

Consigna 1

Basado en la situación problemática, se debe analizar el comportamiento de los algoritmos presentados, estudiando sus particularidades desde el punto de vista de la implementación. Para llevar a cabo esta tarea es necesario ejecutar los algoritmos y luego de realizar diversas ejecuciones, documentar y explicar el comportamiento de tales programas.

Luego de ejecutar el primer algoritmo, se observa en pantalla que se imprime "Soy A", "Soy B" y "Soy C", en lo que aparentan 3 columnas diferentes (apariciencia otorgada gracias a las tabulaciones al comienzo del string).

Podemos ver que:

- El hilo 1 imprime "Soy A" 10 veces, con una demora de 100 milisegundos entre cada impresion.
- El hilo 2 imprime "Soy B" 15 veces, con una demora de 150 milisegundos entre cada impresion.
- El hilo 1 imprime "Soy C" 5 veces, con una demora de 300 milisegundos entre cada impresion.

Al final, se imprime "Fin" a modo de indicar que el programa ha finalizado. Ademas de retornar un valor 0.

El algoritmo 2 es similar, solo que imprime "Soy 1", "Soy 2" y "Soy 3". La diferencia mas notoria está mas bien en el código, ya que se crean los hilos de forma diferente:

`thread P[3]`

Esto crea 3 hilos como si estuvieran en un arreglo de 3 posiciones. Esos hilos son llamados como P[0], P[1] y P[3].

Consigna 2

Con base en el enfoque presentado en el algoritmo 2, desarrollar una nueva versión del programa en el que se ejecuten 15 procesos, de manera que el proceso i-ésimo se pause un tiempo aleatorio (definir un rango en milisegundos que pueda ser perceptible), y que por pantalla se muestre el mensaje correspondiente ("Soy el proceso 1", "Soy el proceso 2", ... , "Soy el proceso n") un número aleatorio de veces.

Algoritmo adjunto como Consigna2.cpp

Consigna 3

Tal como se puede observar, los algoritmos anteriores son claros cuando el comportamiento del proceso puede estar encapsulado en una única función. Pero termina siendo confuso cuando los procesos necesitan llevar a cabo tareas de mayor complejidad. Para ello, un enfoque más general consiste en encapsular (en una clase de objetos) el comportamiento de un proceso, creando las instancias necesarias, y de este modo lanzar la función deseada del objeto en un *thread*. Analice de qué manera podría implementar este enfoque, describa las características y esboce una implementación del algoritmo.

Tomando un paradigma de Programacion Orientada a Objetos, y un poco de IoT, en el programa Consigna3.cpp se observan 3 clases que representan electrodomesticos:

1. AireAcondicionado.
2. Ventana.
3. Termotanque.

Cada una con sus métodos, simulando que se puedan controlar a control remoto, como aumentar o disminuir la temperatura del aire, abrir o cerrar la ventana, o simplemente encender el termotanque.

A modo de ejemplo, he creado un electrodoméstico de cada uno y le he asignado un hilo a cada uno, de modo que cada accion del electrodomestico individual pueda ser ejecutada por un unico hilo.

Dado a que las funciones que usé en este ejemplo son del tipo void, no se muestra en pantalla cuando se ejecuten las acciones.