Arquitectura y Sistemas Operativos

Tecnicatura Universitaria en Programación (TUP) – UTN FRBB

Trabajo Práctico Nº 3

Procesos multihilados (multi-threads)

2) Hilos

Programa "tareas_SIN_hilos.py":

Con respecto al tiempo de ejecución podemos notar que si bien los tiempos no son exactamente iguales, podríamos que decir que la **Tarea n1** tarda aproximadamente 1,5 segundos, la **Tarea n2** tarda aproximadamente 1 segundo y la **Tarea n3** tarda aproximadamente 4 segundos, dando un total de 6,5 segundos aproximados en completar las 3 tareas.

La descarga de archivos de Internet es un ejemplo de proceso de "máxima velocidad posible". La velocidad a la que descargas un archivo depende principalmente de la velocidad de tu conexión a Internet y de la capacidad de tu dispositivo para procesar la información. Cuanto más rápida sea tu conexión y más potente sea tu dispositivo, más rápido será el proceso de descarga. No se puede descargar más rápido que lo que tu conexión y dispositivo te permiten.

La recepción de emails es un ejemplo de proceso de "velocidad de respuesta no dependiente de la velocidad de procesamiento". No importa cuán rápido sea tu computadora o conexión a Internet, la velocidad a la que recibis mails depende de factores externos, como la velocidad de los servidores de correo electrónico. Podes tener una muy buena computadora, pero si el servidor de correo está lento, tus correos van a llegar igual de despacio.

Programa “tareas_CON_hilos.py”:

Podemos ver que con respecto al tiempo de ejecución por tarea es casi idéntico al programa sin hilos, pero podemos apreciar que hubo una mejora de aproximadamente 2 segundos para completar todas las tareas.

Las tareas se ejecutan en el orden establecido pero su finalización es otra historia, en mi caso la primera tarea en finalizar es la **N2** seguida por la **N1** y finalizando con la **N3.**

La búsqueda en un motor de búsqueda como Google es un escenario donde el multi-hilo puede marcar la diferencia en el tiempo de respuesta. Cuando realizas una búsqueda, el motor de búsqueda puede emplear múltiples hilos para procesar varias consultas al mismo tiempo. Esto

significa que puede devolver los resultados más rápidamente, ya que está distribuyendo la carga de trabajo entre diferentes hilos, acelerando así el tiempo de respuesta para el usuario.

3) Condición de carrera (Race Condition)

Con respecto al tiempo de ejecución como el código descargado es diferente al de las imágenes y no podemos ver números exactos, podría decir que los 2 códigos tardan aproximadamente el mismo tiempo en ejecutarse.

Podemos ver que en el programa"sumador-restador.py" el valor final siempre es 0 mientras que en el programa "sumador-restador_CON_race.py" el valor final no es confiable y varia entre números positivos, negativos y 0.

Esto sucede porque ambos hilos leen, modifican y escriben el valor de acumulador de manera concurrente. Cuando un hilo lee el valor de acumulador y lo almacena en una variable temporal tmp, el otro hilo puede modificar acumulador antes de que el primer hilo tenga la oportunidad de actualizarlo. Esto lleva a cambios de contexto en medio de acceso a una zona crítica, causando inconsistencias en el valor final de acumulador.

Podemos usar una protección para asegurarnos de que solo un hilo cambie el acumulador a la vez.

4- Detección y corrección del problema

Con este último código, el valor final del acumulador será 0. Esto se debe a que se utiliza un Lock para sincronizar el acceso y la modificación del acumulador entre los hilos sumador y restador.

El tiempo de ejecución sin tener números exactos es prácticamente el mismo, no se puede ver diferencia a simple vista.