Problema del productor-consumidor para p procesos en cascada

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 el problema del productor-consumidor con semáforos pero generalizándolo para **P procesos**, de manera que el **primero producirá** caracteres que el segundo consumirá. Después, el segundo calculará el **siguiente caracter** consumido y se lo pasará al tercero, y así sucesivamente hasta **el P-ésimo** proceso, que actuará como **consumidor** del (P-1)-ésimo y no actuará como productor de ningún otro. Además se intentará que los P procesos **actúen concurrentemente** en la mayor medida posible.

II. Implementación

Realmente, esta implementación es muy similar a la del ejercicio con los n procesos pero con algún añadido. Primero, que cada par de procesos tiene sus **3 semáforos** (`llenas`, `vacias` y `mutex`), con un nombre que se otorga utilizando un `sprintf` dentro de un bucle for controlado por el número de procesos. Después se utiliza un `fork` para **crear los P procesos** y se hace que el padre espera a que terminen. Cada proceso llama a la función `funcionProceso()`, a la que se le pasa el **número de proceso** para poder identificarlos. Después, si se trata del **primer proceso**, que solo actúa como productor, ejecuta un código similar al del apartado de semáforos. Análogamente, si se trata del **último proceso**, que solo actúa como consumidor, se ejecuta un código también similar al comentado en el informe 2.

El cambio más significativo ocurre cuando se trata de los **procesos intermedios**, ya que debe crear zonas de memoria para **ambos procedimientos** (tanto consumir como producir). Además, en su bucle principal también tiene una **parte dedicada a consumir** el elemento del anterior y **producir** el del siguiente.

III. Ejecución del programa

Se compiló con **gcc** y se ejecutó para **4 procesos**, obteniendo los siguientes resultados:

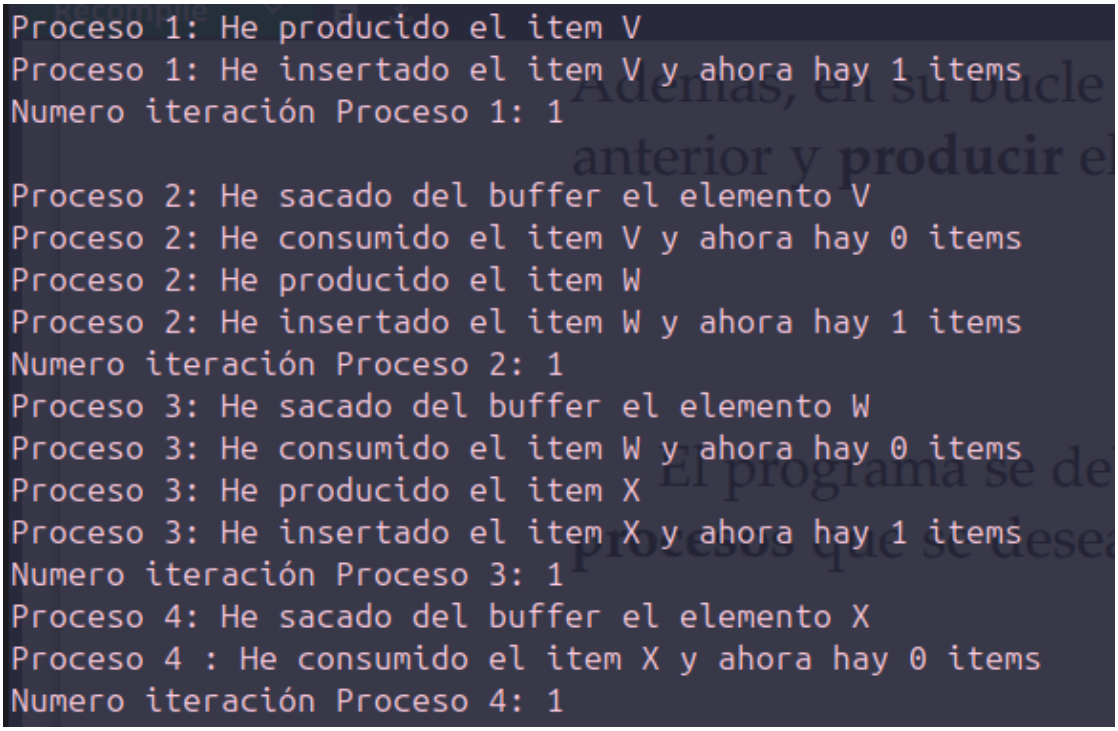

Figura 1: *Ejecución del problema productor-consumidor con 4 procesos*

IV. Conclusión

Como se puede ver, obtuvimos los resultados deseados, ya que el programa funciona correctamente, produciendo y consumiendo en cadena.