

---

**Ejercicio 1.**

Respecto al comportamiento obtenido es el esperado, la condicion de seguridad se mantiene y cada hebra termina su ejecución correctamente. La interpretación es que está bien hecho.

**Ejercicio 2.**

Respecto al comportamiento es algo distinto al anterior ejercicio, ya que, según pude comprobar en la primera ejecución del programa, los hilos se ejecutaban de forma secuencial, sin embargo, a partir de la 2º ejecución, empezaba a darse cierta “aletoriedad” a la hora de ejecutarse cada hilo.

Según he podido comprobar y he leído en numerosos articulos, al parecer este tipo de algoritmo en las cpu con mas de 4 cores (en mi caso quacore -8 hilos logicos de ejecución-), tanto en el lenguaje C como en sus derivados (Java, Python, Ruby, PHP, etc) al parecer por la forma de interpolar las propias hebras, en la primeras ejecuciones al no tener el programa en la memoria cache, tarda mas en ejecutar las hebras y parece que las ejecuta de forma secuencial.

Sin embargo, a partir de la 2º ejecución, al tener el programa cacheado, es mas rapido y entonces si que se nota la “aletoriedad” en la ejecución de las hebras.

Respecto a la interpretación, es que aunque trabajabemos sobre lenguajes que son multiplataforma como es el caso de Java, juega un gran papel las capacidades de la propia maquina en si y el propio Sistema Operativo. Incluso al parecer varia entre las propias distribuciones del propio Linux. En concreto entre las distribuciones minimalistas (Archlinux y Gentoo) y las distribuciones de tipo ‘out the box’ como Ubuntu, Linux Mint, Debian, etc.