Práctica 1: Problema del Productor/Consumidor

Sistemas Concurrentes y Distribuidos

Realizado por: Daniel Díaz Pareja
Fecha: 23/10/2017
Universidad de Granada

1. Variables necesarias y estrategia de escritura/lectura del buffer.

```
const int num_items = 40 ,   // número de items
              tam_vec   = 10;   // tamaño del buffer
unsigned  cont_prod[num_items] = {0}, // contadores de verificación: producidos
          cont_cons[num_items] = {0}; // contadores de verificación: consumidos
int primera_libre = 0; // primera celda libre del vector vec
int vec[tam_vec]; // buffer donde se introducirán/leeran los datos
Semaphore libres = tam_vec, ocupadas = 0, ex_m = 1; // Semáforos para controlar
// él número de casillas libres y ocupadas del vector, y otro para controlar
// que las lecturas/escrituras del mismo se hagan en exclusión mutua.
```

Para la lectura/escritura del buffer se ha utilizado la estrategia LIFO. Para ello se ha usado la variable “primera_libre” que controla cual es la primera casilla libre del buffer. Una vez el productor inserta una casilla, incrementa dicho valor en 1, mientras que el consumidor hace lo contrario cuando la consume.

Estos accesos al vector e incrementos de la variable deben hacerse en exclusión mutua, ya que es posible que mientras que una hebra accede al valor de cierta posición del vector, la otra modifique el valor de “primera_libre” y se acceda a casillas que no tienen datos o cuyos datos ya han sido consumidos.

2. Semáforos utilizados.

Semáforo “libres”: Para controlar el número de casillas libres del buffer. Al principio vale lo mismo que el tamaño del mismo, ya que todas las casillas están libres de inicio. Este semáforo se decrementará (sem_wait) cada vez que la **hebra productora escriba** en el vector, y se incrementará (sem_signal) cada vez que la **hebra consumidora lea** de dicho vector, ya que habrá una nueva casilla libre. Así, si el semáforo vale 0 (vector lleno) la hebra productora quedará bloqueada antes de escribir el nuevo valor en el vector.

Semáforo “ocupadas”: Para controlar el número de casillas ocupadas del buffer. Al principio vale 0, ya que hay 0 casillas ocupadas de inicio. Este semáforo se decrementará (sem_wait) cada vez que la **hebra consumidora lea** un valor del buffer, y se incrementará (sem_signal) cada vez que la **hebra productora escriba** en el vector.

Es necesario utilizar dos semáforos para que puedan producirse varias escrituras/lecturas del vector de manera independiente, es decir, sin que por cada escritura, la hebra productora tenga que esperar a que la hebra consumidora lea el valor antes de producir otro, y viceversa.