

Sincronización

Técnicas de Programación Concurrente

Barriers

Han habido algunos problemas de organización, y los hijos del señor banquero, si bien desean que los inversores sigan trabajando de forma autónoma, deben tomar el dinero de la cuenta y devolverlo al final de la semana, análogamente a como se hacía cuando el señor banquero se encontraba vivo. Recordamos: cada inversor debe tomar exáctamente el mismo dinero.

El problema del barbero

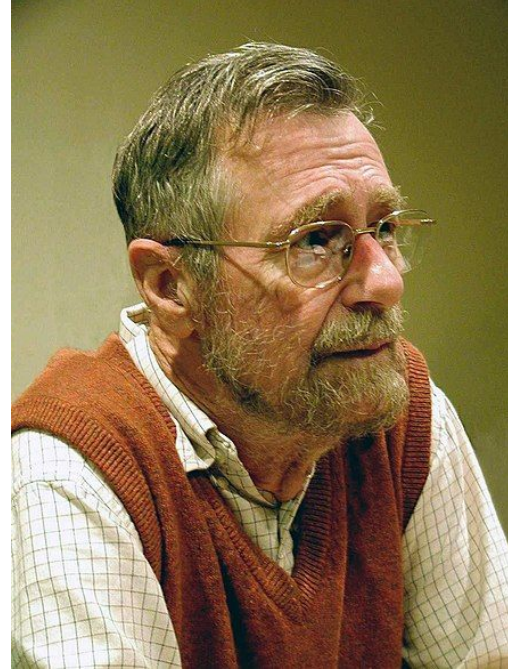
Una barbería tiene una sala de espera con sillas.

Si la barbería está vacía, el barbero se pone a dormir.

Si un cliente entra y el barbero está durmiendo, lo despierta.

Y si el barbero está atendiendo, se sienta en una de las sillas y espera su turno.

El cliente espera a que le corten el pelo

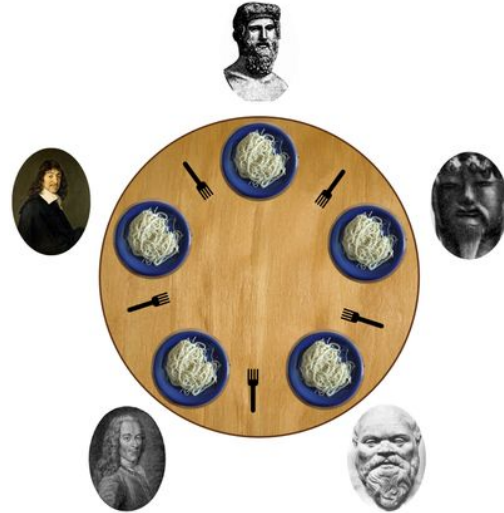


Edsger Dijkstra

Problema de los Filósofos

Cinco filósofos se sientan alrededor de una mesa y pasan su vida cenando y pensando.

Cada filósofo tiene un plato de fideos y un palito chino a la izquierda de su plato. Para comer los fideos son necesarios dos palitos y cada filósofo sólo puede tomar los que están a su izquierda y derecha. Si cualquier filósofo toma un palito y el otro está ocupado, se quedará esperando, con el palito en la mano, hasta que pueda tomar el otro palito, para luego empezar a comer.

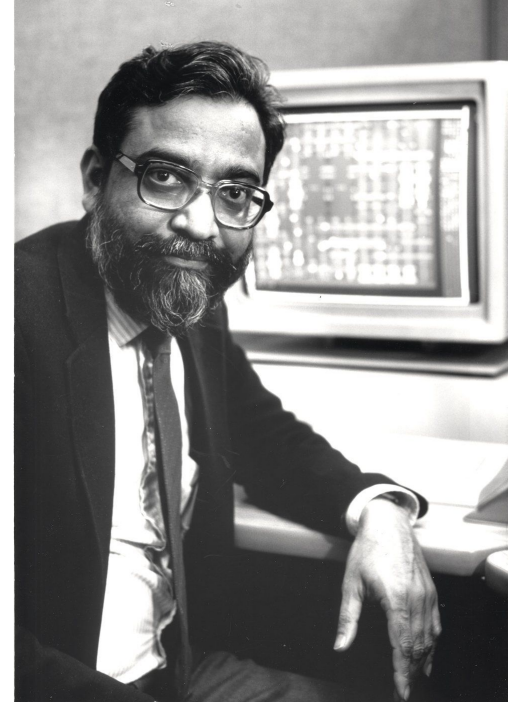


Problema de los fumadores

Consideremos que para fumar un cigarrillo se necesitan 3 ingredientes: tabaco, papel y fósforos.

Hay 3 fumadores alrededor de una mesa. Cada uno tiene una cantidad infinita de uno solo de los ingredientes y necesita los otros dos para fumar. Cada fumador posee un ingrediente distinto.

Existe además un agente que pone aleatoriamente dos ingredientes en la mesa. El fumador que los necesita los tomará para hacer su cigarrillo y fumará un rato. Cuando el fumador termina, el agente pone otros dos ingredientes.



Suhas Patil

Problema de los fumadores

Smoker with Tobacco	Smoker with Paper	Smoker with Match
<code>sem_wait(match_sem);</code> SUCCESS	<code>sem_wait(tobacco_sem);</code> SUCCESS	<code>sem_wait(paper_sem);</code> BLOCKED
<code>sem_wait(paper_sem);</code> BLOCKED	<code>sem_wait(match_sem);</code> BLOCKED	<code>sem_wait(tobacco_sem);</code> BLOCKED

Condvars

Ejemplos:

- Hello world
- Spurious Wakeup
- Wait on multiple
- Multiple Waiters

Problema del Lector-Escritor

Un estado se comparte entre varios procesos.

Algunos procesos necesitan actualizar dicho estado, mientras que otros solo necesitan leerlo.

Mientras que un proceso está leyendo el estado, otros pueden leerlo, pero ninguno modificarlo

Mientras que un proceso está modificando el estado, ningún otro puede leerlo ni modificarlo.



Pierre Jacques Courtois

Problema del Lector-Escritor

Process 1	Process 2	Allowed / Not Allowed
Writing	Writing	Not Allowed
Reading	Writing	Not Allowed
Writing	Reading	Not Allowed
Reading	Reading	Allowed

Problema del Lector-Escritor

- Starvation
- Writer Preference
- Fair

Referencias

Sistemas Operativos Modernos, Tanenbaum, Capítulo 2.3

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

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

Concurrent control with "readers" and "writers" - <https://dl.acm.org/doi/10.1145/362759.362813>