

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 armó 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 ente 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 sucede en el programa “`suma_rasta.py`” al tener las líneas (**`for i in range (1000):`** **`pass`**)

comentadas es que la función sumador y restador se ejecutan en hilos separados, lo que permite que se desarrollen de manera concurrentemente, uno suma 5 al acumulador y lo guarda en la variable “a” y el resultado de “a” se guarda en contador, todo esto se realiza en un rango de 100.000 iteraciones. Lo mismo ocurre en restador, pero disminuyéndole 5 también al acumulador.

En cambio, si ahora descomentamos las líneas (**`for i in range (1000):`** **`pass`**) esto añade 1000 iteraciones mas dentro del primer ciclo for de 100.000, para sumar tiempo denominado de “espera ocupada”, ocupando un tiempo de procesamiento innecesario. Lo que produce que al compartir una misma variable global que es “acumulador”, experimentan un race condición. Ya que un proceso puede sumar y restar en la misma variable, pero en hilos separados y el resultado siempre va a variar dependiendo del que llegue primero a guardarse en el acumulador, y en esa competencia puede ocurrir que la información no sea la esperada.