SCD PRACTICA 1: PROBLEMA ESTANQUERO FUMADORES

ANTONIO GARCIA CASTILLO

Para resolver el problema del estanquero-fumadores he utilizado un semáforo que indicará cuando el estanquero puede servir, significando que el mostrador esta vacío, y un vector de semáforos que representa la disponibilidad de un ingrediente en concreto.
Sempahore mostrar_vacio = 1 ; → Este semáforo esta inicializado a 1 pues en primera instancia el estanquero debe poder servir, no tendría sentido esperar a que se retirara algo sin haberlo servido previamente. Sobre este semáforo actuara la hebra función_hebra_estanquero, haciéndole un sem_wait después de haber generado un numero aleatorio, y la función_hebra_fumador indicándole que puede volver a servir.

Semaphore ingr_disp[3] = {0,0,0}; → Este vector de semáforos tiene todos sus componentes inicializados a cero e incrementara sus componentes cuando se haya servido el ingrediente correspondiente sobre el mostrador. Sobre este semáforo actuará la hebra del estanquero, generando un sem_signal después de haber producido un valor arbitrario comprendido entre el 0 y el 2, y las 3 hebras de los fumadores, activándose cuando su ingrediente este disponible.

A continuación muestro el código explicando como funciona paso por paso.

```
//-------------------------------------------------------------------------
// función que ejecuta la hebra del estanquero
Semaphore mostrar_vacio = 1;      // semaforo inicializado a 1
Semaphore ingr_disp[3] = {0,0,0}; // vector de semaforos inicializados a cero

void funcion_hebra_estanquero()
{
  while(true){
  int num = -1;                   // Variable local inicializado a -1
  num=aleatorio<0,2>() ;          // generamos un valor aleatorio y lo almacenamos en num
  sem_wait(mostrar_vacio);        // Entramos y ponemos decrecemos mostrar_vacio a 0
  cout<< "Estanquero ha puesto "<< num << endl;
  sem_signal(ingr_disp[num]);     // Señal para que se active el semaforo correspondiente
  }
}

//-------------------------------------------------------------------------
// función que ejecuta las hebras de fumadores
void  funcion_hebra_fumador( int num_fumador )
{
   while( true )
   {
     sem_wait(ingr_disp[num_fumador]);  // Esperamos a que se active y una vez activado
                                        // entramos y lo volvemos a poner a cero
     cout<<"Cliente: "<< num_fumador << " ha retirado el ingrediente "<< num_fumador << endl;
     sem_signal ( mostrar_vacio);       // Activamos el semaforo mostrar_vacio para poder continuar
     fumar(num_fumador);                // Invocamos a la funcion fuamr
   }
}

//-------------------------------------------------------------------------
```