**Mini Trabajo Práctico Individual N°2:**Threads y Semáforos

**Sistemas Operativos y Redes**



Nombre: Luis Angel Alcarraz ([luchinico2010@gmail.com](mailto:luchinico2010@gmail.com))

Legajo: 38661651/2014

Comisión: COM 01

Turno: Noche

Profesores:

Mariano Vargas

Noelia Sosa

Fecha de entrega: 15/10/20 (limite 18:00 hs.)

Introduccion:

En el presente trabajo nos propondremos a realizar la lectura de un archivo txt sobre la receta de la preparación de hamburguesas. Se leera dicho archivo desde un archivo .c , donde utilizaremos threads y semáforos , los cuales nos ayudaran a simular la competencia entre tres equipos con la dificultad de que la cocina no cuenta con equipamiento individual para cada equipo, como por ejemplo compartir el salero, ya que hay uno solo, el horno solo puede cocinar dos panes a la vez, y una plancha que solo puede cocinar un medallón (Siempre se debe respetar el orden de la receta). El resultado se plasmara en otro archivo txt, que tendrá al equipo ganador y todos los procedimientos de cada equipo hasta finalizado el mismo.

Desarrollo:

Para empezar defini los mutex, luego una variable de tipo entera, para saber si alguien gano y luego creo la estructura de los semáforos, y dentro de esa esctructura, coloco todos los semáforos que voy a utilizar (mis pasos a seguir para terminar la hamburguesa). Tambien se creo una estructura paso, que tendría la “acción” a los ingredientes que están en un array. Luego aparece una función llamada imprimirAccion que va a mostrar en pantalla las acciones y los ingredientes de la acción. Luego de esto se crean las funciones:

* Cortar: seria la acción de cortar el ajo,la cebolla y el perejil, uso usleep para simular que pasa un cierto tiempo y da la señal a la siguiente acción (mezclar)
* Mezclar: seria la acción de mezclar la carne picada con los ingrendientes anteriormente dichos, también utilizamos un sleep y da la señar a la siguiente acción (salar).
* Salar: Interrumpe el semáforo, para que otro equipo no pueda usar el salero, hace la acción de salar la mezcla y deja el salero para que otro equipo ya pueda usarlo, por ultimo da señal para armar los medallones.
* Hornear: Verifica que el horno no este ocupado y simula la acción de hornear los panes, y da un signal con los panes ya listos.
* Armar: Simula la acción de armar los medallones.
* Planchar: Ve si el semáforo de la plancha esta libre (como en la acción anterior), se usa el usleep para simular un tiempo mas y luego le da un signal al semáforo cuando los medallones están listos.
* Preparar: seria cortar la lechuga y el tomate y le avisa al semáforo que ya están listos para poder armar la hamburguesa.
* Terminar:Verifica que los pasos anteriores estén todos terminados, es decir, los medallones y panes cocinados, y la lechuga y tomate cortadas, para poder simular la acción de armar las hamburguesas y asi poder guardar su numero de equipo en la variable entera “ganador”.

Luego en la función ejecutarReceta se lee el archivo de la receta y agrega las acciones y los ingredientes al arreglo de pasos de cada equipo. Despues crea los hilos y los inicializa para poder ejecutar las tareas con sus semáforos correspondientes.

Por último en el main se inicializan los mutex y se le asigna a cada hilo su estructura y el numero de su equipo. Tambien muestra por pantalla al ganador.

Una dificultad que se me presento en el trabajo fue programar con el lenguaje C, ya que nunca lo había utilizado a la hora de hacer funciones. Fue un trabajo de investigación para obtener mas información sobre este lenguaje. Sumado al template que nos dieron los profesores, se me complico un poco mas cuando lo vi, pero con el paso de los días, pude entender bien el código de ese template y pude interpretarlo mejor que antes.

Prueba de escritorio en pseudocódigo:

Mutex horno

Hornear

V(s\_paneslistos)

Mutex horno

V(sem\_mezclar)

Preparar:

V(s\_lechugaYTomate)

)

P(s\_medalloneslistos)

P(s\_panesListos)

P(s\_lechugayTomate)

Terminar;

P(sem\_planchar)

Mutex plancha

Planchar;

V(s\_medalloneslistos)

Mutex plancha;

P(sem\_armarMedallones)

Armar;

V(sem\_planchar)

P(sem\_salar)

Mutex salero;

Salar;

V(sem\_armarmedallones)

Mutex salero;

P(sem\_mezclar)

Mezclar

V(sem\_salar)

Cortar:

V(sem\_mezclar)