



Área de Arquitectura y Tecnología de Computadores  
Universidad Carlos III de Madrid

# SISTEMAS OPERATIVOS

## Práctica 3 - Programación multi-hilo

Grado de Ingeniería en Informática  
Grado en Matemática Aplicada y Computación  
Doble Grado en Ingeniería Informática y Administración de  
Empresas

Curso 2023-2024

- 1 Introducción
- 2 Descripción de la Práctica
- 3 Material
- 4 Entrega

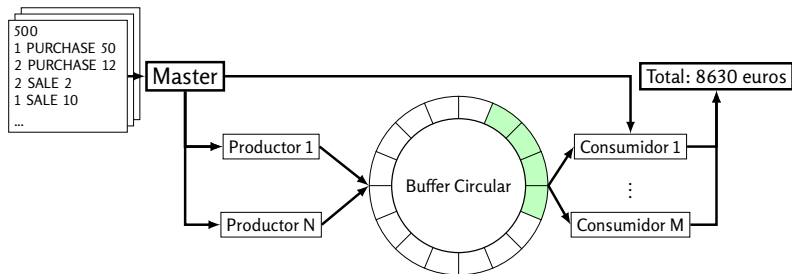
- 1** Introducción
- 2 Descripción de la Práctica
- 3 Material
- 4 Entrega

# Introducción

- Desarrollo de un sistema de cálculo de beneficios y *stock* de una tienda.
  - El usuario proporciona un fichero que indica la compra/venta de unidades de un determinado producto (**fichero de entrada**).
  - El sistema debe proporcionar los beneficios y el *stock* de cada producto una vez se han realizado todas las operaciones (**salida**).
- Para el cálculo de costes:
  1. Cargar los datos del fichero en memoria.
  2. Iniciar un sistema N—productores y M—consumidores
    - Los productores insertan los datos de memoria al buffer circular compartido
    - Los consumidores extraen los datos y calculan el beneficio y *stock* de cada operación (acumulándolos para obtener el beneficio y *stock* parcial).
  3. El beneficio y *stock* total es calculado por el proceso principal.

- 1 Introducción
- 2 Descripción de la Práctica**
- 3 Material
- 4 Entrega

# Proceso de desarrollo



**Figura:** Ejemplo de funcionamiento con  $N$  productores,  $M$  consumidores y un buffer

# Rol: Proceso Principal

- Es el encargado de:
  1. Inicializar las estructuras pertinentes (mutex, variables condición, arrays, ...).
  2. Obtener las operaciones indicadas en el fichero de entrada y almacenarlas en memoria (para posterior procesamiento).
  3. Lanzar los hilos que componen el sistema Nprod—Mcons.
  4. Esperar la finalización de los hilos.
  5. Mostrar el resultado por pantalla.
- NOTA: Se recomienda definir una estructura para cada operación y almacenarlas en un array de estructuras (SoA).

# Rol: Productor

- Puede haber de 1 hasta N (valor indicado por parámetro).
- Su función es:
  1. Obtener las operaciones que le correspondan\* de la estructura en memoria creada por el proceso principal.
  2. Insertarlas en el buffer circular compartido.
  3. Una vez que ha terminado de insertar las operaciones, finaliza su ejecución con *pthread\_exit()*.
- \*Se recomienda que el proceso principal realice un reparto de las operaciones a procesar.

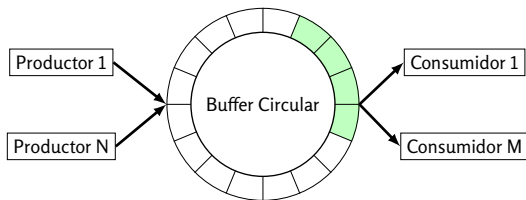


## Rol: Consumidor

- Puede haber de 1 hasta M (valor indicado por parámetro).
- Su función es:
  1. Obtener las operaciones que hay insertadas en el buffer circular compartido.
  2. Para cada una:
    - Calcular el beneficio y *stock* de esta operación.
    - Acumular estos valores para calcular el beneficio y *stock* parcial del hilo.
  3. Una vez que ha terminado de procesar las operaciones, finaliza su ejecución con `pthread_exit()` **retornando el beneficio y *stock* parcial calculado.**

# Buffer circular

- Es la estructura que almacenará las operaciones que se quieren procesar.
- Será utilizada al mismo tiempo por productores y consumidores.
- Los accesos deben ser concurrentes.
- Funciones básicas definidas en *queue.c* y *queue.h*.
- Completar estructura de operación y cola.



# Concurrencia

- El control de la concurrencia debe realizarse utilizando **únicamente mutex y variables condición**.
- Puede realizarse en dos lugares:
  - *queue.c*: La concurrencia se gestiona directamente en el buffer.
  - *store\_manager.c*: La concurrencia es gestionada entre los threads que acceden a la cola.
- Utilizar el método que resulte más sencillo para el grupo.
- **Requisito:** El resultado del cálculo con N-productores debe ser igual que el obtenido con 1-productor.

# Entrada

- Argumentos de entrada:

```
./store_manager <file><prods><cons><bsize>
```

- *File*: Fichero de entrada. Incluye número de operaciones a procesar y la lista de operaciones con el siguiente formato:

```
1 500 #Num max operaciones a procesar
2 1 PURCHASE 50 #[product_id] [operation] [units]
3 2 PURCHASE 12
4 2 SALE 2
5 1 SALE 10
6 ... #Puede haber mas operaciones que num max, pero no menos
```

- *Prods*: Número de productores que se deben ejecutar.
- *Cons*: Número de consumidores que se deben ejecutar.
- *Bsize*: Tamaño del buffer circular (número de elementos máximo que puede almacenar al mismo tiempo).

# Salida

```
1 $> ./store_manager input_file 5 2 10
2 Total: 234234 euros
3 Stock:
4     Product 1: 10 units
5     Product 2: 2 units
6     Product 3: 50 units
7     Product 4: 22 units
8     Product 5: 33 units
9 $>
```

- 1 Introducción
- 2 Descripción de la Práctica
- 3 Material**
- 4 Entrega

## Código Fuente de Apoyo

- Para el desarrollo de la práctica se proporcionará código adicional que puede descargar de Aula Global.
- Los ficheros proporcionados son:

```
1      p3_multihilo_2024/  
2          Makefile  
3          store_manager.c  
4          queue.c  
5          queue.h  
6          autores.txt  
7          file.txt  
8          probador_ssoo_p3.sh
```

- Para compilar la práctica simplemente ejecutar el comando `make`.
- Se deben seguir las normas incluidas en el enunciado.

# Corrector

- Se proporciona un corrector automático que ofrece una tentativa de nota sobre las pruebas funcionales.
- El corrector ejecuta pruebas básicas, luego el código pasará más ejemplos para su evaluación.
- Previo a la ejecución:
  - Comprimir en ZIP con el nombre correcto los ficheros solicitados.
- Ejecución:

```
./corrector_ssoo_p3.sh <fichero_zip>
```

- Ejemplo:

```
$ ./corrector_ssoo_p3.sh p3_ssoo_100254896_100047014.zip
```



- 1 Introducción
- 2 Descripción de la Práctica
- 3 Material
- 4 Entrega**

# Plazo de entrega y grupos

- Grupos de 3 personas máximo.
- Entrega de:
  - Código Fuente en un archivo comprimido
  - Memoria de práctica en PDF a través de entregador TURNITIN
  - Solamente podrá entregar un integrante del grupo
- Fecha de entrega:  
**10 de mayo de 2024 (hasta las 23:55h)**



Área de Arquitectura y Tecnología de Computadores  
Universidad Carlos III de Madrid

# SISTEMAS OPERATIVOS

## Práctica 3 - Programación multi-hilo

Grado de Ingeniería en Informática  
Grado en Matemática Aplicada y Computación  
Doble Grado en Ingeniería Informática y Administración de  
Empresas

Curso 2023-2024