

1) Ejecutar varias veces los códigos escritos en Python: `sinhilos.py` y `conhilos.py`

a) ¿Qué se puede notar con respecto al tiempo de ejecución? ¿Es predecible?

b) Comparar con un compañero el tiempo de ejecución. ¿Son iguales?

c) Ejecutar el archivo `suma_rasta.py` unas 10 veces, luego descomentar (borrar el #) las líneas 11, 12, 19 y 20 guardarlo y ejecutarlo otras 10 veces. ¿Qué pasó? ¿Por qué? Escribir la respuesta de estas preguntas en el `README.md` del repo de GitHub.

2) Burger Robot es una empresa multinacional de Software que se especializa en modelados de situaciones de empresas, en especial hamburgueserías, están teniendo problemas con un algoritmo para repartir las hamburguesas de forma equitativa entre los comensales. Como su equipo de expertos no pudo resolver este enigma, llamaron a un Hacker Ruso, el señor BinaryBorinsky78, este sr, escribía sus códigos con un sistema de encriptación muy particular, ponía las líneas de código importantes, mezcladas y en un documento comentado aparte. Pero ese no fue el problema, sino que no armo el algoritmo antes de que se lo llevara la Interpol.

a) Tu tarea será colocar las líneas de código en los lugares adecuados para poder resolver el problema de las hamburguesas de distribución equitativa.

b) Viendo la figura 2-22, como sería para el problema de las hamburguesas con 2 comensales y 8 hamburguesas.

1. a. En el proceso “`conhilos.py`” se puede observar que el tiempo de ejecución oscila entre los 4.013” hasta los 4.178”. En el proceso “`sinhilos.py`” el tiempo de ejecución oscila entre los 5.175” y los 5.203”. Esto es predecible ya que al ejecutar el proceso “`conhilos.py`” los hilos se ejecutan simultáneamente, es decir que mientras se ejecuta una tarea otras pueden estar en espera, en contra del “`sinhilos.py`” que primero se ejecuta el hilo 1, termina su ejecución, procede a iniciar el hilo 2, termina su ejecución y por último inicia el hilo 3 y termina.

b. Si son prácticamente los mismos, no cambian drásticamente, solo milésimas de segundo. Por lo que los procesos con multihilos son mas eficientes a la hora de ejecutar más de 1 proceso.

c. Lo que podemos observar en este programa, son 2 hilos que se ejecutan en paralelo, uno va a ejecutar el proceso “sumador” y el otro va al proceso “restador”.

Lo que sucede es que cuando se ejecuta el programa, cada proceso van a entrar a competir en la zona crítica por el recurso compartido (race condition) que en este caso es la variable global “acumulador” .

Al producirse tan rápido estos procesos en ocasiones no es perceptible el error.

Pero al quitar las líneas de comentario, se le agraga un ciclo `for`, sumándole tiempo de iteración en cada uno de los procesos, por lo que se puede observar mejor el error que se produce al ejecutar ambos procesos con el recurso compartido