

Generalización del problema del productor-consumidor para n procesos

SALVIA SISA CORTÉS, EZEQUIEL SOTO SEOANE

Sistemas Operativos II

Grupo 05

{salvia.sisa, ezequiel.soto}@rai.usc.es

I. INTRODUCCIÓN

En este informe se tratará de implementar una solución al problema del productor-consumidor con semáforos, pero con un **número arbitrario de procesos**, seleccionado por el usuario. El aspecto más interesante de esta implementación resulta la utilización de una zona de **memoria compartida** por todos los procesos, y los casos en los que nos encontramos con más procesos de un tipo que del otro.

II. IMPLEMENTACIÓN

La implementación es igual que la comentada en el informe de semáforos, pero con la diferencia de que debemos crear **varios consumidores y productores**, por lo que utilizamos la llamada al sistema `fork()` para este propósito. Así creamos tantos productores o consumidores como se indique por línea de comandos (o 1 solo por defecto) mediante un **bucle** controlado por la variable `num_procesos`. El resto del código es idéntico al del apartado 2, por lo que no se comentará.

III. EJECUCIÓN DE LOS PROGRAMAS

Se compilan tanto el productor como el consumidor con `gcc` y se ejecutan cada uno en una terminal, pasándoles como argumento el **número de procesos** que se quiere crear. Por ejemplo, en una ejecución con **4 productores** y **2 consumidores** se obtuvieron estos resultados: En este caso, como los productores acaban sus 60 iteraciones antes de que los consumidores las

```
Numero iteración productor 1: 33
Productor 4: He producido el item V
Productor 4: He insertado el item V y ahora hay 7 items
Productor 3: He producido el item N
Numero iteración productor 4: 33
Productor 3: He insertado el item N y ahora hay 8 items
Numero iteración productor 3: 30
Productor 4: He producido el item Z
Productor 4: He insertado el item Z y ahora hay 7 items
Numero iteración productor 4: 34
Productor 2: He producido el item S
Productor 2: He insertado el item S y ahora hay 8 items
Numero iteración productor 2: 31
Consumidor 2 : He consumido el item V y ahora hay 6 itens
Numero iteración consumidor 2: 59
Consumidor 1: He sacado del buffer el elemento N
Consumidor 1 : He consumido el item N y ahora hay 6 itens
Numero iteración consumidor 1: 59
Consumidor 2: He sacado del buffer el elemento V
Consumidor 2 : He consumido el item V y ahora hay 6 itens
Numero iteración consumidor 2: 60
Consumidor 1: He sacado del buffer el elemento V
Consumidor 1 : He consumido el item V y ahora hay 6 itens
Numero iteración consumidor 1: 60
Buffer final: BEBJNJZS
String local:
```

Figura 1: Ejecución del problema productor-consumidor

alcancen, los consumidores **acaban de ejecutarse antes que los productores**, por lo que estos terminan por llenar el búfer y se quedan **esperando para siempre**.

IV. CONCLUSIÓN

Con este método sigue habiendo **carreras críticas**, al igual que en los anteriores, y, aunque pueda sonar rebuscado, realmente resulta muy **fácil generalizar** el problema del productor-consumidor a un número cualquiera de productores y de consumidores.