

Computación Concurrente - Reporte de lectura 3

Andrés Urbano Guillermo Gerardo

8 de Noviembre del 2021

Capitulo 6 The Problem of the Deadly Embrace

En este nuevo capitulo empiezan abordar lo problemas que surgen en la cooperación entre procesos, surge con la idea sencilla de pensar que dos personas no pueden entrar simultáneamente a una puerta, debe haber una cooperación y comunicación entre las personas para entrar.

Empieza a contarnos que los procesos pueden tomar programas para realizar ciertos cálculos y esto puede tomar cierto tiempo de ejecución durante la cual la información deber estar almacenada en la computadora. Una vez que el proceso termine de realizar sus cálculos esa memoria será liberada para que otro proceso la ocupe y así sucesivamente. Este almacenamiento disponible se divide en lo llama "páginas" y nos habla acerca de la demanda actual del espacio libre, esto podría ser una función que varia en el tiempo como el proceso necesite. Nos muestra una tabla en la cual podemos ver la cantidad de demanda máxima que necesita un proceso como también el prestamos de paginas que da el sistema operativos a los procesos. Nos hace énfasis en el caso de que un proceso necesite más paginas de lo que hay disponible, esta situación la llama "The Deadly Embrace" que es cuando un proceso solo puede continuar matando a otro proceso primero para obtener los recursos necesarios para continuar, cada proceso están compitiendo a nivel de recursos.

También plantea el algoritmo del banquero que consiste con las ideas transmitidas anteriormente acerca de la memoria, es un banquero que posee un capital finito de florines para prestamos y puede prestar a los clientes siempre y cuando cumplan ciertas condiciones, uno es que el préstamo del cliente se utilizará para una transacción que se llevará a cabo en un tiempo finito, esto representa el gestor de memoria de un sistema operativo, cada proceso le pedirá prestado al gestor y el decidirá cederle más memoria de lo permitido. Otro supuesto es que siempre que el préstamo no exceda su necesidad máxima de florines, el cliente puede aumentar o disminuir la cantidad de préstamos, una analogía muy clara para nuestros procesos. Y el último supuesto es sí un cliente solicita un aumento en su préstamo original, este tendrá que esperar a que el banquero le pueda dar el aumento, si es que no tiene disponible florines en ese momento, esta analogía que el autor nos narra nos hace ver la importancia de administrar la memoria correctamente cuando ejecutamos varios procesos concurrentemente.

Capítulo 7 Concluding Remarks

En este capítulo el autor nos hace énfasis sobre las observaciones que encontramos cuando hablamos de programación concurrente como los términos relacionados. Nos dice sobre la confusión que hay cuando hablamos de multiprogramación y computación concurrente. Se dice que cuando más de un proceso opera sobre el mismo trabajo hablamos de concurrencia y multiprogramación es cuando un único proceso es dividido por cantidades de tiempo para realizar diferentes trabajos. Nos habla que el número de procesos secuenciales que tienen que cooperar unos con otros, también aplica igualmente a la concurrencia y la multiprogramación. Es importante mencionar que también hace énfasis sobre que debemos preocuparnos por la correctes de nuestros programas y nos dice que esta habilidad es más de psicología que matemáticas, y para justificarlo nos muestra un pequeño snippet de código sobre un problema, el cual menciona que al leerlo a primera vista podemos tener la impresión que cierto valor representa el número de veces que realiza tal operación. Al final nos menciona que es importante especificar que escenarios los procesos son aplicables y nos da una visión general acerca como de entender e interpretar los programas.