**¿Qué es la programación multiproceso y los dos tipos?**

La programación multiproceso es un enfoque de la programación la cual permite aprovechar al máximo, la capacidad de procesamiento de una computadora mediante el uso de varios procesadores o núcleos de procesamiento en paralelo para realizar tareas.

Existen dos tipos de multiprocesos: programación de procesos o de hilos. El más normal es la programación de procesos es aquella que crea varios procesos independientes, cada uno con su propia memoria y conjunto de instrucciones, que se ejecutan simultáneamente en paralelo. Para lograrlo se utilizan varios procesos diferentes, cada uno con su propio espacio de memoria e instrucciones. Estos se ejecutan en diferentes procesadores, permitiendo realizar varias tareas al mismo tiempo y con ello aumentando el rendimiento de la computadora. Un ejemplo de esto, es una pantalla que a la vez de estar escribiendo, también se escucha música.

El proceso de hilos utiliza un solo proceso que utiliza muchos hilos de ejecución, cada hilo posee su propio tipo de instrucciones y acceso a la memoria compartida. Estos hilos comparten el mismo espacio de memoria y los recursos del sistema. Por ejemplo Pongo como ejemplo, una factura con muchos artículos, donde cada artículo se calcula en un core diferente calculando el importe en función de la cantidad, los descuentos, el IVA etc.

Diagrama

Descripción generada automáticamenteAmbos son casi como temas diferentes y cada uno posee problemas diferentes.

Este tipo de programación es esencial para desarrollar aplicaciones con alto rendimiento y escalabilidad. Sin embargo también puede ser bastante compleja y hay que tener una gran cantidad de cuidado para evitar distintos problemas como la condición de carrera, la sincronización de hilos y los problemas de memoria compartida.

En el primer caso, los problemas pueden ocurrir por utilizar simultáneamente en dos programas diferentes un mismo registro o cualquier otro servicio ligado al Hardware, la impresora, el altavoz, la misma pantalla y suele ser un problema a solucionar por el SO, Pero se encuentran algunas excepciones.

En resumen la programación multiproceso es esencial para desarrollar aplicaciones de alto rendimiento y escalabilidad, pero también requiere una gran cantidad de cuidado y atención para evitar problemas comunes.

**¿COMO SE UTILIZA LA PROGRAMACIÓN DE MULTIPROCESOS EN PYTHON?**

Texto

Descripción generada automáticamenteLa programación multiproceso en Python se utiliza para ejecutar varios procesos concurrentes en un sistema operativo. Esto se puede lograr con la biblioteca/modulo ***multiprocessing*** de python***.*** Este modulo usa una interfaz como la de ***threading*** pero a diferencia de esta, ***multiprocessing*** utiliza procesos en lugar de hilos.

Para empezar a utilizar ***multiprocessing*** debemos crear un objeto llamado Procces, este define la función de ejecutar en un proceso separado, y luego se llama al método ***start()*** del objeto para iniciar el proceso.

Texto

Descripción generada automáticamente

Para distribuir tareas entre varios procesos podemos utilizar la función ‘***Pool.map()***’ o ‘***Pool.aply()***’

Texto

Descripción generada automáticamentePara comunicarse entre procesos se utiliza el objeto ‘***Queue’***. Esto sirve para obtener y almacenar objetos de forma ordenada.

Esto es útil en caso de que se quiera enviar un gran número de mensajes entre los diferentes procesos.

Gracias a este objeto se pueden evitar interbloqueos debido a que los procesos solo se pueden ingresar a los datos en la cola en el mismo orden que fueron insertados.

Texto

Descripción generada automáticamentePara sincronizar los procesos lo mejor es utilizar ‘***Lock’,*** que sirve para que solo un proceso tenga acceso a un recurso o sección crítica a la vez. Esto evita conflictos y errores. Su utilización es importante cuando varios procesos acceden a un recurso compartido, ya que si ambos llegasen a acceder al mismo tiempo podrían causar conflictos de acceso y datos corruptos

Texto

Descripción generada automáticamente

Para crear recursos compartidos entre objetos se utiliza el ‘***manager’***. El ‘***manager’*** proporciona una interfaz para crear y administrar objetos compartidos como, listas y diccionarios, los cuales pueden ser accedidos y modificados por distintos procesos de forma segura. Los objetos de este se almacenan en un espacio de memoria separado y se accede a estos mediante una interfaz remoto. Es decir, que los cambios hechos por un proceso en un objeto, son visibles a los otros procesos. Algunas de sus ventajas es que posee una interfaz muy simple para crear y acceder a objetos compartidos. Aunque los objetos creados mediante ‘***manager’*** son más lentos que los objetos normales

Nombre: Enrique Martínez Ferrer

Curso: Dam2

Asignatura: PSP

Fecha:23/01/2023