

Sockets

Técnicas de Programación Concurrente

Hello World!

4.2BSD fue el primer sistema operativo en implementar TCP/IP y la api de sockets posix que llega hasta nuestros días. Lanzado en 1983

Rust hace gala de las zero-cost abstractions con una API de sockets muy sencilla y sintética



Mascota de BSD

Problema de la oficina y la impresora

En una oficina hay una impresora compartida

Habitualmente las personas que trabajan allí envían documentos para ser impresos. Sin embargo muchas veces se generan inconvenientes ya que varias personas envían documentos al mismo tiempo, y luego las impresiones se encuentran intercaladas en la bandeja de salida, entorpeciendo el trabajo.

Mutex: Centralizado

- Un proceso es elegido coordinador
 - Cuando un proceso quiere entrar a la SC, envía un mensaje al coordinador
 - Si no hay ningún proceso en la SC, el coordinador envía OK; si hay, el coordinador no envía respuesta hasta que se libere la SC
-
- Cómo se implementa?
 - Ventajas y desventajas

Algoritmo de Ricart–Agrawala

Cuando un proceso quiere entrar en una sección crítica, construye un mensaje con el nombre de la sección crítica, el número de proceso y el timestamp. Al recibir el mensaje:

1. Si no está en la CS y no quiere entrar, envía OK
2. Si está en la CS, no responde y encola el mensaje. Cuando sale de la SC, envía OK
3. Si quiere entrar en la CS, compara el timestamp y gana el menor
4. Cuando recibe OK de todos, puede entrar a la sección crítica



Algoritmo Token Ring

- Se conforma un anillo mediante conexiones punto a punto
- Al inicializar, el proceso 0 recibe un token que va circulando por el anillo
- Sólo el proceso que tiene el token puede entrar a la SC
- Cuando el proceso sale de la SC, continúa circulando el token, el proceso no puede entrar a otra SC con el mismo token

Tarea

Implementar el problema del banquero utilizando alguno de los 3 algoritmos.

Referencias

- Ricart, Glenn; Agrawala, Ashok K. (1 January 1981). "An optimal algorithm for mutual exclusion in computer networks". *Communications of the ACM*. 24 (1): 9–17 <https://dl.acm.org/doi/abs/10.1145/358527.358537>
- Tanenbaum. *Distributed Systems: Principles and paradigms*. Capítulo 6