

Channels y Actores

Técnicas de Programación Concurrente

Channels

Implementar el problema original de banquero utilizando channels.

Recordamos:

Al inicio de cada semana les envía el dinero para invertir a sus amigos; A todos los amigos les envía el mismo monto.

Luego espera hasta el final de la semana a que le envíen el resultado de esas inversiones.

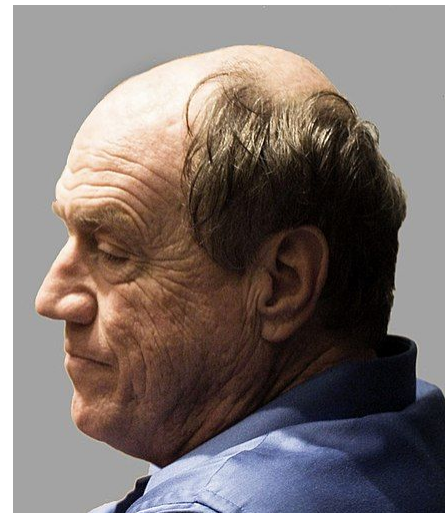
Channels

Ahora la AFIP obliga a todos los inversores a notificar los resultados de las inversiones, como afecta nuestro programa?

Actores

Vamos con unos ejemplos introductorios con Actix

- Hola mundo!
- Actores con estado interno
- Uso de sleep



Carl Hewitt
Paper original de **1973!**

Problema del banquero

Solucionemos el problema de banquero utilizando actores

- Qué actores necesitamos?
- Qué tipos de mensaje?

Tips:

- El progreso de un proceso asíncrono se sigue/controla desde el estado interno del actor
- Recursión de actores

Filósofos

Cómo podemos implementar el problema de los filósofos usando pasaje de mensajes?

Filósofos

- Un "mozo" que provee los palitos. Todos los filósofos hablan con el mozo.
- Channels entre cada par de filósofos adyacentes, por los que se pueden pasar palitos
- Thread propio de cada palito.

Filósofos

Solución de Chandy/Misra

- For every pair of philosophers who contend for a chopstick, one chopstick is created, assigned to the philosopher with the lower ID number (**deadlock avoidance**).
- Each chopstick can either be *dirty* or *clean*. Initially, all chopsticks are dirty.
- When a philosopher wants to use a set of resources (*i.e.*, eat), said philosopher must obtain the chopsticks from their contending neighbors. For all such chopsticks the philosopher does not have, they send a request message.
- When a philosopher with a chopstick receives a request message, they keep the chopstick if it is clean, but give it up when it is dirty. If the philosopher sends the chopstick over, they clean the chopstick before doing so.
- After a philosopher is done eating, all their chopsticks become dirty. If another philosopher had previously requested one of the chopsticks, the philosopher that has just finished eating cleans the chopstick and sends it.



Referencias

<http://www.opencsf.org/Books/csf/html/CigSmokers.html>

<https://greenteapress.com/semaphores/LittleBookOfSemaphores.pdf>

<https://www.ijcai.org/Proceedings/73/Papers/027B.pdf>

<https://www.cs.utexas.edu/users/misra/scannedPdf.dir/DrinkingPhil.pdf>