

Práctica 3

Comunicación y sincronización entre procesos

Abril 2019

Índice

1. Introducción	2
2. La práctica	2
2.1. El paradigma productor-consumidor	2
2.2. El algoritmo a implementar	3
3. Planificación e implementación	4
4. Entrega	5

1. Introducción

La práctica 3 se centra utilizar algunos de los métodos que nos ofrece el sistema operativo para comunicar procesos entre sí. Hay múltiples métodos de comunicación entre procesos entre los cuales podemos mencionar las tuberías, los ficheros, la red así como la las señales.

En esta práctica utilizaremos diversos métodos de comunicación entre procesos para desarrollar lo que se denomina un productor-consumidor. El problema del productor-consumidor es un ejemplo clásico de problema de sincronización de procesos. De forma general, en este paradigma de programación hay varios procesos, los productores, que "producen" información y se la "entregan" a los consumidores para que realicen alguna operación sobre ellos. El problema de sincronización en este paradigma de programación consiste en que el consumidor no intente coger datos si el buffer está vacío. De forma similar, el productor no puede introducir datos en el buffer hasta que el consumidor los haya cogido.

Esta práctica se basa en este paradigma, aunque se modifica el esquema de funcionamiento para facilitar la sincronización de los productores y los consumidores.

2. La práctica

En esta práctica sólo habrá un productor y un consumidor: el productor obtendrá la información de un fichero externo. El productor obtiene los datos a procesar y se los entregará al consumidor, el cual cogerá los datos filtrados y los procesa. La transferencia de datos entre el productor y el consumidor se realizará mediante una tubería, mientras que la sincronización entre ellos mediante señales.

Es importante mencionar que en esta práctica se utilizará la tubería de una forma diferente a la que se ha utilizado en la línea de comandos. Cuando se ejecuta el comando "ls | wc -l", por ejemplo, se construye una tubería entre el comando "ls" y el comando "wc -l". El comando "ls" va introduciendo los nombres de ficheros en la tubería y el comando "wc -l" va leyendo los datos de la tubería. No se utilizan mecanismos de sincronización entre ambos procesos.

En esta práctica la forma en que se utiliza la tubería es diferente. Por ello se utilizan mecanismos de sincronización entre ambos procesos. La razón de proceder así es implementar el paradigma de productor-consumidor mediante tuberías y señales. Se podrían haber propuesto otros mecanismos para realizar la transferencia de datos y sincronización entre el productor y consumidor, pero aquí se utilizaran tuberías y señales. Los detalles se describen a continuación.

2.1. El paradigma productor-consumidor

El productor lee los datos del fichero `data.csv`. Se trata de un fichero de texto que se ha obtenido de <http://publish.illinois.edu/dbwork/open-data>. El fichero original tiene 14,7 millones líneas de texto y aquí se ha recortado a 100000 líneas (99999 líneas de datos más la cabecera). El fichero contiene datos sobre viajes en taxi de la ciudad de Nueva York.

Cada línea del fichero contiene los datos de un viaje y los datos éstos están separados entre sí por comas (Comman Separated Values).

El productor extraerá de este fichero de texto las columnas 8 y 9 que nos indican cuantos pasajeros iba en el coche en cada viaje y cuanto ha durado cada viaje, en segundos. El productor generará los datos a transmitir en bloques de N datos (cada dato se corresponde a la columna 8 y 9 de una línea del fichero de entrada). El productor introducirá los datos de cada bloque en la tubería, el buffer de comunicación entre productor y consumidor. Cada vez que el productor introduzca un bloque de datos en el buffer avisará al consumidor.

El consumidor, por su parte, leerá los datos que el productor ha introducido en la tubería, los procesará y avisará al productor para que vuelva a introducir un bloque de datos en la tubería. El productor volverá a introducir otro bloque de datos en la tubería y avisa al consumidor. Así de forma sucesiva hasta que no haya más datos a enviar.

Una vez el consumidor haya leído todos los datos que el productor quería introducir en la tubería, el consumidor imprimirá por pantalla la media de pasajeros por viaje así como la media de duración de viaje.

Para poder implementar la propuesta de esta práctica se requiere sincronizar productor y consumidor. Para ello se hará uso de las señales. El algoritmo se detalla en la siguiente sección.

2.2. El algoritmo a implementar

En esta práctica se implementa el paradigma del productor-consumidor los cuales se comunican mediante una tubería. Para que esta comunicación sea posible, es necesario que entre ellos haya una relación de padre-hijo. Para esta práctica, se propone que el productor sea el padre y que el consumidor sea el hijo.

A continuación se detalla el algoritmo a implementar:

1. El productor lee N líneas del fichero de texto y extrae las columnas 8 y 9 de este fichero, ver sección 3. El productor envía, a través de la tubería, los datos al consumidor. Para enviar los datos, el productor introduce primero en la tubería el tamaño del bloque. A continuación introduce en la tubería los datos. Una vez ha introducido los datos, envía un SIGUSR2 al consumidor.
2. Una vez recibida la señal, el consumidor lee el tamaño del bloque de la tubería y lee los datos (el consumidor desconoce el valor de N). Una vez haya leído y procesado los datos, avisa al productor mediante la señal SIGUSR1 para que vuelva a introducir datos en la tubería.
3. Una vez el productor ya no tenga más datos a enviar, introducirá un tamaño de bloque de 0, le enviará el señal correspondiente al consumidor y finalizará. El consumidor leerá el valor de 0, sabrá que ya no hay más datos a leer de la tubería, imprimirá por pantalla la media de pasajeros y la media de duración de los viajes, y finalizará.

3. Planificación e implementación

Junto con esta práctica se entrega una plantilla a partir de la cual se puede comenzar a trabajar. En esta plantilla se incluye el código para extraer la columna 8 y 9 de un fichero. Se recomienda utilizar esta plantilla puesto que de esta forma únicamente habrá que concentrarse en los puntos primordiales y básicos de esta práctica: la sincronización de procesos mediante señales y la comunicación entre procesos mediante la tubería.

En el código plantilla que se entrega se lee un bloque de N líneas del fichero de datos y se calcula la media de pasajeros así como la media del tiempo de duración del viaje.

Ejecutad el código y comprobad el resultado que os da cuando se lee todo el fichero. En este caso el fichero es pequeño y por ello es factible la implementación tal como está realizada.

En caso que se tenga que trabajar con ficheros muy grandes se puede hacer más difícil trabajar con ellos. Es por ello que típicamente se utilizan paradigmas de productor-consumidor en que el primero lee los datos y el segundo los procesa. De forma similar a como se ha realizando en la práctica 2, los esquemas de productor-consumidor pueden estar encadenados uno detrás de otros para procesar los datos.

Se indican a continuación una serie de puntos a tener en cuenta a la hora de implementar la práctica, así como unos primeros pasos a seguir para implementar la práctica

1. En esta práctica el productor y el consumidor serán, respectivamente, padre e hijo. El programa será el mismo, la diferencia está en el hecho que padre e hijo ejecutaran funciones diferentes. El padre ejecutará, por ejemplo, una función denominada **productor** mientras que el hijo ejecutará una función denominada **consumidor**.
2. Se puede comenzar con una primera implementación de prueba para comprobar el funcionamiento de la tubería. Todavía no se implementa el esquema de productor-consumidor. Para esta implementación de prueba, utilizad la plantilla que se os proporciona.

En particular, cread la tubería así como el productor (padre) y consumidor (hijo). El productor lee N líneas del fichero de datos y únicamente introduce los valores de la columna 8 (número de pasajeros) en la tubería. Mientras el productor va introduciendo los datos en la tubería, el consumidor irá leyendo los valores de la tubería. Se recomienda que la transferencia de datos se realice en formato binario¹.

Cuando el productor haya introducido todos los datos en la tubería, introduce un valor 0 (número de pasajeros 0) en la tubería. Cuando el consumidor lea el valor 0, sabrá que ya no hay más datos a leer. El consumidor imprimirá por pantalla el valor medio de número de pasajeros. Tener en cuenta que el consumidor no sabe cuantos elementos ha leído el productor. Por lo tanto, el consumidor tendrá que tener un contador para saber cuantos elementos ha leído de la tubería.

¹Para realizar la transferencia hay que utilizar las funciones `read` y `write`. No utilizar funciones como `fprintf`

3. Una vez se haya acabado con la anterior parte, se puede proceder a realizar la implementación del algoritmo descrito en la sección 2.2. En esta sección el productor leerá el fichero de datos en bloques de N líneas y se utilizará el mecanismo de sincronización entre productor y consumidor mediante señales. Uno de los **requisitos** en la implementación de la práctica es que el productor llene, en cada iteración, *la tubería con tantos datos como sea posible*. Se recomienda que la *transferencia de datos se realice en formato binario*.

4. Entrega

Cada grupo de prácticas tiene que entregar un fichero que contenga todos los ficheros necesarios para compilar el código. No es necesario incluir el fichero de datos. El nombre del fichero debe indicar los componentes de la pareja, por ejemplo, `GarridoLluis_BazanJordi.zip`.