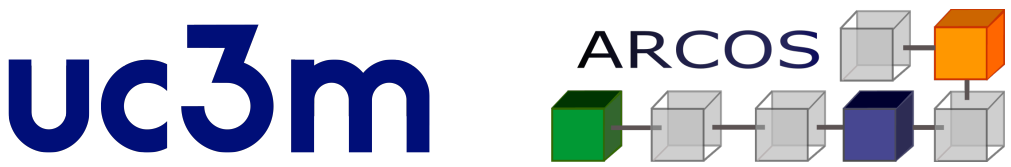


Área de Arquitectura y Tecnología de Computadores

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 2022-2023

Índice

Índice

Introducción

- Desarrollo de un programa que actue como un banco realizando operaciones sobre cuentas desde cajeros automáticos.
  - El usuario proporciona un fichero con una lista de operaciones sobre cuentas: CREAR, INGRESAR, RETIRAR, SALDO, TRASPASAR (**fichero de entrada**)
  - El sistema debe ejecutar las operaciones sobre las cuentas bancarias indicadas y mostrar el saldo total del banco (**salida**).
- Para el cálculo del saldo:
  - Cargar los datos del fichero en un array en memoria.
  - Iniciar un sistema N−productores y M−consumidores
    - Los productores (cajeros) insertan los datos de memoria al buffer circular compartido
    - Los consumidores (trabajadores) extraen los datos del array y realizan la operación bancaria indicada, actualizando el saldo total del banco.

Índice

Proceso de desarrollo

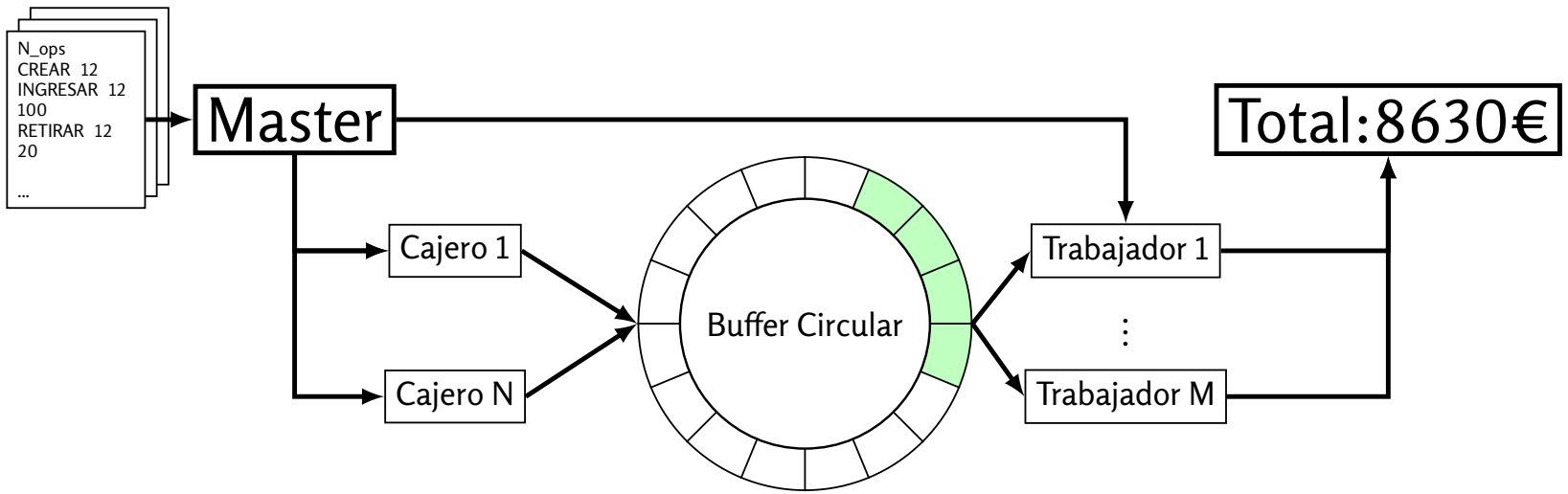


Figura: *Ejemplo de funcionamiento con N cajeros, M trabajadores y un buffer*

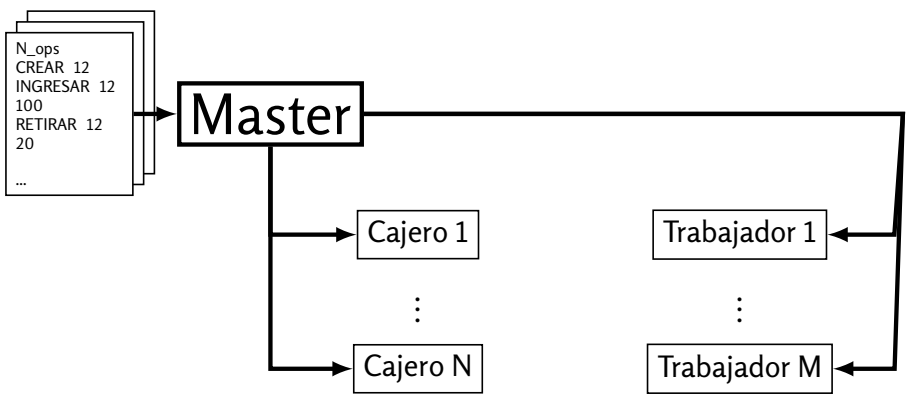
Rol: Proceso Principal

- 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 el array de 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 (*global_balance*).
- NOTA: Se recomienda definir una estructura para cada operación y almacenarlas en un array de estructuras (AoS).

Rol: Productor

1. Puede haber de 1 hasta N productores (valor indicado por parámetro).

2. Su función es:
   1) Obtener los datos de la operación, aumentar la variable de operación de cliente (*client_numop*) de forma segura.
   2) Crear un elemento con los datos de la operación para insertar en la cola.
   3) Insertar el elemento en la cola compartida.
   4) Una vez que ha terminado de insertar las operaciones, finaliza su ejecución con *pthread_exit()*.

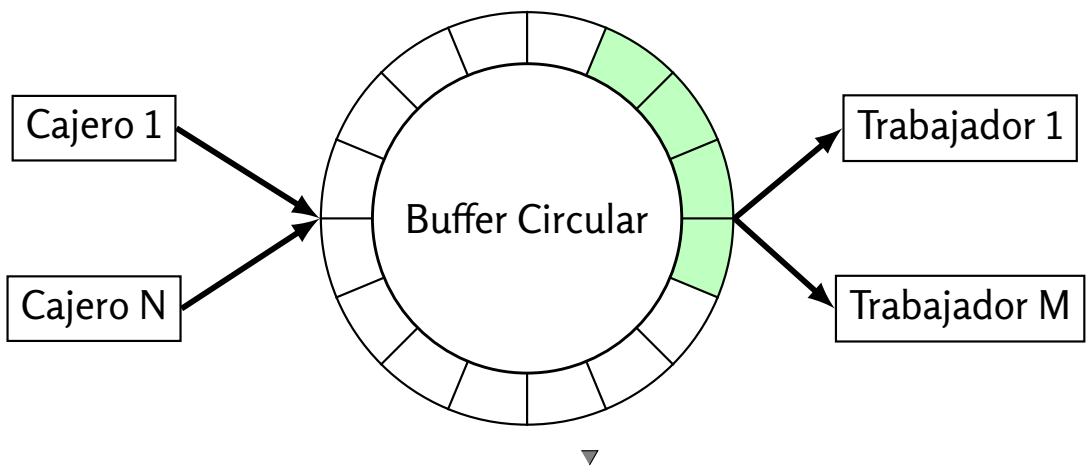


Rol: Consumidor

- Puede haber de 1 hasta M consumidores (valor indicado por parámetro).
- Su función es:
  1. Incrementar el número de operación del banco (*bank_numop*) y extraer de la cola la operación con ese número.
  2. Para cada elemento extraído del array de operaciones se realiza la operación bancaria indicada sobre las cuentas asociadas y además se actualiza el saldo global.
  3. Imprimir por pantalla el tipo de operación sus parámetros y el saldo resultante de hacerla.
  4. Una vez que ha terminado de procesar las operaciones, el consumidor finaliza su ejecución con *pthread_exit()*.

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.
  - *bank.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 Nprod- Mcons debe ser igual que el obtenido con 1prod-1cons.

Entrada

- Argumentos de entrada:

```
./bank <file><prods><cons><max_cuentas><bsize>
```

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

```
50 #Num max operaciones a procesar
CREAR 12
INGRESAR 12 100
RETIRAR 12 20
CREAR 25
TRASPASAR 12 25 30
SALDO 25
... #El número de operaciones tiene que ser el indicado en la primera línea
```

- *Prods*: Número de productores (cajeros) que se deben ejecutar.
- *Cons*: Número de consumidores (trabajadores) que se deben ejecutar.
- *max_cuentas*: Número máximo de cuentas bancarias que puede haber en el banco.
- *Bsize*: Tamaño del buffer circular (número de elementos máximo que puede almacenar al mismo tiempo).

Salida

```
$> ./bank input_file 5 3 50 20
1 CREAR 12 SALDO=0 TOTAL=0
2 INGRESAR 12 100 SALDO=100 TOTAL=100
3 RETIRAR 12 20 SALDO=80 TOTAL=80
4 CREAR 25 SALDO=0 TOTAL=80
5 TRASPASAR 12 25 30 SALDO=30 TOTAL=80
6 SALDO 25 SALDO=30 TOTAL=80
...
$>
```

Índice

Código Fuente de Apoyo

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

```
ssoo_p3_multihilo_2023
    Makefile
    bank.c
    queue.c
    queue.h
    autores.txt
    file.txt
    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:

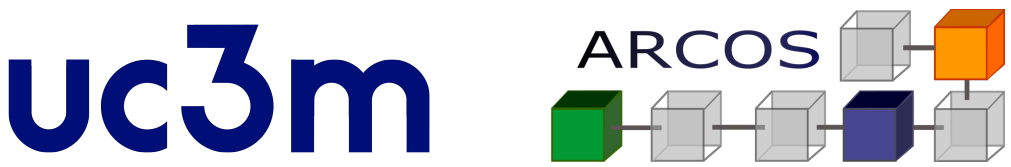
```
$ ./corrector_ssoo_p3.sh ssoo_p3_100254896.zip
```

Índice

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:

12 de mayo de 2023 (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 2022-2023