

----Miguel García López----

PRÁCTICA PRODUCTORES-CONSUMIDORES CON MPI:

° Para la realización del ejercicio de productores y consumidores con MPI he utilizado la plantilla de prodcons2 y he realizado las siguientes modificaciones:

- Vamos a trabajar con varios productores y varios consumidores, por tanto declaro dos variables globales (num_prod, num_cons) con valores 4 y 5 respectivamente.
- Utilizamos dos etiquetas para diferenciar mensajes de productores, consumidores y del búffer. Igualmente los más necesarios son los dos primeros.
- Las id's de productores, consumidores y búffer cambian. Según el pdf a seguir las id's entre 0 y el num_prod - 1 pertenecen a procesos productores, la id con valor num_prod pertenece al búffer y las id's con valor entre num_prod + 1 y num_prod + num_cons son de procesos consumidores.
- La función producir ahora depende del productor que la use (n), por lo que ahora los productos tendrán un valor $n * (\text{num_items} / \text{num_prod})$.
- Ahora en las funciones Ssend y Recv se les pasa la etiqueta correspondiente y el id del proceso adecuado, por ejemplo, si mandamos un mensaje de la función producir especificamos que la manda un productor con la etiqueta y que el receptor debe ser el búffer con el id de este.
- Diversos cambios relacionados con la ahora numerosidad de productores y consumidores, tales como en la función consumidor, en la que el bucle for itera hasta num_item / num_cons para distribuir la carga entre los diversos consumidores (lo mismo con productores).
- Dentro de la función del búffer, como ya se nos había advertido en el pdf, no nos es posible restringir a los emisores, pero podemos hacer uso de las etiquetas para diferenciar entre productores y consumidores. MPI_ANY_SOURCE hace que se puedan recibir mensajes de cualquier proceso, pero con eti_acceptable hacemos que se pueda controlar.
- En el switch de la función búffer cambiamos de estado.MPI_SOURCE a estado.MPI_TAG, ya que ahora los metadatos que estamos utilizando para diferenciar son las etiquetas.
- En el main comprobamos que id's se mandan a cada función (recordemos que tenemos acotados los valores asociados a cada rol).

PRÁCTICA FILÓSOFOS CON MPI:

° Para la realización del ejercicio de filósofos con MPI he utilizado la

plantilla de filósofos. La única modificación para la versión con interbloqueo ha sido la de añadir las funciones Ssend Y Recv necesarias para cada acción, además de añadir la variable "valor" que falta en la función de filósofos, el valor que se manda en Ssend. El problema del interbloqueo ocurre porque todos los filósofos cogen los tenedores en el mismo orden, entonces si un filósofo coge a la vez que otro su correspondiente tenedor izquierdo, al ir a coger el derecho se va a encontrar que el otro filósofo ya lo ha cogido. Esto se soluciona alterando el orden de toma de los tenedores de un filósofo, el primero por ejemplo, de forma que se compite por el primer tenedor.

PRÁCTICA FILÓSOFOS-CAMARERO CON MPI:

° Para la realización del ejercicio de filósofos-camarero con MPI he utilizado el programa anterior. Lo primero es cambiar el main, pues si el proceso es el número 10 debemos llamar a la función del camarero, por lo demás el main no cambia. En cuanto a la función del camarero simplemente aplicamos lo ya descrito en el pdf, si el número de filósofos sentados es menor a 4 se podrán sentar los que lo soliciten, en otro caso deben esperar. Para esto utilizamos etiquetas, diferenciando cada acción y recibiendo así solo las peticiones apropiadas. Hemos declarado una variable global para contabilizar cuantos filósofos sentados hay. Dentro de la función de filósofos, al generarse el id de los tenedores con el número total de procesos, hay veces que un tenedor obtiene la id del camarero, por ello debemos tenerlo en cuenta.