

Generalización de procesos en el problema del productor-consumidor

Asier Cabo Lodeiro y Hugo Gilsanz

1. Introducción

Siguiendo el esquema de los anteriores informes, el problema del productor-consumidor es un ejemplo clásico de sincronización en sistemas concurrentes (véase *Problema del productor-consumidor con memoria compartida*). Para evitar condiciones de carrera y garantizar un acceso seguro a la memoria compartida, se emplean mecanismos de sincronización como los semáforos (véase *Uso de semáforos para la resolución de carreras críticas*). En este informe, se presenta una solución basada en hilos (**threads**) (véase *Compatibilidad de semáforos e hilos en el problema del productor-consumidor*), de manera que el número de procesos pueda ser determinado por el usuario.

2. Problemática y posible solución

2.1 Descripción del problema

Ídem informe anterior

2.2 Incorporación de semáforos

Ídem informe anterior

2.3 Incorporación de hilos

Ídem informe anterior

2.4 Generalización de procesos mediante hilos

Se solicita al usuario una vez ejecutado el programa o bien por línea de comandos (número de productores como `argv[1]` y número de consumidores como `argv[2]`) la cantidad de productores y consumidores que corresponderá con el tamaño de un array de hilos que llamarán a las funciones homónimas.

3. Estructura y Análisis del código

Como se está empleando como base el código del apartado anterior (véase el informe *Compatibilidad de semáforos e hilos en el problema del productor-consumidor*), el código resultante sencillamente añadirá una verificación de argumentos y en caso de no haber una solicitud al usuario del número de procesos, que se generarán en sus respectivos bucles con `pthread_create` y las consiguientes esperas también generadas por bucles `pthread_join`.

3.1 Implementación en C

Ídem informe anterior

4. Instrucciones de Compilación y Ejecución

5.1 Compilación

Para compilar el código, se puede usar el compilador GCC de la misma manera que anteriormente, pero siendo ahora imprescindible incluir la biblioteca de hilos (`-pthread`). Ejemplo de compilación: `gcc op1.c -o op2 -lrt -pthread`

5.2 Ejecución

Existen dos opciones de ejecución por el usuario:

1. En una terminal se ejecuta el programa sin argumentos: `./op2`. Esto llevará a que el programa solicite mediante `scanf` que se introduzcan manualmente el número de productores y consumidores.
2. En una terminal se ejecuta el programa con argumentos de la forma: `./op2 n_prod n_cons`, siendo `n_prod` el número de procesos productor que generará el programa y `n_cons` el número de consumidores.

5. Conclusiones

La implementación con hilos y semáforos permite una ejecución segura y coordinada del problema productor-consumidor con múltiples actores. Gracias al uso de semáforos, se evitan problemas de condiciones de carrera y se logra una comunicación eficiente entre los hilos. Además, la introducción de pausas aleatorias simula un comportamiento más realista en entornos de producción y consumo concurrentes.