**Trabajo Practico Nº 3**

**Primera parte:**

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

Con respecto al tiempo de ejecución se puede notar que, es variable ya que depende de la velocidad real de la maquina donde se esta ejecutando.

No es predecible, porque depende de la velocidad del procesador de cada máquina.

**- 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 sería la vista previa de un archivo en la pc, ya que depende de la velocidad de procesamiento de la misma.

**- 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 ejemplo es el reconocimiento de voz de las aplicaciones de asistentes virtuales (como Siri de Apple, Google Assistance), porque depende de la precisión y eficiencia de los algoritmos de reconocimiento de voz y la calidad de audio recibido ( puede haber interferencia de ruidos )

**Segunda parte:**

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

Cuando ejecutas el código varias veces, es probable que se note que el tiempo que tarda en completarse puede variar.

Con el uso de hilos, es probable que, el tiempo de respuesta mejora en comparación con el mismo programa sin hilos. Esto se debe a que los hilos permiten que las tareas se realicen al mismo tiempo, lo que hace que el programa sea más eficiente y rápido.

**– ¿Completan las funciones su ejecución en el orden establecido?**

Para el orden de ejecución, no hay una garantía de que las funciones se completen en el mismo orden en que se iniciaron los hilos. Cada hilo puede ejecutarse de manera independiente. Además el orden en que se completen las funciones puede variar dependiendo de cómo el sistema operativo gestione los hilos y de la disponibilidad de recursos en la computadora.

**- 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 de mejora del tiempo de respuesta con multi-hilo donde el uso de hilos puede mejorar significativamente el tiempo de respuesta es en un navegador web que carga páginas de internet.

Cuando un navegador carga una página web, suele descargar varios elementos al mismo tiempo (como imágenes, scripts y estilos).

Si el navegador utiliza hilos para manejar estas descargas de forma simultánea, la página se cargará más rápido, ya que múltiples elementos se descargan al mismo tiempo en lugar de esperar a que uno se descargue antes de comenzar con el siguiente. Esto hace que la experiencia de navegación sea más fluida para el usuario.

**Tercera Parte:**

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

Respecto al tiempo de ejecución, es posible notar que el programa tarda más de lo esperado en completarse debido a la función time.sleep(0**)** que se usa en cada iteración de los subprocesos.

**- ¿Qué sucede con el valor final del acumulador?**

El valor final del acumulador podría no ser el valor esperado. Cuando varios subprocesos intentan acceder y modificar el acumulador simultáneamente, puede haber interrupciones entre la lectura y la actualización de la variable, lo que hace un valor final impredecible.

**- ¿Por qué sucede esto?**

En este caso, dado que tanto el sumador como el restador acceden y modifican al acumulador simultáneamente, podrían ocurrir cambios de contexto en medio de la actualización del valor de acumulador. Esto significa que un subproceso podría leer el valor de acumulador, luego ser interrumpido, y otro subproceso podría actualizar acumulador antes de que el primer subproceso complete su operación.

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

Para corregir esta condición de carrera sin dejar de utilizar hilos, se puede emplear un mecanismo de sincronización, como un bloqueo (lock), para garantizar que solo un subproceso pueda acceder y modificar el acumulador a la vez. Esto evita que ocurran interrupciones entre la lectura y la actualización de la variable.

**Cuarta parte:**

**- ¿Qué sucede con el valor final del acumulador?**

Dado que se está utilizando un bloqueo para sincronizar el acceso al acumulador, el valor final de acumulador debería ser predecible y correcto. En este caso, como ambos hilos acceden y modifican al acumulador de manera alternada y coordinada, se puede esperar que el valor final sea predecible.

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

Con respecto al tiempo de ejecución, este podría ser más largo que el tiempo necesario para ejecutar 1000 iteraciones de sumar 5 y restar 5 debido al uso de time.sleep(0) en cada iteración. Aunque este tiempo de espera es mínimo, se acumula a medida que los hilos realizan múltiples iteraciones. Sin embargo, en este caso, como la sección crítica está protegida por un bloqueo, el tiempo de espera debería ser menor en comparación con el código original, donde la condición de carrera podría provocar retrasos adicionales debido a los cambios de contexto inesperados.