Carlos Francés Sánchez - DAM2

PROGRAMACIÓN MULTIPROCESO E HILOS EN PYTHON

DEFINCIÓN:

La programación multiproceso es en una técnica de manejo de los procesos que consiste en la utilización de varios procesadores para ejecutar varias tareas de manera simultanea. También llamado “multitarea” o “multitasking”, los procesadores modernos cuentan con varios núcleos que nos permiten el desarrollo de dicha técnica, aumentando considerablemente la eficiencia a la hora de ejecutar programas.

De esta definición podemos sacar en claro que la programación multiproceso actualmente es un sistema prácticamente de obligado uso para el correcto y eficiente funcionamiento de un programa, ya que utiliza de mejor manera las capacidades de los procesadores.

Además de este concepto de utilizar varios núcleos de manera simultanea, existe el concepto de hilos, que no es más que una distinción más ligera de los procesos, y que se encuentran dentro de este ámbito.
De esta manera, un procesador actual puede manejar diversos procesos de manera simultánea (multiproceso) y, además, diversos hilos dentro de cada uno de estos procesos, de manera que la fluidez en el procesamiento de tareas y el paso de información es lo más rápido posible.

Carlos Francés Sánchez - DAM2

PROGRAMACIÓN MULTIPROCESO EN PYTHON

Python es un lenguaje que suele apoyarse en librerías para realizar todo tipo de tareas. Una de las librerías que pueden ayudarnos a gestionar el multiproceso y los hilos es la librería **multiprocessing**

```
1  from multiprocessing import Process
2
3
4  def greeting():
5      print 'hello world'
6
7  if __name__ == '__main__':
8      p = Process(target=greeting)
9      p.start()
10     p.join()
```

En este ejemplo podemos ver como importar la librería a nuestro archivo para poder trabajar con ella. Si ejecutamos este archivo como script, se creará un objeto de tipo proceso para la función **greeting(),** el cual se va a ejecutar al llegar a **p.start()**, y desaparecerá al llegar a **p.join()**.

También podemos crear un proceso con argumentos de la siguiente manera:

```
1  from multiprocessing import Process
2
3
4  def greeting(name):
5      print 'hello' + " " + name
6
7  if __name__ == '__main__':
8      p = Process(target=greeting, args=('world',))
9      p.start()
10     p.join()
```

multiprocessing nos permite la creación de varios procesos a la vez.
Es tan simple como definir ambos procesos y después ejecutarlos con **start()** o terminarlos con **end()**, tal y como se muestra en la siguiente imagen

```python
from multiprocessing import Process


def square(x):

    for x in numbers:
        print('%s squared is %s' % (x, x**2))


def is_even(x):

    for x in numbers:
        if x % 2 == 0:
            print('%s is an even number ' % (x))


if __name__ == '__main__':
    numbers = [43, 50, 5, 98, 34, 35]

    p1 = Process(target=square, args=('x',))
    p2 = Process(target=is_even, args=('x',))

    p1.start()
    p2.start()

    p1.join()
    p2.join()

    print "Done"

#result

43 squared  is  1849
50 squared  is  2500
5 squared  is  25
98 squared  is  9604
34 squared  is  1156
35 squared  is  1225
50 is an even number
98 is an even number
34 is an even number
Done

```

La librería contiene una extensa cantidad de métodos que nos facilitan la gestión de procesos e hilos. Para conocerlos todos, basta con viajar a este link: https://docs.python.org/es/3.9/library/multiprocessing.html para poder leer toda la documentación.