

Práctica 3 - Sincronización de procesos con hilos

Informe Entregable: Consumidores Intermedios

LUCÍA PÉREZ GONZÁLEZ, ADRIÁN QUIROGA LINARES

Sistemas Operativos II, Grupo 04

lucia.perez.gonzalez1@rai.usc.es, adrian.quiroga@rai.usc.es

I. INTRODUCCIÓN

Esta práctica extiende el clásico problema del *productor-consumidor* con una jerarquía de hilos en tres niveles:

- **Productores:** Leen caracteres de ficheros de entrada y los insertan en un buffer compartido.
- **Consumidores Intermedios:** Retiran datos del primer buffer, escriben en sus ficheros, y si el carácter es numérico, lo reinsertan en un segundo buffer.
- **Consumidores Finales:** Consumen únicamente caracteres numéricos del segundo buffer y calculan su suma.

El objetivo es ilustrar la comunicación entre hilos múltiples a través de buffers sincronizados, aplicando técnicas de exclusión mutua y condiciones para gestionar el acceso concurrente.

II. EJECUCIÓN

Compilación

```
gcc -o opt3_prod_cons opt3_prod_cons.c -lpthread
```

Ejecución

```
./opt3_prod_cons <num_productores> <num_consumidores_intermedios>  
                  <num_consumidores_finales>
```

Durante la ejecución, se solicitarán:

- Nombre del fichero de entrada y tiempo de espera para cada productor.
- Tiempo de espera para consumidores intermedios y finales.

III. FUNCIONAMIENTO DEL CÓDIGO

A. Productores

Cada productor abre un fichero de texto del cual lee carácter a carácter. Si el carácter es alfanumérico, se inserta en el buffer principal. El marcador '*' indica el fin de los datos y también se inserta para señalar el fin de la producción.

B. Consumidores Intermedios

Extraen caracteres del buffer principal. Si el carácter es numérico, lo insertan en un segundo buffer (numérico). También escriben en un fichero de salida. Cuando detectan que todos los productores han terminado, concluyen su trabajo.

C. Consumidores Finales

Estos hilos leen exclusivamente caracteres del buffer numérico. Su función es calcular la suma total de los dígitos consumidos. Al encontrar un carácter '*', terminan su ejecución.

D. Sincronización

Se emplean dos buffers protegidos por mutexes y variables de condición:

- **Buffer principal:** Compartido entre productores y consumidores intermedios.
- **Buffer numérico:** Usado entre consumidores intermedios y finales.

Se sincroniza el inicio de todos los hilos mediante una barrera para garantizar una ejecución ordenada.

IV. CONCLUSIONES

Esta implementación ejemplifica una solución eficaz al problema jerárquico de productor-consumidor utilizando múltiples niveles de comunicación entre hilos, tal y como podemos observar en las Figuras 1 y 2. Los aspectos clave incluyen:

- Uso correcto de `pthread_mutex` y `pthread_cond_t` para evitar condiciones de carrera.
- Doble buffer sincronizado que permite separar las fases de procesamiento.
- Cierre controlado y seguro de hilos mediante el uso del carácter '*' como señal de finalización.
- Modularidad y escalabilidad del sistema, adaptable a distintos números de hilos.

La separación de responsabilidades y el diseño estructurado permiten un sistema robusto, fácil de entender y de mantener. Esta práctica refuerza el conocimiento sobre concurrencia, sincronización y diseño de sistemas multihilo en C.

```

> ./opt3_prod_cons 2 1 1
Productor 1) Nombre del fichero de entrada: file1.txt
    Tiempo a dormir (T): 1
Productor 2) Nombre del fichero de entrada: file2.txt
    Tiempo a dormir (T): 2
Consumidor Intermedio 1) Tiempo a dormir (T): 1
Consumidor Final 1) Tiempo a dormir (T): 1

    INICIO DE LA EJECUCIÓN

Productor 1: Inserta 'a' en la posición 0
    Consumidor Intermedio 1: Retira 'a' de la posición 0
Productor 1: Inserta 'b' en la posición 1
    Consumidor Intermedio 1: Retira 'b' de la posición 1
Productor 1: Inserta 'c' en la posición 2
    Consumidor Intermedio 1: Retira 'c' de la posición 2
Productor 1: Inserta 'd' en la posición 3
    Consumidor Intermedio 1: Retira 'd' de la posición 3
Productor 1: Inserta 'e' en la posición 4
Productor 1: Inserta '1' en la posición 5
Productor 1: Inserta '2' en la posición 6
Productor 1: Inserta 'F' en la posición 7
    Consumidor Intermedio 1: Retira 'e' de la posición 4
Productor 1: Inserta 'G' en la posición 0
    Consumidor Intermedio 1: Retira '1' de la posición 5
Consumidor Intermedio 1: Inserta '1' en buffer numérico en la posición 0
    Consumidor Intermedio 1: Retira '2' de la posición 6
Consumidor Intermedio 1: Inserta '2' en buffer numérico en la posición 1
    Consumidor Final 1: Retira '1' de la posición 0 del buffer numérico
Productor 1: Inserta '*' en la posición 1
    Consumidor Final 1: Retira '2' de la posición 1 del buffer numérico
    Consumidor Intermedio 1: Retira 'F' de la posición 7
    Consumidor Intermedio 1: Retira 'G' de la posición 0
    Consumidor Intermedio 1: Retira '*' de la posición 1
Productor 2: Inserta '0' en la posición 2
    Consumidor Intermedio 1: Retira '0' de la posición 2
Consumidor Intermedio 1: Inserta '0' en buffer numérico en la posición 2
    Consumidor Final 1: Retira '0' de la posición 2 del buffer numérico
Productor 2: Inserta '1' en la posición 3
    Consumidor Intermedio 1: Retira '1' de la posición 3
Consumidor Intermedio 1: Inserta '1' en buffer numérico en la posición 3

```

Figura 1: Ejecución del programa ./opt3_prod_cons 2 1 1

```

    Consumidor Final 1: Retira '1' de la posición 3 del buffer numérico
Productor 2: Inserta 'm' en la posición 4
    Consumidor Intermedio 1: Retira 'm' de la posición 4
Productor 2: Inserta 'n' en la posición 5
    Consumidor Intermedio 1: Retira 'n' de la posición 5
Productor 2: Inserta '8' en la posición 6
    Consumidor Intermedio 1: Retira '8' de la posición 6
Consumidor Intermedio 1: Inserta '8' en buffer numérico en la posición 4
    Consumidor Final 1: Retira '8' de la posición 4 del buffer numérico
Productor 2: Inserta '7' en la posición 7
    Consumidor Intermedio 1: Retira '7' de la posición 7
Consumidor Intermedio 1: Inserta '7' en buffer numérico en la posición 0
    Consumidor Final 1: Retira '7' de la posición 0 del buffer numérico
Productor 2: Inserta 'W' en la posición 0
    Consumidor Intermedio 1: Retira 'W' de la posición 0
Productor 2: Inserta 'e' en la posición 1
    Consumidor Intermedio 1: Retira 'e' de la posición 1
Productor 2: Inserta 's' en la posición 2
    Consumidor Intermedio 1: Retira 's' de la posición 2
Productor 2: Inserta 'd' en la posición 3
    Consumidor Intermedio 1: Retira 'd' de la posición 3
Productor 2: Inserta '*' en la posición 4
    Consumidor Intermedio 1: Retira '*' de la posición 4
    Consumidor Final 1: Retira '*' de la posición 1 del buffer numérico
Consumidor Final 1: Suma total = 19

```

Figura 2: Ejecución del programa ./opt3_prod_cons 2 1 1