

# Clase 8- RPC y Rendezvous

jueves, 17 de octubre de 2024

10:35

- Pasaje de mensajes es comunicación unidireccional
- Para cliente servidor necesito 2 canales (requerimientos y respuestas)
- **RPC (Remote Procedure Call) y Rendezvous P técnicas de comunicación y sincronización entre procesos que suponen un canal bidireccional P ideales para programar aplicaciones C/S**
  - a interfaz "tipo monitor" con operaciones exportadas a través de llamadas externas (CALL) con mensajes sincrónicos (demoran al llamador hasta que la operación llamada se termine de ejecutar y se devuelvan los resultados).
- RPC declara un procedure para cada operación y crea un proceso para manejar cada llamado. RPC pq llamador y procedure pueden estar en distintas máquinas.
- rendezvous con un proceso existente. Un rendezvous es servido por una sentencia de Entrada (o accept) que espera una invocación, la procesa y devuelve los resultados (Ej:Ada)

## RPC

Programas descompuestos en módulos con procesos y procedures que pueden estar en lugares distintos. Procesos de un módulo pueden compartir variables y llamar a procedures dentro del módulo y se puede comunicar con otros módulos invocando procedimientos que exporte

Un llamado intermódulo es distinto que para uno local pq pueden estar en distintos lugares. Un nuevo proceso sirve el llamado y los argumentos se pasan como mensajes entre el llamador y el sv

El llamador se demora mientras el servidor ejecuta el cuerpo del procedure

Cuando el sv vuelve del otro módulo manda los rtas al llamador y termina, sigue el llamador

Si el llamador y el otro módulo están en el mismo lugar se puede evitar hacer un nuevo proceso

**Llamado en general es remoto: se crea un proceso server**

RPC es solo un mecanismo de comunicación

El único rol del sv es actuar en nombre del llamador. La sincronización es implícita

Existen dos enfoques para proveer sincronización, dependiendo de si los

procesos en un módulo ejecutan

**Con exclusión mutua (un solo proceso por vez).**

Las variables compartidas se protegen automáticamente del acceso concurrente pero hay que programar sincronización x condición

**Concurrentemente.**

Se necesitan mecanismos para programar exclusión mutua y sincronización por condición. Se puede usar semáforos, monitores o rendezvous.

- Si dos procesos de diferentes módulos deben intercambiar valores, cada módulo debe exportar un procedimiento que el otro módulo llamará.

```
module Intercambio [i = 1 to 2]
  op depositar(int);
body
  int otrovalor;
  sem listo = 0;

  proc depositar(otro)
  { otrovalor = otro;
    V(listo);
  }

  process Worker
  { int mivalor;
    call Intercambio[3-i].depositar(mivalor);
    P(listo); .....
  }
end Intercambio
```

## Rendezvous

RCP da mecanismo de comunicación entre módulos. Dentro de un módulo hay que arreglar sincronización

Rendezvous da comunicación y sincronización:

- Proceso cliente invoca operación mediante call, pero esta operación la da un proceso que ya existe NO UNO NUEVO
- Proceso servidor usa sentencia de entrada para esperar call y actuar
- Operaciones se atienden de a una

El sv es un proceso activo