

Comentarios al apunte de Martinez

Si bien el apunte introduce desde la base a la programación cliente servidor, es conveniente destacar algunas diferencias conceptuales con un Sistema Distribuido REAL, que han sido soslayadas en aras de simplificar la comprensión, a saber:

Los parámetros definidos para las funciones `ropen()`, `rread()`, `rwrite()` y `rclose()` no son conceptualmente correctos. Si se mira lo de RPC se puede ver fácilmente: el cliente y el servidor en esta primera versión comparten memoria, algo que no se da en los sistemas distribuidos y seguro no es posible en los sistemas c/s con TCP/IP. Viola una de las características generales de las aplicaciones distribuidas: memoria distribuida.

Todo un problema también conceptual que el stub del cliente llame explícitamente a una función del servidor y otra vez con los datos compartidos. Viola otra de las características generales de las aplicaciones distribuidas: concurrencia.

Algo que puede confundir, aunque menor: la cantidad de datos a comunicar que se utiliza en la función de comunicaciones es asimétrica. Siempre se conoce del lado del que envía y del lado del que recibe se utiliza el tamaño total del buffer.

Algunos detalles que son "menores" pero pueden confundir: en el apunte se trabaja con `fd` en el servidor, mientras que en c/s con TCP/IP se usa la estructura (y puntero a) `FILE`.

El tratamiento de la heterogeneidad es un poco "raro" al definir `RINT_T` asumiendo que es `short int` que por experiencia del autor siempre es de 16 bits... me parece mejor asumir que es siempre igual o directamente se usa una representación externa.

Algo similar a lo anterior pasa con lo de la alineación/almacenamiento de las estructuras, a lo que también se da por resuelto con no habilitar optimizaciones del compilador y miembros de estructuras "no inferiores de short". En realidad, tanto esto como la propia representación se resuelven juntos, con XDR o traducción de representación.