

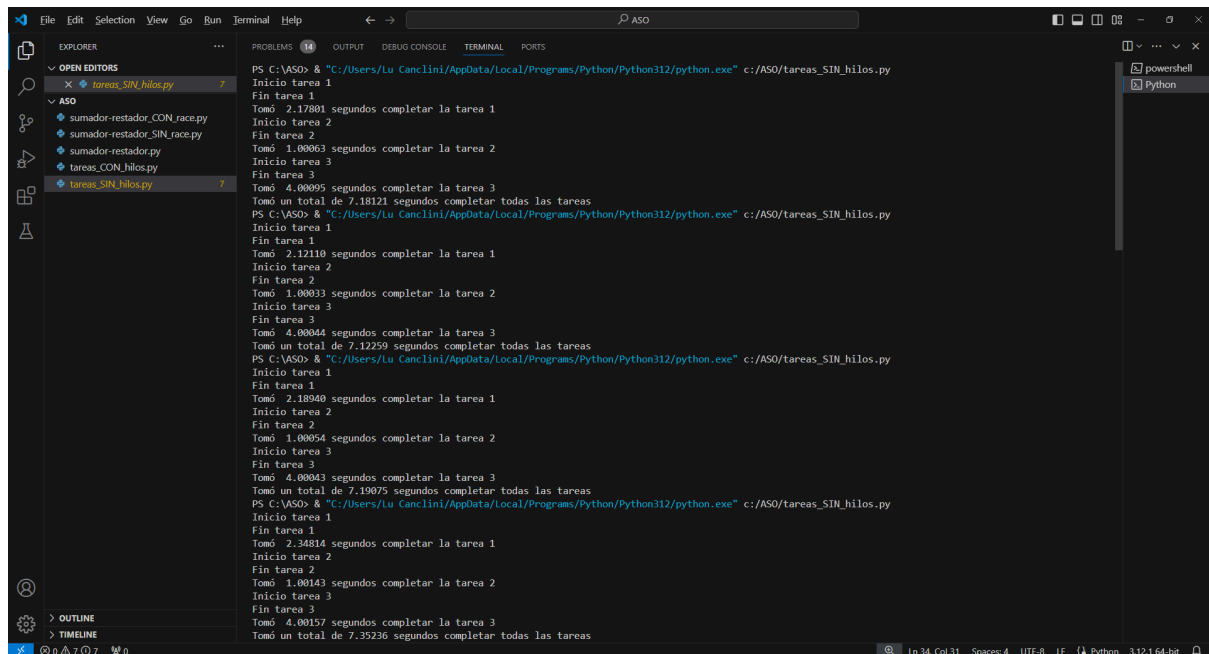
TRABAJO PRÁCTICO N°3

Arquitectura y sistemas operativos

1) Hilos

tareas_SIN_hilos.py

Ejecución



```
PS C:\ASO> & "C:/Users/Lu Cancini/AppData/Local/Programs/Python/Python312/python.exe" c:/ASO/tareas_SIN_hilos.py
Inicio tarea 1
Fin tarea 1
Tomó 2.17881 segundos completar la tarea 1
Inicio tarea 2
Fin tarea 2
Tomó 1.08063 segundos completar la tarea 2
Inicio tarea 3
Fin tarea 3
Tomó 4.00095 segundos completar la tarea 3
Tomó un total de 7.18121 segundos completar todas las tareas
PS C:\ASO> & "C:/Users/Lu Cancini/AppData/Local/Programs/Python/Python312/python.exe" c:/ASO/tareas_SIN_hilos.py
Inicio tarea 1
Fin tarea 1
Tomó 2.12110 segundos completar la tarea 1
Inicio tarea 2
Fin tarea 2
Tomó 1.08033 segundos completar la tarea 2
Inicio tarea 3
Fin tarea 3
Tomó 4.00044 segundos completar la tarea 3
Tomó un total de 7.12259 segundos completar todas las tareas
PS C:\ASO> & "C:/Users/Lu Cancini/AppData/Local/Programs/Python/Python312/python.exe" c:/ASO/tareas_SIN_hilos.py
Inicio tarea 1
Fin tarea 1
Tomó 2.18040 segundos completar la tarea 1
Inicio tarea 2
Fin tarea 2
Tomó 1.08054 segundos completar la tarea 2
Inicio tarea 3
Fin tarea 3
Tomó 4.00043 segundos completar la tarea 3
Tomó un total de 7.19075 segundos completar todas las tareas
PS C:\ASO> & "C:/Users/Lu Cancini/AppData/Local/Programs/Python/Python312/python.exe" c:/ASO/tareas_SIN_hilos.py
Inicio tarea 1
Fin tarea 1
Tomó 2.34814 segundos completar la tarea 1
Inicio tarea 2
Fin tarea 2
Tomó 1.00143 segundos completar la tarea 2
Inicio tarea 3
Fin tarea 3
Tomó 4.00157 segundos completar la tarea 3
Tomó un total de 7.35236 segundos completar todas las tareas
```

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

El tiempo de ejecución no es tan predecible a simple vista ya que al estar las tres funciones en un mismo hilo se ejecuta la tarea 1 primero, luego la tarea 2 la cual “duerme” por un segundo y por último la tarea 3, que duerme 4 segundos más.

- Nombrar un proceso o función de la vida real que pueden ser considerados procesos de “máxima velocidad posible” que dependen casi exclusivamente de la velocidad de la máquina que los ejecuta (ej. Ordenar una lista)

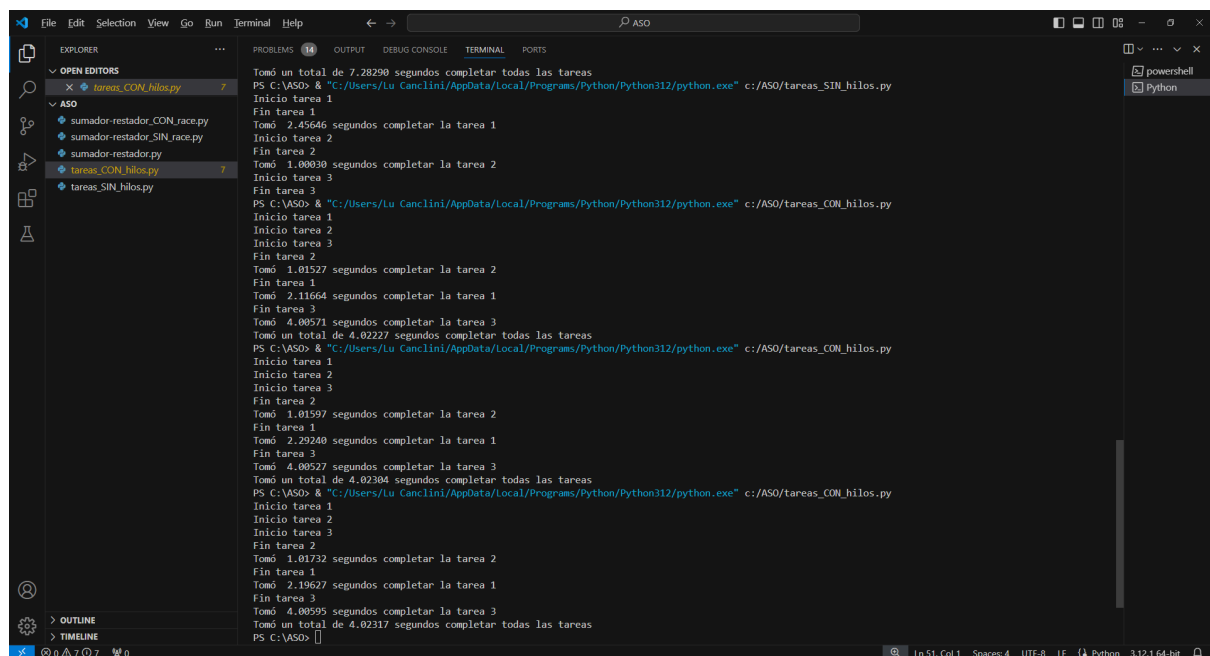
Un ejemplo es la compilación de grandes proyectos de software, como el kernel de Linux, es intensivo en CPU y puede ser acelerado con un hardware más rápido, por lo tanto, un proceso de “máxima velocidad posible” depende casi exclusivamente de la velocidad de la máquina que lo ejecuta.

- Nombrar un proceso o función de la vida real que pueden ser considerados procesos de “velocidad de respuesta no dependiente de la velocidad de procesamiento” o que sea de naturaleza impredecible o externa (ej. Leer un archivo externo)

Un proceso de la vida real que podría considerarse como "velocidad de respuesta no dependiente de la velocidad de procesamiento" es la interacción con un servidor remoto a través de una red, ya que Cuando un programa cliente solicita datos o servicios a un servidor remoto a través de una red la velocidad de respuesta puede variar por factores externos como la latencia de la red, el ancho de banda, la carga del servidor, entre otros.

tareas_CON_hilos.py

Ejecución



```
Tomó un total de 7.28290 segundos completar todas las tareas
PS C:\ASO> & "C:/Users/Lu Cancini/AppData/Local/Programs/Python/Python312/python.exe" c:/ASO/tareas_SIN_hilos.py
Inicio tarea 1
Fin tarea 1
Tomó 2.45646 segundos completar la tarea 1
Inicio tarea 2
Fin tarea 2
Tomó 1.00030 segundos completar la tarea 2
Inicio tarea 3
Fin tarea 3
PS C:\ASO> & "C:/Users/Lu Cancini/AppData/Local/Programs/Python/Python312/python.exe" c:/ASO/tareas_CON_hilos.py
Inicio tarea 1
Inicio tarea 2
Inicio tarea 3
Fin tarea 2
Tomó 1.01527 segundos completar la tarea 2
Fin tarea 1
Tomó 2.11664 segundos completar la tarea 1
Fin tarea 3
Tomó 4.00571 segundos completar la tarea 3
Tomó un total de 4.02227 segundos completar todas las tareas
PS C:\ASO> & "C:/Users/Lu Cancini/AppData/Local/Programs/Python/Python312/python.exe" c:/ASO/tareas_CON_hilos.py
Inicio tarea 1
Inicio tarea 2
Inicio tarea 3
Fin tarea 2
Tomó 1.01597 segundos completar la tarea 2
Fin tarea 1
Tomó 2.29240 segundos completar la tarea 1
Fin tarea 3
Tomó 4.00527 segundos completar la tarea 3
Tomó un total de 4.02304 segundos completar todas las tareas
PS C:\ASO> & "C:/Users/Lu Cancini/AppData/Local/Programs/Python/Python312/python.exe" c:/ASO/tareas_CON_hilos.py
Inicio tarea 1
Inicio tarea 2
Inicio tarea 3
Fin tarea 2
Tomó 1.01732 segundos completar la tarea 2
Fin tarea 1
Tomó 2.19627 segundos completar la tarea 1
Fin tarea 3
Tomó 4.00959 segundos completar la tarea 3
Tomó un total de 4.02317 segundos completar todas las tareas
PS C:\ASO>
```

- ¿Qué se puede notar con respecto al tiempo de ejecución? ¿Se mejoró el tiempo de respuesta con respecto al mismo programa sin hilos?

Con respecto al tiempo, se reduce considerablemente, pasando de 7 a 4 segundos aprox

- ¿Completan las funciones su ejecución en el orden establecido?

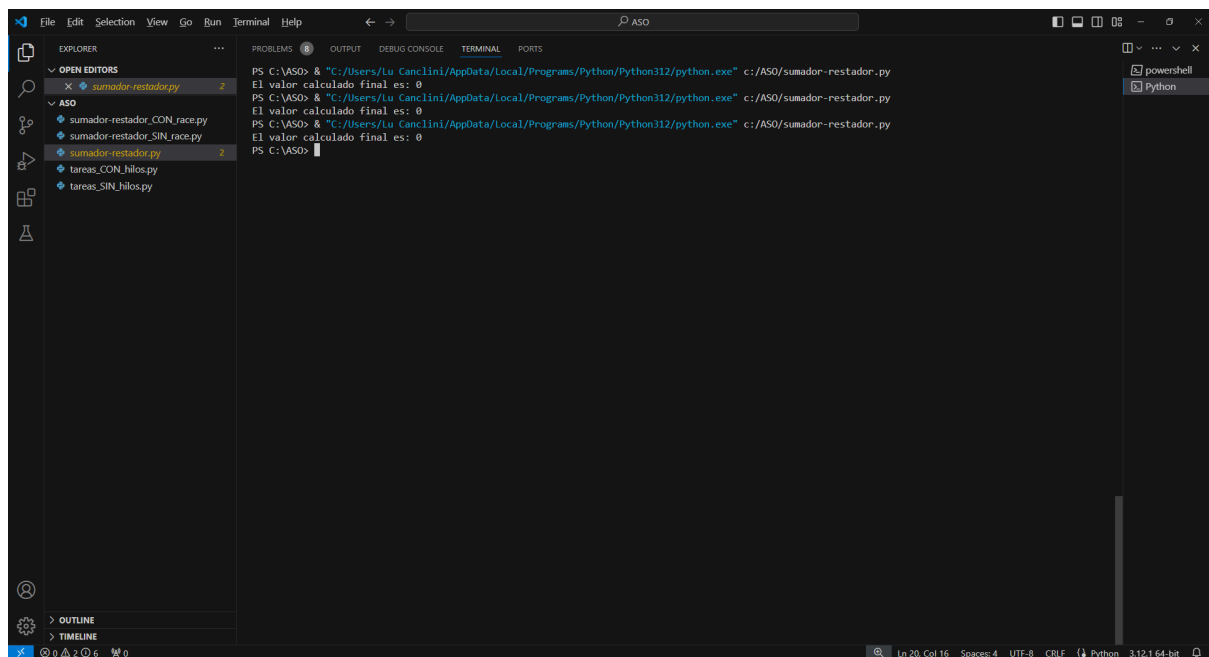
No necesariamente, gracias a los hilos las tres tareas se ejecutan de simultáneamente, a diferencia de la ejecución sin hilos, la tarea 2 duerme por un segundo y logra ejecutarse primero que la tarea 1, ya que esta última depende de la velocidad de la CPU y finalmente la tarea 3 se ejecuta luego de haber “dormido” por 4 segundos.

- Nombrar un escenario real donde el multi-hilado puede mejorar considerablemente el tiempo de respuesta de un sistema (ej. Carga de una página WEB en un navegador)

Un escenario real donde el uso de múltiples hilos puede mejorar considerablemente el tiempo de respuesta de un sistema es en el procesamiento de lotes en aplicaciones de procesamiento de datos, ya que, en aplicaciones que procesan grandes cantidades de datos, como análisis de datos, procesamiento de imágenes o procesamiento de vídeo, el uso de múltiples hilos puede distribuir la carga de trabajo y acelerar el tiempo de procesamiento.

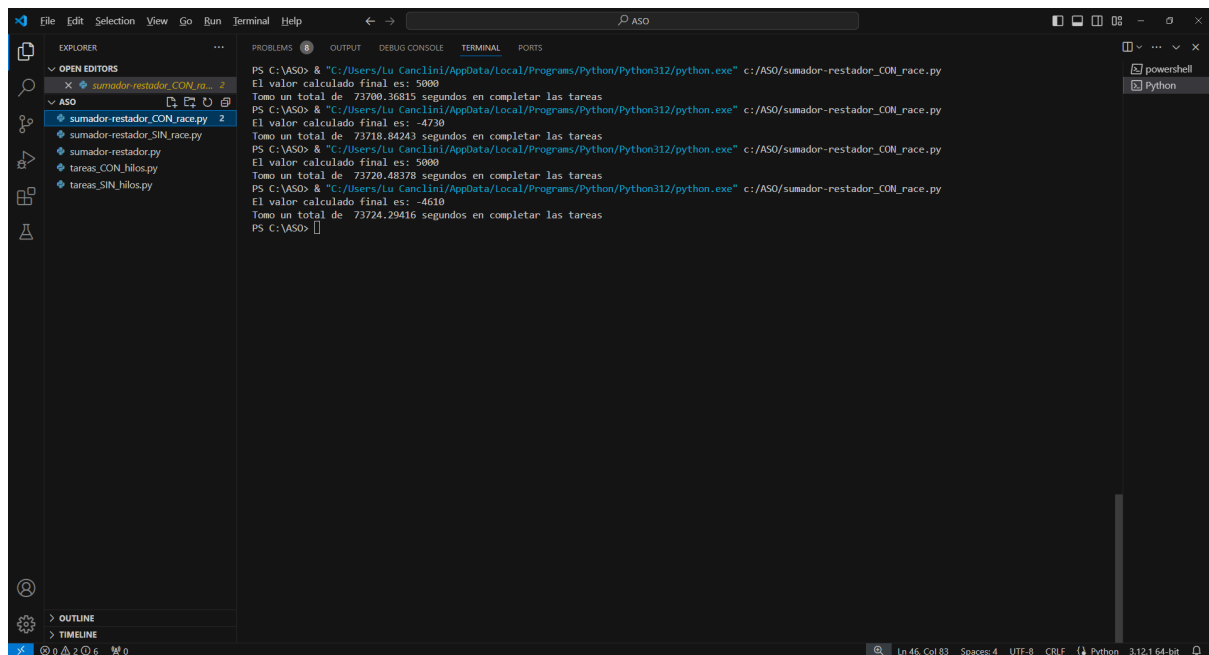
2) Condición de carrera (Race Condition)

sumador-restador.py



The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal displays the execution of a Python script named `sumador-restador.py` using the command `PS C:\ASO> & "C:/Users/Lu Cancilini/AppData/Local/Programs/Python/Python312/python.exe" c:/ASO/sumador-restador.py`. The output of the script is `El valor calculado final es: 0`, which is printed three times, indicating that the script was executed multiple times. The Explorer panel on the left shows the file structure of the project, including `sumador-restador.py`, `sumador-restador_CON_race.py`, `sumador-restador_SIN_race.py`, `tareas_CON_hilos.py`, and `tareas_SIN_hilos.py`. The status bar at the bottom indicates the current file is `sumador-restador.py` at line 20, column 16, with a UTF-8 encoding and CRLF line endings.

sumador-restador_CON_race.py



```
PS C:\ASO> & "C:/Users/Lu Canclini/AppData/Local/Programs/Python/python312/python.exe" c:/ASO/sumador-restador_CON_race.py
El valor calculado final es: 5000
Tomo un total de 73700.36815 segundos en completar las tareas
PS C:\ASO> & "C:/Users/Lu Canclini/AppData/Local/Programs/Python/python312/python.exe" c:/ASO/sumador-restador_CON_race.py
El valor calculado final es: -4730
Tomo un total de 73718.84243 segundos en completar las tareas
PS C:\ASO> & "C:/Users/Lu Canclini/AppData/Local/Programs/Python/python312/python.exe" c:/ASO/sumador-restador_CON_race.py
El valor calculado final es: 5000
Tomo un total de 73720.48378 segundos en completar las tareas
PS C:\ASO> & "C:/Users/Lu Canclini/AppData/Local/Programs/Python/python312/python.exe" c:/ASO/sumador-restador_CON_race.py
El valor calculado final es: -4610
Tomo un total de 73724.29416 segundos en completar las tareas
PS C:\ASO>
```

- ¿Qué sucede con el valor final del acumulador?

El valor final del acumulador varía entre 5000 y -4000 aprox

- ¿Por qué sucede esto?

Sucede porque dos funciones del programa intentan cambiar el valor del acumulador al mismo tiempo, y una parte puede modificarlo antes que la otra, por lo tanto, el resultado final varía en ese rango de números.

¿Cómo se puede corregir esta condición de carrera sin dejar de utilizar hilos?

Implementando algún tipo de función que permita bloquear un recurso cuando se esta utilizando.

3) Detección y corrección del problema

Ejecución

```
PS C:\ASO> & "C:/Users/Lu Canclini/AppData/Local/Programs/Python/Python312/python.exe" c:/ASO/sumador-restador_SIN_race.py
El valor calculado final es: 0
PS C:\ASO> & "C:/Users/Lu Canclini/AppData/Local/Programs/Python/Python312/python.exe" c:/ASO/sumador-restador_SIN_race.py
El valor calculado final es: 0
PS C:\ASO> & "C:/Users/Lu Canclini/AppData/Local/Programs/Python/Python312/python.exe" c:/ASO/sumador-restador_SIN_race.py
El valor calculado final es: 0
PS C:\ASO> & "C:/Users/Lu Canclini/AppData/Local/Programs/Python/Python312/python.exe" c:/ASO/sumador-restador_SIN_race.py
El valor calculado final es: 0
PS C:\ASO> & "C:/Users/Lu Canclini/AppData/Local/Programs/Python/Python312/python.exe" c:/ASO/sumador-restador_SIN_race.py
El valor calculado final es: 0
PS C:\ASO> & "C:/Users/Lu Canclini/AppData/Local/Programs/Python/Python312/python.exe" c:/ASO/sumador-restador_SIN_race.py
El valor calculado final es: 0
PS C:\ASO>
```

- ¿Qué sucede con el valor final del acumulador?

El valor siempre es igual a 0 debido a que el recurso se bloquea mientras una de las dos funciones lo utiliza y se desbloquea cuando este finaliza para que la otra función también lo pueda modificar.