

Trabajo practico N°2 SOR

Semaforos e hilos

Leandro Nehemias Camperoz

12 de septiembre de 2021

Olimpiadas SOR

Con el objetivo de poner en practica los conceptos de sincronizacion y exclusion mutua entre hilos de ejecucion, en este trabajo practico se simulo una carrera entre 3 equipos de 2 integrantes cada uno. En esta carrera un participante debe correr hasta cierto punto y luego pasarle la posta a su compañero para que recorra una distancia igual. Cuando el segundo integrante del equipo termine su recorrido se considera que a llegado a la meta.

Soluciones

- Simulamos la distancia a correr los participantes con un ciclo que realiza 2147483647 iteraciones sumando 1 a una variable inicializada en 0.
- Los participantes son simulados con hilos de ejecucion (pthread_t) y cada uno tiene un semaforo (sem_t).
- El primer hilo de cada equipo tiene habilitado su semaforo para realizar las operaciones, cuando termina habilita el semaforo de su compañero (segundo hilo).
- La llegada a la meta tambien se simula con un semaforo que es compartido entre los 3 equipos.
- Se muestran mensajes por pantalla cuando se habilita el segundo semaforo y el segundo hilo de ejecucion termina las operaciones y bloquea el semaforo de llegada.

Errores, problemas y dificultades

- **Contadores y representacion de los atletas.**

Los hilos de un mismo equipo utilizaban el mismo contador para realizar las operaciones, el primero realizaba la mitad de las operaciones y el segundo el resto.

Para que los atletas realizaran las operaciones se definio una funcion para cada uno y se indentificaban mediante muestras por pantalla.

- **Errores con semaforos.**

Se inicializaban todos los semaforos en 1 haciendo que todos los hilos se ejecutaban al mismo tiempo.

Se intento utilizar un solo semaforo por cada equipo, pero muchas veces se ejecutaba primero el segundo hilo del equipo, llegando a la meta antes de pasar la posta.

- **Llegada a la meta.**

En un principio no se usaba un semaforo para por lo que podian llegar a la meta varios hilos al mismo tiempo. Luego se implemento un semaforo por cada equipo pero no cumplia con el enunciado del trabajo practico.

Soluciones en la implementacion final

Para poder reutilizar codigo e identificar mejor a los atletas, se definio una estructura *Equipo* con los siguientes datos.

- 1 identificador para el equipo de tipo int.
- 2 punteros de tipo char para identificar a los atletas
- 2 contadores de tipo int.
- 3 punteros de tipo semaforo 1 para atleta y uno para la llegada a al meta.

Se definio una funcion para cada atleta de equipo. Estas funciones no son de tipo puntero como las de los ejemplos vistos en clase. Para poder usarlas en la funcion *pthread_create* se las castea con el tipo (*void **), lo mismo se hace cuando pasamos como argumento la estructura *Equipo*.

En las funciones se usa el argumento *struct Equipo *e* con el operador de union para acceder convenientemente a los datos y utilizarlos para hacer las operaciones, habilitar/bloquear semaforos y mostrar por pantalla cuando seceden los eventos de paso de posta y llegada a la meta.