Jose Antonio Padial Molina

josepadial@correo.ugr.es



SISTEMAS CONCURRENTES Y DISTRIBUIDOS

PRACTICA 1

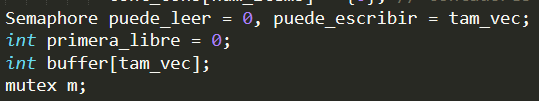
Contenido

[PRODUCTOR-CONSUMIDOR 2](#_Toc527135281)

[FUMADORES-PLANTILLA 3](#_Toc527135282)

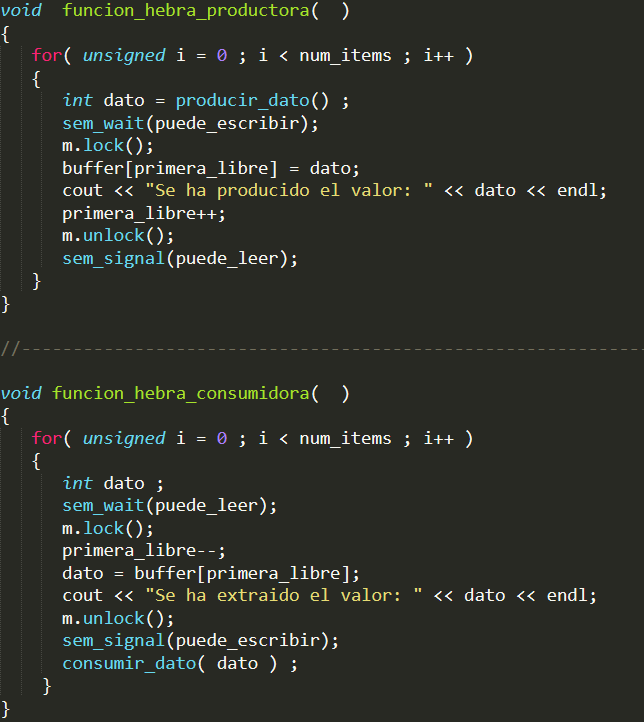
# PRODUCTOR-CONSUMIDOR

Como variables globales hemos creado los semáforos, un entero para controlar el buffer, un vector de enteros para usarlo de buffer y un mutex para usarlo de cerrojo.



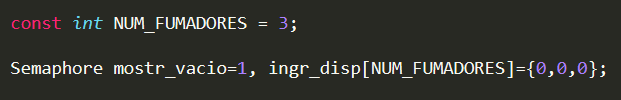
El semáforo puede\_leer se inicializa a ‘0’ para que no pueda leer sin antes haber producido un valor. En el caso del semáforo de puede\_escribir se inicializa al tamaño del vector ya que inicialmente está vacío el buffer y lo puede llenar entero.

La función hebra productora va a producir 40 elementos usando un bucle for desde 0 hasta 40. Donde se una la función producir\_dato() para obtener un entero aleatorio. Hacemos por seguridad un sem\_wait de puede\_escribir y bloqueamos el cerrojo para que no se interrumpa el proceso de producción. Al finalizar el proceso libreamos el cerrojo e indicamos que ya hay un elemento para consumir con sem\_signal a puede leer. La función consumir sigue la misma estructura solo que hace los semáforos de forma inversa.

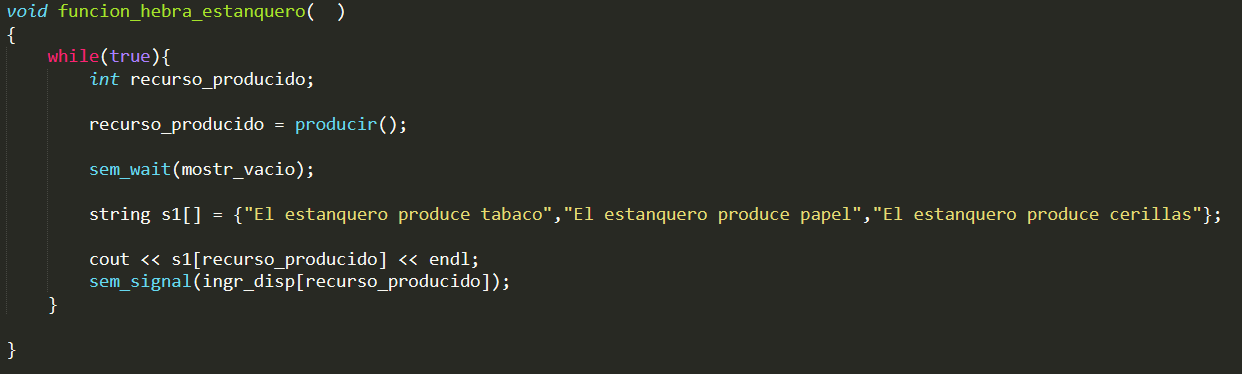


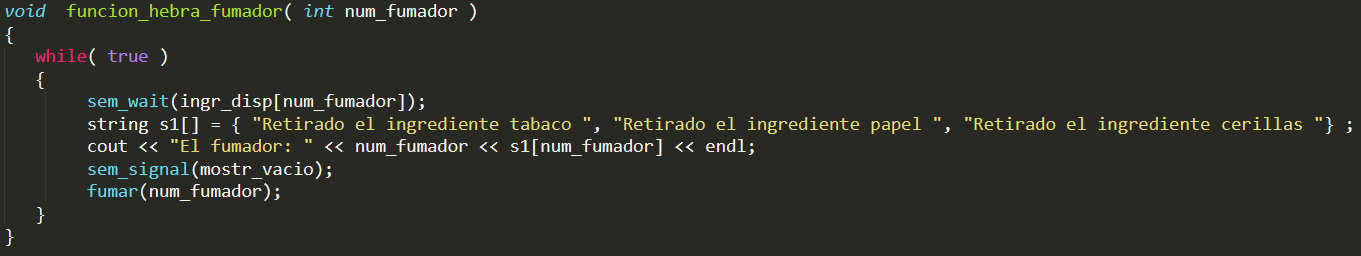
# FUMADORES-PLANTILLA

Como variables globales tenemos el número de fumadores y los semáforos que van a controlar el mostrador y los ingredientes disponibles.



La función hebra productora contiene un while que no contiene fin el cual va a producir un recurso, en la primera vez va a indicar que ya no está el mostrador vacío con un sem\_wait e indicamos que tenemos un igrediente disponible con sem\_signal. La función hebra fumador es idéntica a la anterior solo que se hace el sem\_wait a los ingredientes disponibles y el sem\_siganl al mostrador vacio.





En el main se crean las hebras de los fumadores con un bucle for y la hebra del estanquero.

