

Laboratorio de Sistemas Operativos

Ingenierías Técnicas en Informática de Gestión y de Sistemas

Hoja de prácticas 7

Curso 2005/2006

1. Escribir un programa para calcular el cociente $x(x-1)(x-2) \cdots 2 \cdot 1 / (x-n)$, para x y n dos números naturales dados como argumentos al programa. Para ello habrá de crearse un proceso hijo que calcule el numerador y se lo pase al padre a través de memoria compartida.
2. Escribir un programa que cree un proceso hijo. El hijo irá escribiendo en un buffer de 5 posiciones (que estará en memoria compartida) los números del 1 al 200; el padre los leerá y los mostrará por la salida de error estándar, sin perder ninguno.
3. Ejemplo de sistema productores-consumidores. Escribir tres programas, `almacen`, `productor` y `consumidor` con el siguiente comportamiento:
 - `almacen` recibe un natural n como argumento y crea en memoria compartida una estructura con un buffer con n posiciones, dos índices con las posiciones de acceso de consumidores y productores, y el tamaño del buffer. Además, se encarga de crear un semáforo que controlará cuándo pueden escribir los productores en el buffer, otro análogo para los consumidores y dos más para controlar los huecos libres y ocupados en el buffer.
 - `productor` crea tantos procesos productores como se le indiquen mediante un argumento. Estos procesos se limitan a escribir en el buffer creado por `almacen` siempre que haya un hueco libre.
 - `consumidor` crea tantos procesos consumidores como se le indiquen mediante un argumento. Estos procesos se limitan a consumir la información escrita por los productores.

Tanto los productores como los consumidores mostrarán un mensaje por pantalla informando de la acción que han realizado.

Los semáforos y la memoria compartida no se destruyen desde el código de estos programas; en su lugar, deben eliminarse utilizando `ipcrm`. Alternativamente, se podría escribir un programa `eliminar` que los destruya y que se ejecutaría al forzar la terminación de los anteriores.

4. Implementar la siguiente variación del ejercicio 3 de la práctica 6. Escribir un programa `abece` que muestre por `stderr` las letras del alfabeto separadas por un espacio en blanco y terminando con un salto de línea. Ejecutar varias copias de este programa concurrentemente y comprobar que la salida aparece entrelazada; para ello, utilizar un script `lanzar` que reciba un número n como argumento y lance en segundo plano n copias del programa `abece`.

Una vez hecho esto, modificar el programa utilizando semáforos para que la salida producida al ejecutar varias copias del proceso no aparezca entrelazada. Obsérvese que, a diferencia de lo que ocurría en la práctica 6, aquí los procesos son independientes y no tienen una relación padre-hijo. Será necesario, por tanto, utilizar una clave creada con `ftok` para referirse al semáforo. Además, el semáforo tendrá que inicializarse tan solo una vez; para ello, utilizar un variable entera en memoria compartida que tome el valor 1 tras la inicialización y que se utilice como guarda para decidir si hace falta inicializar el semáforo o no; para acceder a ella también habrá que utilizar una clave creada con `ftok`. Eliminar el semáforo y la memoria compartida con `ipcrm`, no desde el código del programa.