

Sistemas Distribuidos

LLAMADAS A PROCEDIMIENTOS REMOTOS

Generalidades de las Llamadas a procedimientos

Una llamada a un procedimiento (función o subrutina) es un método bien conocido para **transferir el control de una parte del programa a otra**, con un retorno del control a la primera.

Asociado con la llamada a un procedimiento están el pase de **argumentos** y el **retorno de uno o varios resultados**.

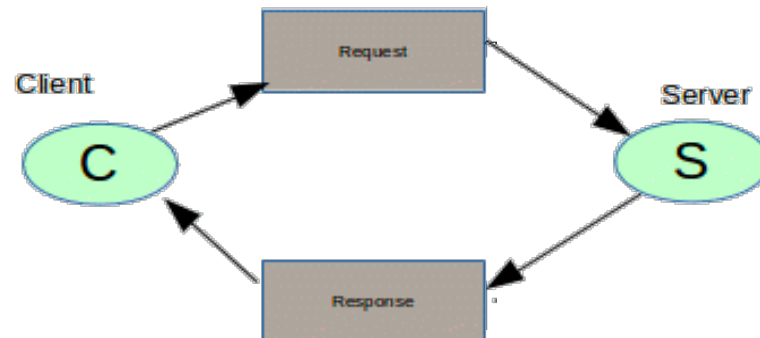
```
.....
resultado = invertir(num);
.....
int invertir(int num)
{
    int inverso = 0, cifra;
    while (num != 0)
    {
        cifra = num % 10;
        inverso = inverso * 10 + cifra;
        num = num / 10;
    }
    return inverso;
}
```

Cuando el código que invoca a un procedimiento y dicho procedimiento están en un mismo proceso en una computadora dada, se dice que ha ocurrido una **llamada a un procedimiento local**.

Llamadas a procedimientos remotos

En una llamada a un procedimiento remoto (RPC, Remote Procedure Call) el sistema local invoca, a través de la red, a una **función alojada en otro sistema**.

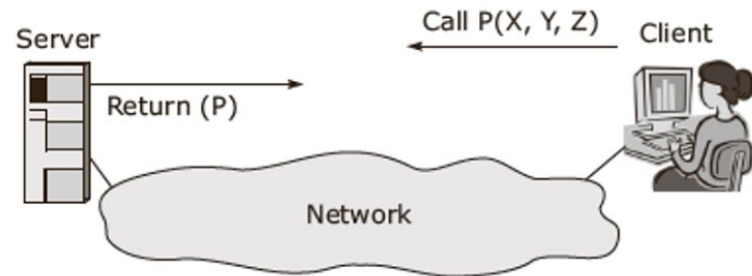
Lo que se pretende es hacerle parecer al programador que está ocurriendo una simple llamada local.



Se utiliza el término **solicitud** para referirse a la llamada que realiza el cliente al procedimiento remoto (servidor) y, el término **respuesta**, para describir el resultado devuelto por éste último.

Llamadas a procedimientos Remotos (RPC)

- ❑ Permiten a los programas llamar procedimientos localizados en otras máquinas.
- ❑ Un proceso X en una máquina A (cliente), puede llamar un procedimiento localizado en una máquina B (servidor).



Llamadas a procedimientos Remotos (RPC)

Características de RPC

- ❑ **Bases del paradigma cliente/servidor.** Es una técnica para el desarrollo de aplicaciones distribuidas basadas en el paradigma cliente/servidor.
- ❑ **Se basa en el empaquetado.** La información puede llevarse del proceso invocador al invocado dentro de los parámetros de envío y respuesta.
- ❑ **Transparencia.** Ningún mensaje u operación de E/S es visible para el programador.

Llamadas a procedimientos Remotos (RPC)

❑ Problemas a resolver:

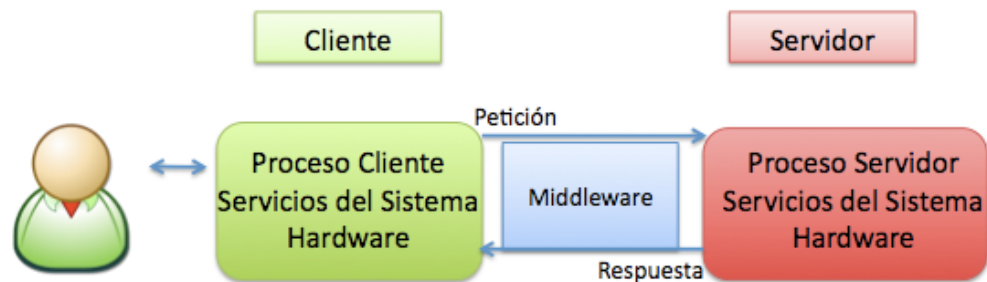
- Procedimiento invocador e invocado se ejecutan en diferentes máquinas, i.e. diferentes direcciones y posiblemente diferentes arquitecturas.
- Ambas máquinas pueden fallar
 - El servidor no está disponible: fallo del servidor.
 - El cliente no puede localizar el servicio



Llamadas a procedimientos Remotos (RPC)

Componentes de un sistema basado en RPC

- El cliente
- El servidor
- Middleware



Llamadas a procedimientos Remotos (RPC)

□ El cliente

- Es el proceso que realiza la **llamada** a una función.
- Envía los parámetros de la petición.
- Queda en **espera** del resultado.
- Recibe resultado

Llamadas a procedimientos Remotos (RPC)

□ El servidor

- Debe estar *escuchando*, es decir en espera de una petición.
- **Recibe** un mensaje que consiste de uno o varios argumentos.
- Los argumentos son usados para llamar a un **procedimiento** en el servidor.
- El resultado del procedimiento se envía al middleware.

Llamadas a procedimientos Remotos (RPC)

❑ Middleware

- **Empaqueta** los argumentos de la petición del cliente.
- Envía el paquete por el canal de comunicación al servidor.
- Recibe la respuesta del servidor, empaquete el resultado y se envía de regreso al cliente por el canal de comunicación.

Algunos paquetes RPC ya manejan el middleware de manera transparente.

Llamadas a procedimientos Remotos (RPC)

¿Qué es un *stub* cliente o servidor?

- ❑ *Middleware cliente y Middleware servidor.*
- ❑ Es una pieza de código usada para **convertir los parámetros** enviados entre el cliente y el servidor en una llamada a procedimiento remoto.

Empaquetado y desempaquetado

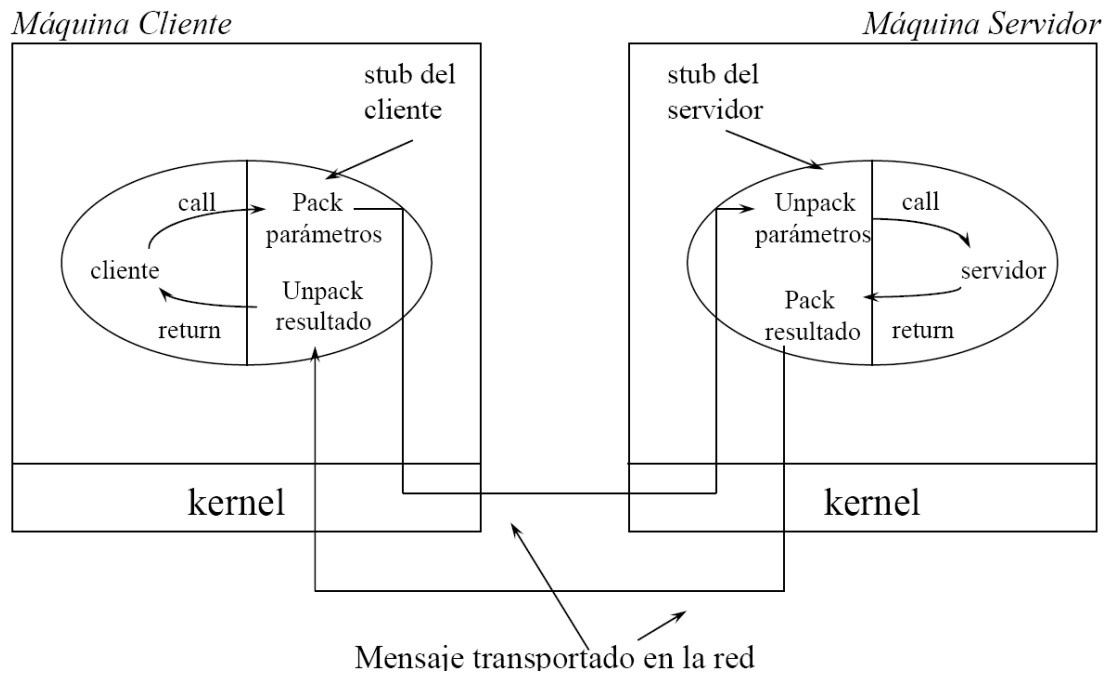
Llamadas a procedimientos Remotos (RPC)

Operaciones básicas de un sistema basado en RPC

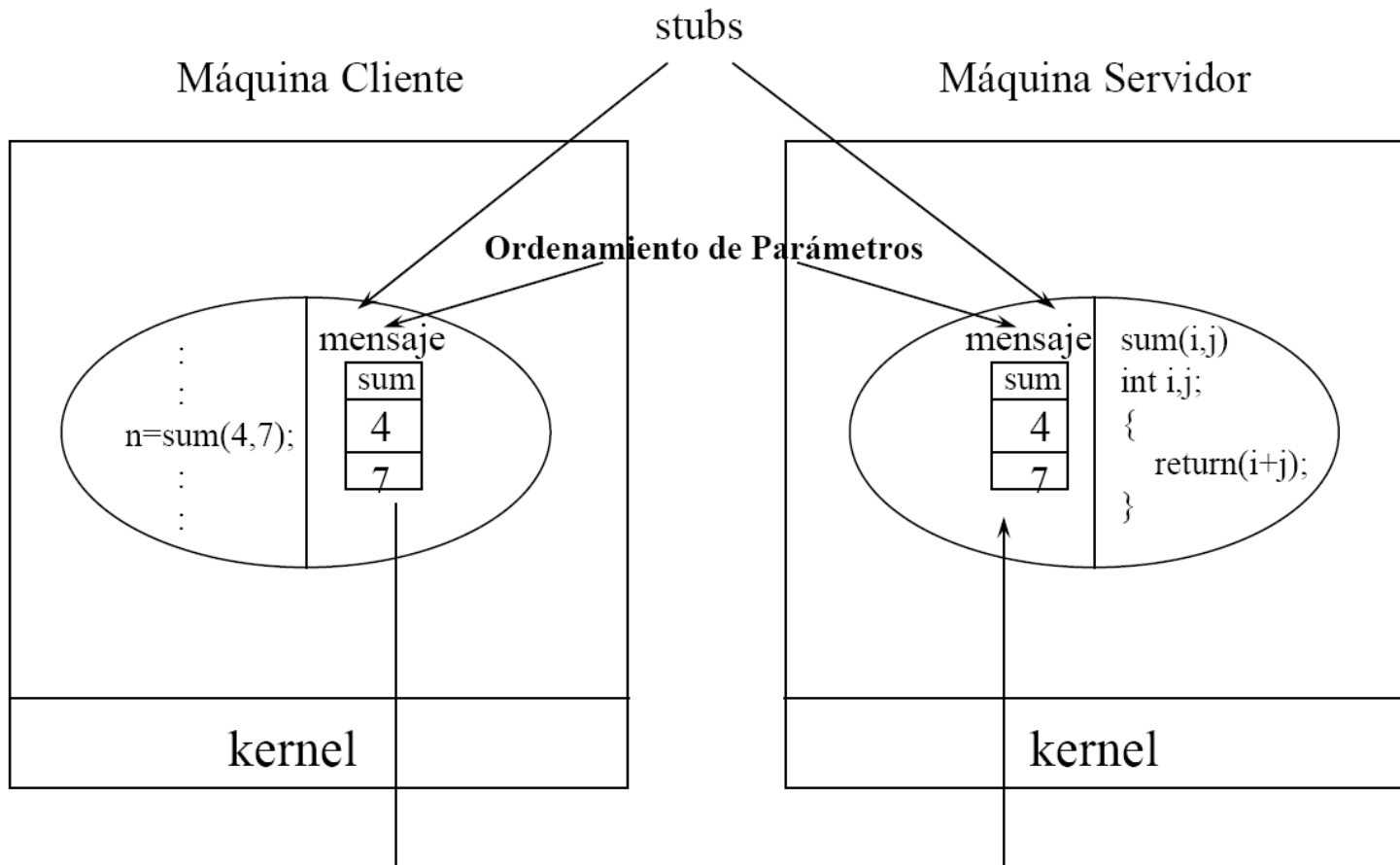
- ❑ **Publicar servidor.** El procedimiento llamado debe estar cargado en memoria.
- ❑ **Enviar parámetros.** El cliente debe colocar los parámetros como mensajes en el Middleware
- ❑ **Transmitir parámetros.** El Middleware transmite los parámetros por el canal de comunicación (Empaquetado o desempaquetado)
- ❑ **Recibir parámetros.** El servidor tomará el mensaje que contienen los parámetros.
- ❑ **Ejecutar operación.** Toma los parámetros y ejecuta la operación.

Llamadas a procedimientos Remotos (RPC)

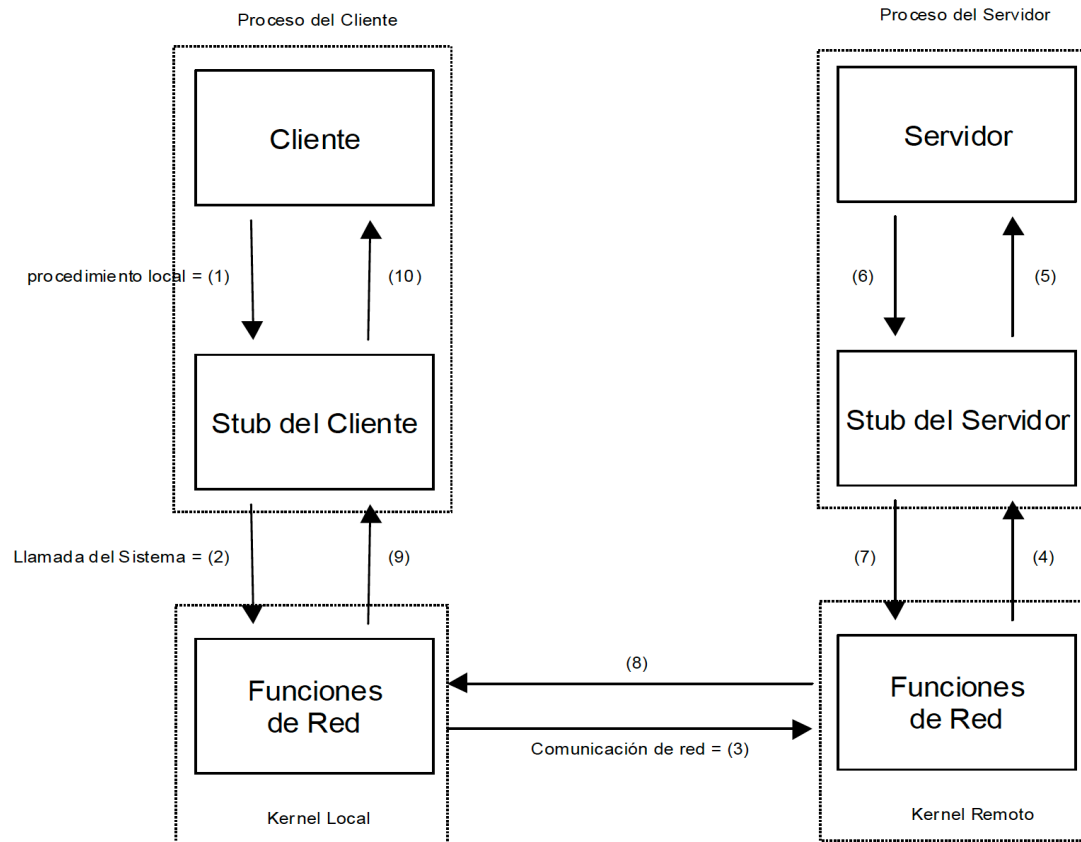
Arquitectura de un sistema basado en RPC



Llamadas a procedimientos Remotos (RPC)



Llamadas a procedimientos Remotos (RPC)



Llamadas a procedimientos Remotos (RPC)

Pasos de un RPC considerando los *stubs*

1. El cliente **llama** al *stub-cliente*
2. El *stub-cliente* **construye** un mensaje y hace un señalamiento al núcleo local.
3. El núcleo local **envía** el mensaje por el canal al núcleo remoto.
4. El núcleo remoto **recibe** y **proporciona** el mensaje al *stub-servidor*.
5. El *stub-servidor* **desempaqueta** el mensaje y llama al servidor.
6. El servidor **realiza** el trabajo y **devuelve** el resultado al *stub*.
7. El *stub-servidor* **empaqueta** el resultado y hace un señalamiento al núcleo.
8. El núcleo remoto **envía** el mensaje por el canal al núcleo local.
9. El núcleo local **proporciona** el mensaje al *stub-cliente*
10. El *stub-cliente* **desempaqueta** el resultado y lo regresa al cliente.

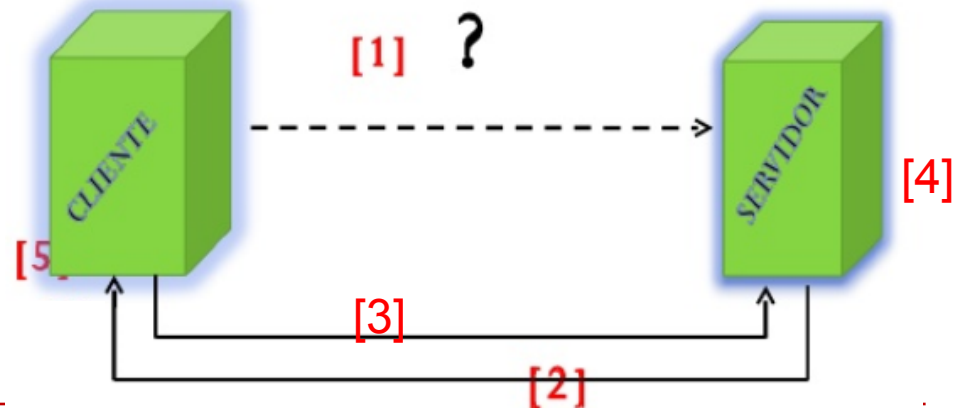
¿Qué puede fallar en un RPC?

Posibles fallos

Llamadas a procedimientos Remotos (RPC)

Fallos del cliente

1. El cliente es incapaz de localizar al servidor.
2. El mensaje de respuesta del servidor al cliente se perdió.
3. El mensaje de petición del cliente al servidor se perdió.
4. El servidor falló después de recibir una petición.
5. El cliente falló después de enviar una petición.



Llamadas a procedimientos Remotos (RPC)

Fallos del servidor

1. El servidor podría estar inactivo.
2. El servidor podría estar utilizando una versión diferente del procedimiento al ser publicado.
3. El *stub* del servidor puede no coincidir con el *stub* del cliente.
4. Un procedimiento del servidor no está regresando valor.

Llamadas a procedimientos Remotos (RPC)

Pérdidas de mensaje

1. Se puede emplear alertar y **retransmisiones**, pero se debe considerar:
 1. ¿Se perdió la petición?
 2. ¿Se perdió la respuesta?
 3. ¿el servidor va lento?
2. Se utiliza un **cronómetro** para determinar la pérdida de mensajes.
3. Si no llega una respuesta en un **tiempo razonable** se tiene que enviar la solicitud nuevamente.

Resumen

1. Explicar el concepto de Llamadas a procedimientos remotos (RPC).
2. Mencionar las características de los RPC
3. Menciona los componentes principales de un sistema basado en RPC.
4. Menciona las operaciones básicas de un sistema basado en RPC
5. Mencionar los 10 pasos de un RPC
6. Menciona las fallas posibles en el servidor
7. Mencionar las posibles fallas en el cliente