

Curso 2019 - 2020

Práctica Final – Procesamiento de números

Objetivos

- Practicar la creación y gestión de hilos en UNIX.
- Sincronizar el funcionamiento de hilos en UNIX.
- Practicar la gestión y control de acceso a memoria compartida mediante semáforos.

Descripción general

El objetivo de la práctica es diseñar un sistema concurrente de procesamiento de números desarrollado en el lenguaje de programación C. Para ello, se dispondrá de un conjunto de números enteros aleatorios comprendidos entre 0 y 3.000.000, desordenados y almacenados en el fichero **numeros.dat**. La cantidad de números que contiene el archivo es desconocida y se pueden encontrar líneas que no son números y que deben ser filtradas. El archivo se puede obtener de jair.lab.inf.uva.es en la ubicación `~javramo/pfinal/numeros.dat`.

El sistema deberá constar de una serie de hilos que deben lanzarse al comienzo del proceso de forma concurrente:

- **Productor (Lector/Limpiador):** Leerá del fichero de datos **numeros.dat** un número desconocido de líneas que contienen cadenas de caracteres, mayormente números enteros. Los validará, transformará a números enteros y los irá almacenando en un buffer circular (**buffer1**) compartido con los *Consumidores*.
- **Consumidor/es:** Los consumidores se encargarán de tratar un rango específico de los valores que el productor vaya escribiendo en el **buffer1**. De tal forma, que, si se tienen, por ejemplo, 10 consumidores, el consumidor 1 tratará el rango de [0,299.999] el segundo el rango [300.000,599.999] y así sucesivamente hasta el último que se encargará de tratar los datos restantes en el caso de que la división no sea exacta.

Todos los consumidores competirán por leer del **buffer1** de forma concurrente y sólo sacarán el dato del buffer cuando el dato leído del buffer esté dentro del rango que debe tratar.

Con los datos recogidos del buffer circular (**buffer1**), cada consumidor realiza los cálculos necesarios para obtener:

- El número de datos procesados por el consumidor.
- El valor mínimo de los números procesados.
- El valor máximo de los números procesados.
- La suma total de los números procesados.
- La media de los números procesados.
- La mediana de los números procesados.
- El primer cuartil de los números procesados.

El consumidor terminará cuando el productor indique que se han leído todos los datos y una vez realizados los cálculos cada uno de los consumidores almacenará el resultado de estos cálculos en un **vector** de tamaño igual al número de consumidores, de tal forma que cada consumidor tiene una posición del **vector** asignada para almacenar sus resultados.

- **Consumidor Final (Lector):** recogerá la información almacenada en el **vector** y la hará persistente en un fichero de texto plano una vez que algún consumidor termine su proceso. El formato de salida para cada consumidor terminador será el siguiente:

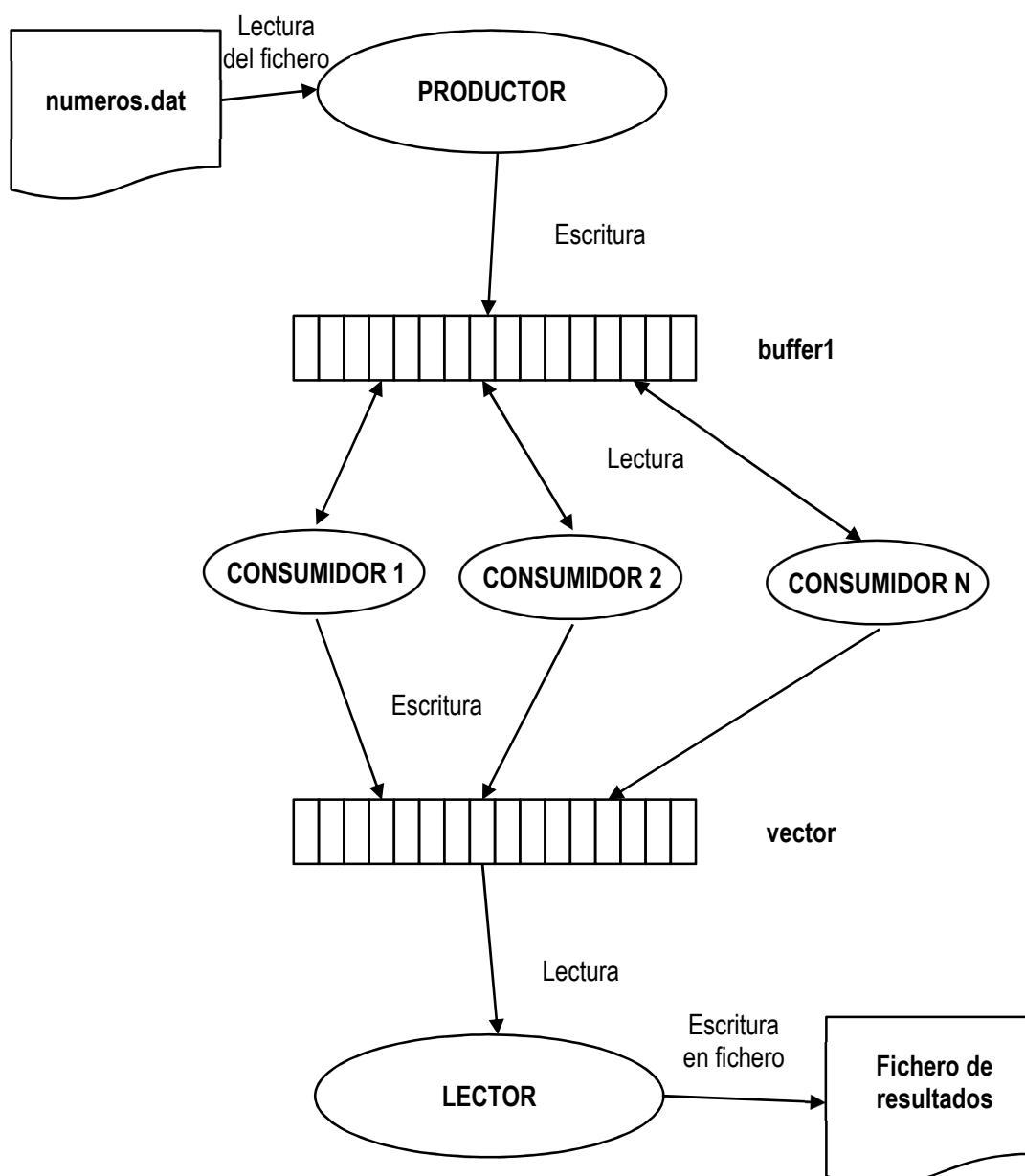
```

===== RANGO del Hilo 10 [2000000]-[2199999]
El número de datos del hilo 10 es: 69588.000000
El mínimo del hilo 10 es: 2000000.000000
El máximo del hilo 10 es: 2199995.000000
La suma total del hilo 10 es: 146124636160.000000
La media del hilo 10 es: 2099854.000000
La mediana del hilo 10 es: 2099923.500000
El primer Cuartil del hilo 10 es: 2049562.000000

===== RANGO del Hilo 1 [200000]-[399999]
El número de datos del hilo 1 es: 69737.000000
El mínimo del hilo 1 es: 200001.000000
El máximo del hilo 1 es: 399997.000000
La suma total del hilo 1 es: 20926322688.000000
La media del hilo 1 es: 300074.906250
La mediana del hilo 1 es: 300414.000000
El primer Cuartil del hilo 1 es: 249889.000000

```

En la siguiente imagen se muestra de manera esquemática los diferentes hilos y recursos de los que consta el sistema.



Desarrollo de la práctica:

Parte 0 (OBLIGATORIA): 1 único consumidor – Nota máxima alcanzable 4.0 puntos sobre 10.

La aplicación a realizar tendrá la siguiente sintaxis a la hora de ser invocada para su ejecución:

```
./<program> <inputFile> <outputFile> <tamBuffer> <numConsumidores>
```

IMPORTANTE - El programa realizará las siguientes tareas:

- El **programa principal** se encargará de validar los parámetros pasados en la línea de comandos, reservar la memoria necesaria y crear hilos, asignando a éstos la tarea/función que les corresponda, y esperará a que estos hilos creados finalicen su ejecución de manera correcta. Se valorará el uso de estructuras y memoria dinámica.
- El hilo **Productor** deberá:
 - Leer las cadenas de caracteres almacenadas en cada línea del fichero de entrada <inputFile>, pasado como primer argumento al programa, validarlas y convertirlas a números enteros.
 - Almacenar los números en el **buffer1**, cuyo tamaño será el del tercer argumento pasado al programa <tamBuffer>.
- Existirá un único hilo **Consumidor** que:
 - Realizará los cálculos siguientes, número de datos procesados, el valor mínimo, el valor máximo, la suma total y la media.
- Todos los hilos deben lanzarse a la vez y han de ejecutarse de manera concurrente, coordinando, sus operaciones mediante semáforos.
- El tamaño del buffer circular **buffer1** será <tamBuffer> (tercer argumento al programa).
- Los resultados obtenidos deben mostrarse por pantalla en el formato explicado anteriormente por el programa padre una vez terminen todos los hilos.

Parte 1 (OBLIGATORIA): Varios consumidores y lector final - Nota máxima alcanzable 8.0 puntos sobre 10.

La aplicación será una extensión de la parte 0 donde

- El número de consumidores se obtendrá del cuarto argumento <numConsumidores>, que además almacenará en el **vector** los mismos resultados calculados en la parte 0.
- Existirá un único hilo **Lector** que recogerá los datos del **vector** y los irá almacenando en el fichero según terminen los consumidores y cuyo nombre aparece como segundo argumento al programa <outputFile>.
- Todos los hilos deben lanzarse a la vez y han de ejecutarse de manera concurrente, coordinando, si fuera necesario, sus operaciones mediante semáforos.
- El tamaño del **vector** será igual al número de consumidores que se proporcionan como argumento del programa. Cada consumidor almacenará sus resultados en una posición fija, correspondiente al identificador del hilo.

En las pruebas es importante probar diferentes tamaños del **buffer1**, para comprobar que la coordinación con semáforos es correcta. Se pedirá que se prueben diferentes tamaños del buffer durante la defensa.

Parte 2: Cálculo de la mediana y los cuartiles - Nota máxima alcanzable 10.0 puntos sobre 10.

Hay que tener en cuenta que para calcular la mediana y el primer cuartil se necesita crear una lista ordenada de los datos que trata el consumidor. Recomendamos hacer una lista enlazada para realizar estos cálculos.

Se añadirá el cálculo de la mediana y el primer cuartil para lo que será necesario ordenar los datos recolectados por cada consumidor.

Seguimiento y entrega parcial

La práctica se debe realizar tanto en las sesiones de prácticas como fuera de estas. Cada grupo de prácticas debe dejar constancia al terminar cada sesión de prácticas del trabajo que lleva realizado y de cómo va a afrontar los próximos pasos.

La parte 0 de la práctica debe entregarse en el Moodle de la asignatura la semana del **2/12/2019** justo antes de que comience la sesión de prácticas de cada grupo. Se entregará lo que se haya hecho hasta ese momento independientemente de si se han conseguido todos los requisitos de esa fase.

La semana del **9/12/2019 2019** justo antes de que comience la sesión de prácticas de cada grupo, deberá realizarse una nueva entrega para realizar el seguimiento del grupo.

Las partes 1 (obligatoria) y 2 (opcional) se entregarán la semana del **16/12/2019** antes de que comience la sesión de prácticas de cada grupo. **En esa última sesión de prácticas cada grupo defenderá la práctica ante el profesor, donde se evaluará entre otras cosas que sea coherente con el trabajo realizado a lo largo de las sesiones de prácticas previas.**

Condiciones que deben cumplir las entregas

- La entrega se realizará mediante la web de la asignatura en la plataforma Moodle aulas.inf.uva.es
- Se entregará el código fuente del programa en lenguaje C realizado.
- El código debe estar desarrollado en C estándar y compilar en la máquina jair.lab.inf.uva.es.
- El programa debe ser concurrente.
- Para una mejor identificación, el nombre del programa indicará el grupo de laboratorio y primer apellido de cada miembro de la pareja; será algo así como: X2_GonzalezMerino_ParteX.c. IMPORTANTE: si los apellidos tienen acento no ponerlo en el nombre del programa.
- En las primeras líneas de cada fichero fuente enviado, debe aparecer como comentario el nombre, dos apellidos y DNI de cada uno de los autores.
- Se podrá incluir cualquier información adicional que se considere que puede ser útil para la evaluación de la práctica.
- Se valorará la claridad del código y la inclusión de comentarios aclaratorios.