

Lecture 6.g

Client Server Communication In Class Test 1 Review

Dr. Gabriele Pierantoni

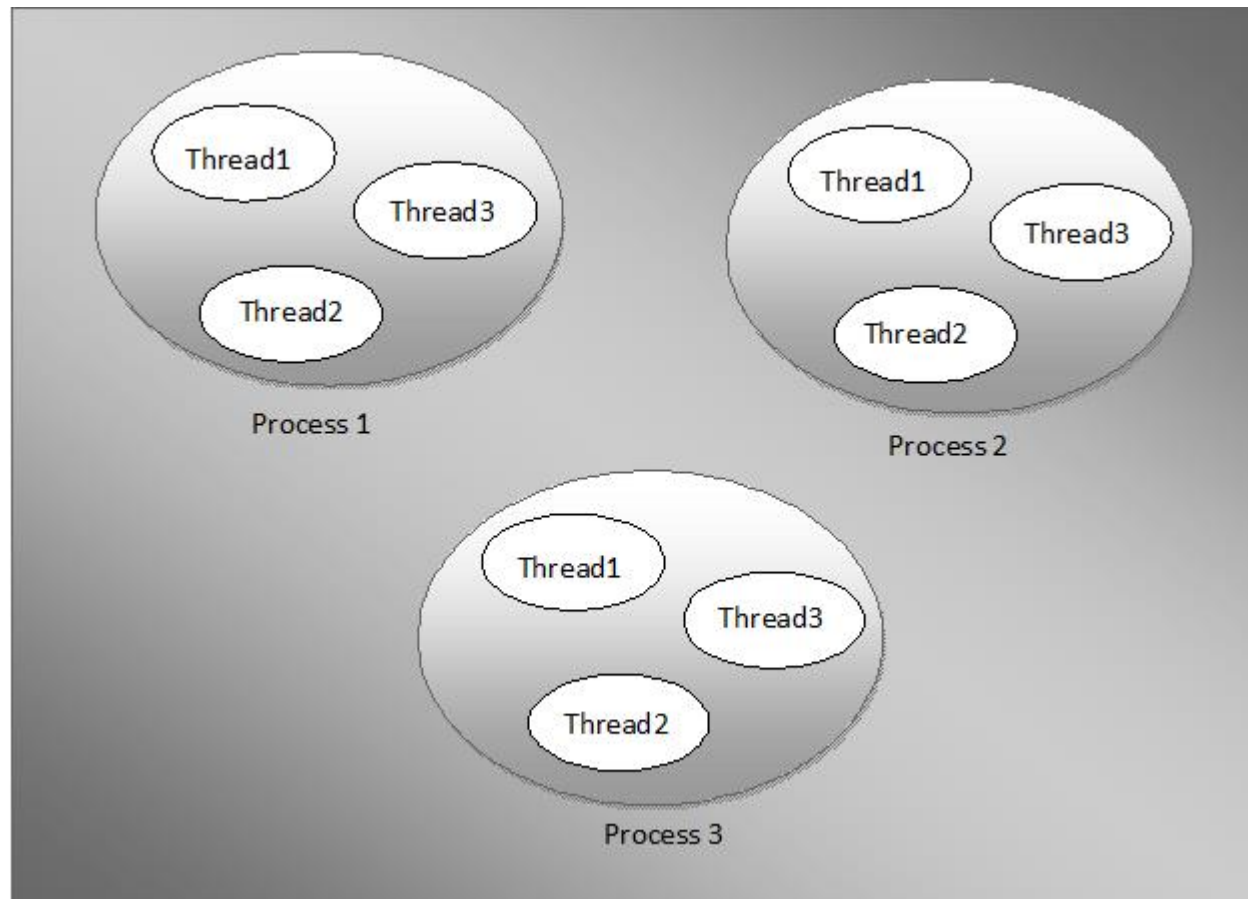
24.02.2021

Client/Server Communication

Some Basic Concepts - Program

- A computer **program** is a collection of instructions that performs a specific task when executed by a computer.
- A **process** is an instance of a computer **program** that is being executed. It contains the **program** code and its current activity. Depending on the operating system (OS), a **process** may be made up of multiple threads of execution that execute instructions concurrently
- A process can have multiple threads. Each thread will have their own task and own path of execution in a process. For example, in a notepad program, one thread will be taking user inputs and another thread will be printing a document.
- All threads of the same process share memory of that process. As threads of the same process share the same memory, communication between the threads is fast.

Some Basic Concepts – Program, Processes and Threads



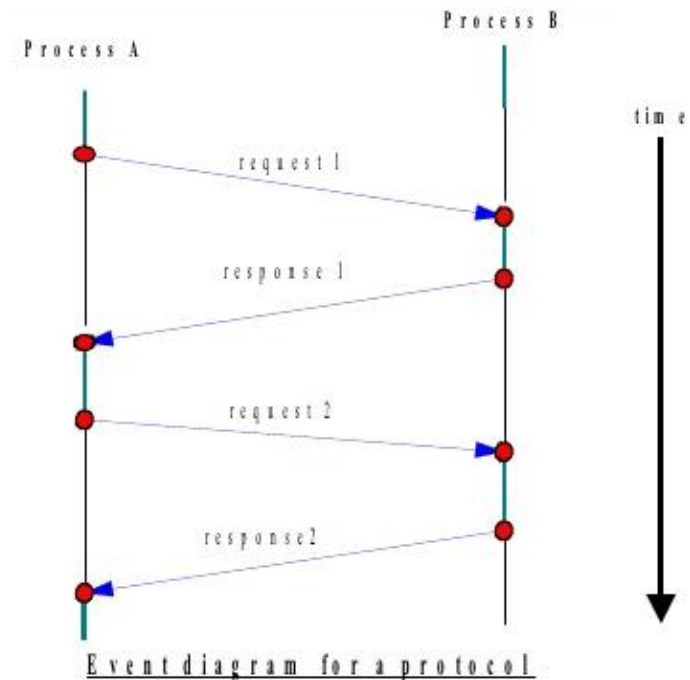
Inter-Process Communication (IPC)

Definition:

- IPC = a set of methods for the exchange of data among multiple processes or computers
- IPC provides mechanism for processes to communicate and to synchronize their actions

IPC types:

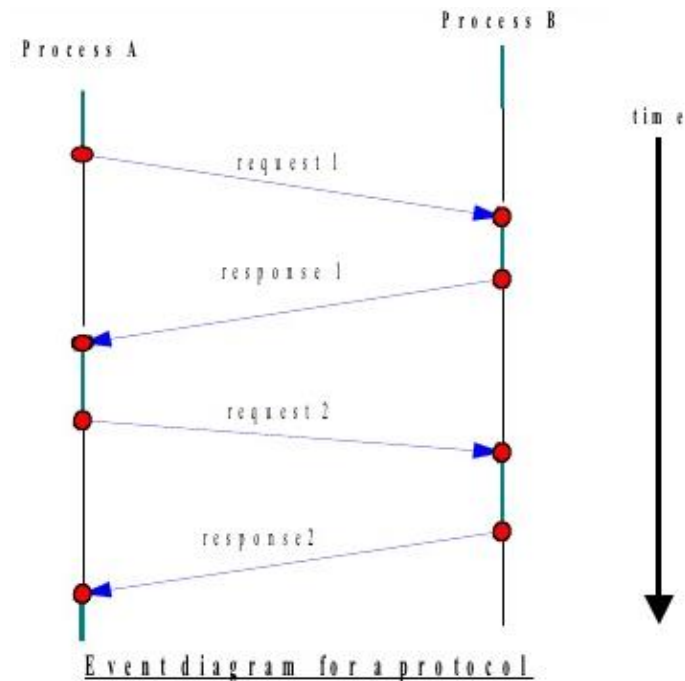
- Message Passing Communication
- Request/Reply Communication
- Transaction Communication



Inter-Process Communication (IPC)

IPC types:

Request/Reply Communication: one of the basic methods computers use to communicate with each other, in which the first computer sends a request for some data and the second computer responds to the request. Browsing a web page is an example of request–response communication.



Inter-Process Communication: RPC

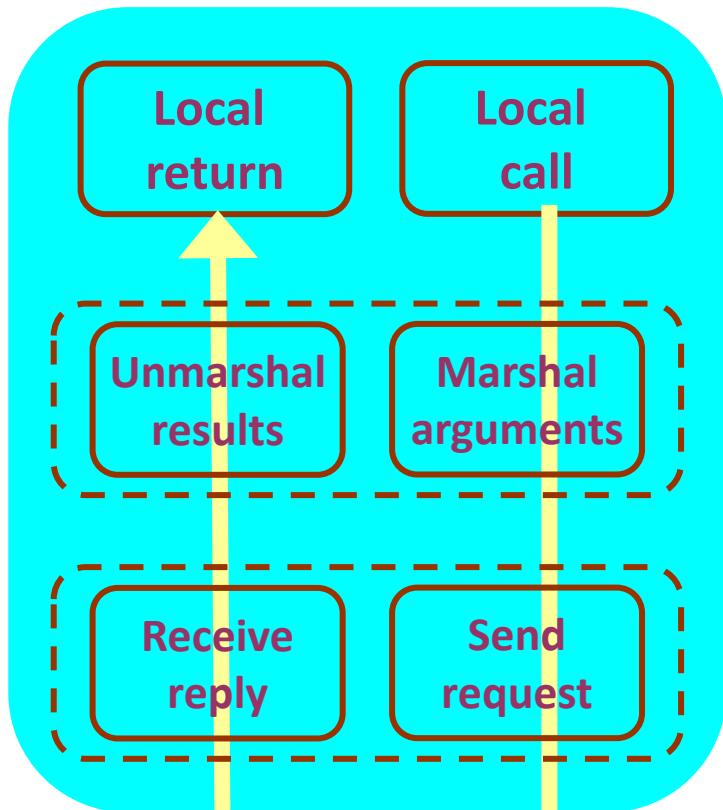
Remote Procedure Call (RPC)

In distributed computing, a **remote procedure call (RPC)** is when a computer program causes a procedure (subroutine) to execute in a different address space (commonly on another computer on a shared network), which is coded as if it were a normal (local) procedure call, without the programmer explicitly coding the details for the remote interaction.

That is, the programmer writes essentially the same code whether the subroutine is local to the executing program, or remote. This is a form of client–server interaction (caller is client, executor is server), typically implemented via a request–response message-passing system.

Inter-Process Communication: Request - Reply

Client computer



client
process

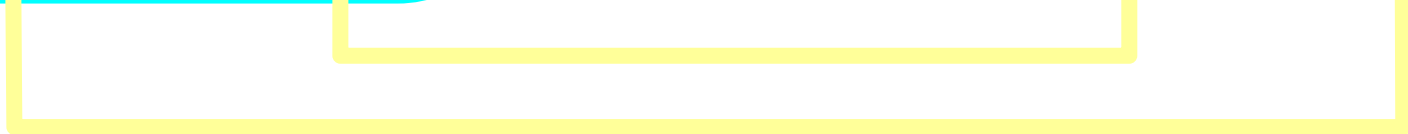
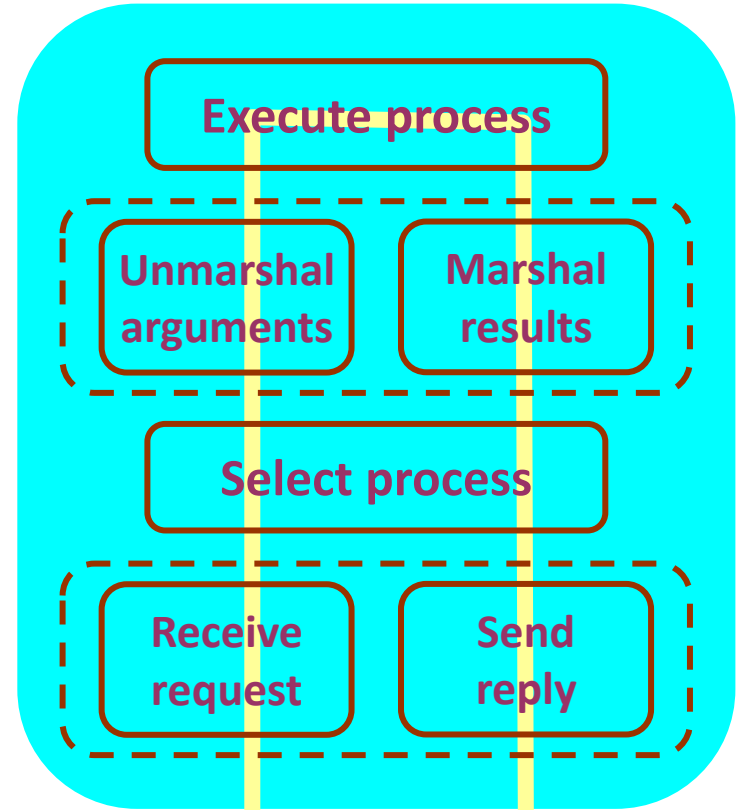
client
stub
process

communication
module

service
process

server
stub
process

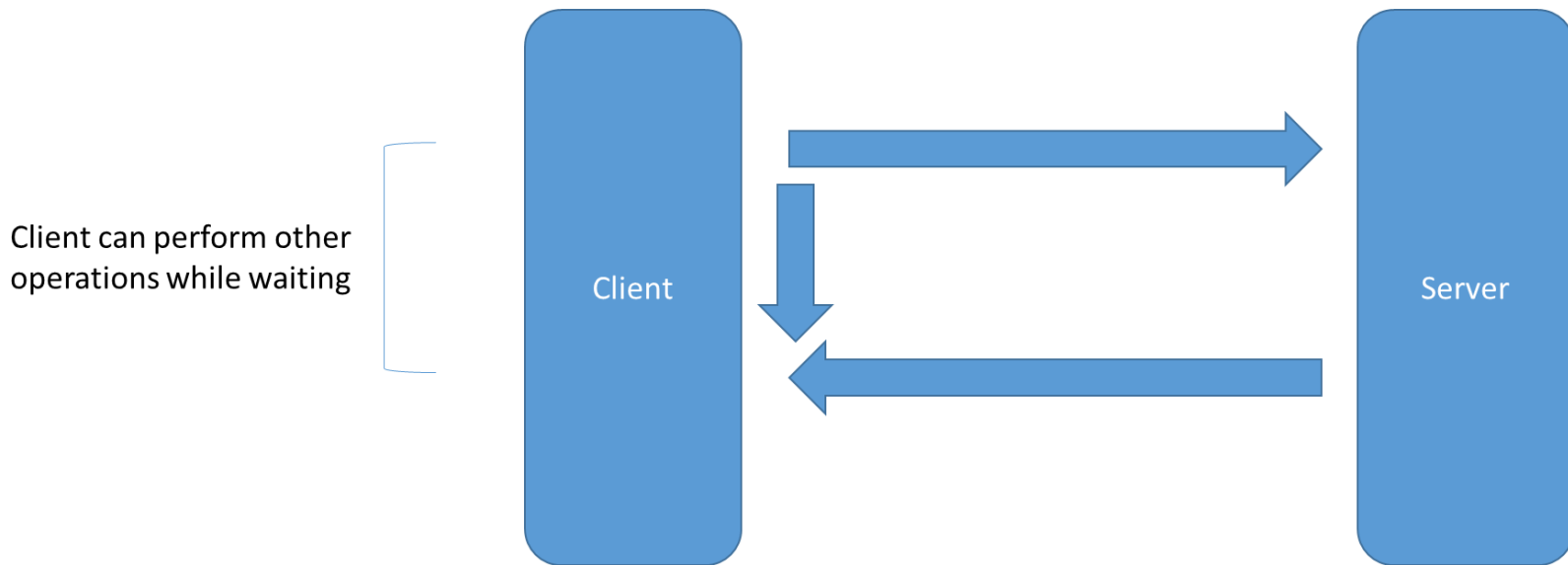
Server computer



Asynchronous Message Passing

asynchronous communication (or non-blocking)

- **non-blocking send** - sender sends the message and continues
- **non-blocking receive** - receiver receives a valid message or null



Synchronous Message Passing

synchronous communication (or blocking)

- **blocking send** - sender is blocked until the message is received
- **blocking receive** - receiver is blocked until a message is available

