# Lecturas computo concurrente.

Alvarado Morán Óscar Anuar
IIMAS, UNAM
Computación concurrente Grupo: 001

10 de Septiembre de 2019

Si me preguntaran cuál de las dos lecturas me gustó más, diría que aquella que es más extensa, llamémomsle -Lectura 1-, por lo que el artículo de unas cuantas páginas será la -Lectura 2-. Aunque en un principio pensé que la Lectura 1 sería tediosa y aburrida, creo que tiene una agradable forma de llevar los conceptos de la mano con el lector, ya que, al parecer va dirigida hacia un grupo de estudiantes, sea cual sea el nivel.

No sé si mis pocos conocimientos en computación ayudaron o repercutieron en el entendimiento de alguna de las dos lecturas, pero creo que se pueden tornar un poco confusas; por ejemplo, para la Lectura 1 creo que empieza muy bien con la explicación de su -máquina- para obtener el número máximo de un conjunto de muchos números, sin embargo, al querer pasar a la siguiente seccción lo sentí como 'por otro lado...' y me creí desubicado, aunque después de alguna manera intenta juntar los conceptos de ambas secciones, para este punto estaba confundido, ya que la segunda sección la empieza de manera extraña y va refinando esta extrañez hasta hacer su método de cómputo concurrente más 'seguro' como el lo dice, aunque al final del capítulo, agradecí que hubiera empezado como lo hizo, ya que su método final es extenso y puede confundir más, así, desde el principio te plantea bien la idea. Después, al intentar generalizar este método que en principio era con dos variables (procesos), más preguntas surgieron: ¿Ahora qué variable va a hacer que otro proceso pueda empezar? ¿Es esto en verdad cómputo concurrente tal y como lo hemos visto en clas? ¿El hacer y ejecutar este código en realidad todos estos procesos se activan al mismo tiempo? ¿Qué es cómputo concurrente en realidad? ¿Por qué no hice mi tarea antes para poder leer y entender esto con más calma? En esa parte me parece interesante que describe lo que significa un 'for' y un 'and', después de su método de computación concurrente es de cierta forma hasta un chascarrillo.

Creo que a final de cuentas todo lo visto en la sección 2 es más bien una introducción al cómputo concurrente y no una ejecución en realidad. Más bien el objetivo, al menos para mí, fue checar esto de la exclusión entre procesos y creo que de igual forma esto es parte bastante importante en la Lectura 2. Como hemos visto en clase, tiene sentido esto, ya que siempre se suele hacer la ejecución de distintos procesos pero independientes, es decir, no te pones a calcular cosas que dependan de otras, ya que tendrás que parar para obtener el resultado de esta otra, por lo que ya no se estarían ejecutando los procesos concurrentemente.

Al llegar a la sección tres se puede ver por qué la forma de definir los procesos en la sección 2. Se definen todas estas variables que cambiaban entre 0 y 1 como "semáforosz las operaciones que cambian a estos enteros las define como operaciones P y V. Creo que es muy explicativo el término -semáforo- dada la funcionalidad que se le estuvo dando. Por otro lado, las operaciones P y V, por el contrario, no nos dicen nada por su nombre, pero en escencia entiendo que son funciones que decrecen o incrementan en 1 el valor del semáforo sobre el que estén actuando, respectivamente. Como lo explican en la Lectura 1, el uso de estas funciones tienen sentido en el ámbito de programación concurrente. Porque cualquiera diría que se podría usar un '$a+=1$' o algo por el estilo, pero es error de la forma de pensar del humano, secuencialmente.

Parece ser que para el capítulo 4 se intentan formalizar lso conceptos ya vistos en la sección 2. Por un lado, en el capítulo 3 ya se vieron los semáforos y las funciones mencionadas, ahora, en el capítulo 4 se presenta el concepto de 'bufffer', que como lo entiendo, es el intermediario de paso de mensajes entre dos procesos; éste se rige por el mantra de vida FIFO (First-In-First-Out), que resulta evidente por la forma en que se trabaja con el programa de esa seccion. El productor y el consumidor deben de estar comunicado spor algo (el buffer) y el consumidor debe consumir algo que previamente hizo el productor en el orden en que se hace.

Me agradó esa explicación que se da sobre el barbero y los clientes. Creo que expresa bien la idea del código que se tiene en dicha sección que en realidad no es muy claro o hay que revisar con atención línea por línea para entender bien. Por lo que como dije, el autor se jacta de poder o al menos intentar traer de traer vuelta al lector a la realidad. Se aprecia bastante ya que en realidad para mí no había sido nada claro el código realizado en la sección 4.

Todo lo anteriormente visto nos lleva (o ya nos trajo) a la importancia de la comunicación entre los procesos. En la sección 4 se ve una comunicación de una sola vía. Después, evidentemente se tuvo que generalizar el problema del consuimidor y del productor. Empezando por productos de diferentes tamaños, más consumidores y más productores. Para este punto del texto, las variables agregadas al programa van creciendo bastante, metiendo ahora 'variables de estado' o de 'bloqueo de buffer', pero creo que en general se va sobre entendiendo todo con el apoyo de los ejemplos. Por ejemplo, en la sección cinco, en su mayoría es la expliación de un algoritmo de envío de mensajes y uno de los mayores problemas del cómputo concurrente: la comunicación para el recibimiento y distribución de mensajes, que forzosamente pasan por un cuello de botella.

Al llegar a la parte en que mete los 'mutex' me pareció algo confuso todo, ya no entendí bien qué significan estos y mucho menos al entrar a la parte del código, ya se salió de control todo, ocupa tres páginas completas con muchas variables (entre éstas, unos semáforos) que no son claras.

Para eso de la sección seis y sus cosas del 'abrazo de la muerte', el banco y los florines, me pareció que otra vez se volvía al mundo real, con ligaduras entre semáforos y buffers, sin embargo, para esta sección ya estaba perdido respecto al código y en realidad no sabría explicar qué hacía cada cosa, pero por lo que entendí, es básicamente lo mismo que del peluquero y el stock, es decir, hay cosas en el stock, entonces hay producto por comprar, y si no hay, entonces hay que esperar. Todo señalizado con semáforos para un buen control de dichos procesos.

Por otro lado, para la Lectura 2, aunque 'pequeña', puede llegar a ser bastante engorrosa. Pero en general, hace un énfasis a que, en el cómputo concurrente, es importante la disjunción entre procesos. Este lo leí dos veces, la primera porque pensé que sería más fácil leer la lectura corta, lo que me llevó a la desesperación. Al leerla por segunda vez después de haber leido la Lectura 1 me pareció más clara. Creo que en realidad está describiendo algunas cosas de la Lectura 1 pero con otra notación. El Algoritmo 3 no me es muy claro, aunque segùn esto, es el bueno para la eficiencia de tiempo y cuidado de recursos. También me surge la posibilidad sobre si de aquì saliò todo lo de cómputo concurrente, sabiendo que ahí hablan sobre que esos algoritmos lo usan los SO. Creo que es buena idea haber dejado estas dos lecturas juntas, ya que en cierto sentido sí se complementan, sin embargo, hasta donde sé, no se ha visto nada de esto en clase como tal (semáforos), entonces creo que una explicación previa habría estado bien para haber tenido esa familiarización con el tema.