- The stubs are responsible for managing all details of the remote communication between client and server.

Related Posts:

1. RPC messages
2. RPC Implementation
3. Advantages Disadvantages of DS
4. Distributed computing models
5. Goals of DS
6. Hardware software concepts
7. Issues in designing ds

8. Design and Implementation Issues DS
9. Structure of share memory space
10. DSM Architecture & its Types
11. File Application & Fault tolerance
12. File service architecture
13. Desirable features of good distributed file system
14. Distributed shared memory
15. Election algorithm
16. Client server communication
17. Data representation and Marshalling
18. Communication between distributed objects
19. Load distributing algorithm
20. Task migration and its issues
21. Deadlock issues in deadlock detection & resolution
22. Distributed Scheduling-Issues in Load Distributing
23. Characteristics of Multimedia Data
24. Case Study of Distributed System
25. Distributed multimedia system
26. Distributed DBMS
27. Advantages of DDBMS over centralised DBMS