

Universidad de Guadalajara

Ingeniería en computación



Computación Tolerante a Fallas 2023^a

“An Introduction to Scaling Distributed
Python Applications”

Olguín Hernández Jair Benjamín.

217439707

Los hilos y los demonios son conceptos importantes en programación que se utilizan para controlar la ejecución de programas en Python. Un hilo es una unidad básica de ejecución que se utiliza para ejecutar tareas en paralelo dentro de un mismo proceso. En Python, se pueden crear hilos utilizando el módulo **threading**, y cada hilo puede ejecutar una función o método específico. Los hilos comparten la misma memoria y recursos del proceso padre, lo que permite una comunicación y sincronización entre ellos.

Los demonios, por otro lado, son procesos en segundo plano que se ejecutan de forma independiente del programa principal y que no están diseñados para interactuar con el usuario directamente. En Python, se pueden crear demonios utilizando el módulo **daemon**. Los demonios se utilizan para ejecutar tareas de forma automatizada y continuada, como por ejemplo, tareas de mantenimiento, actualización de datos, etc.

La principal diferencia entre los hilos y los demonios es que los hilos se ejecutan dentro del mismo proceso principal y comparten los mismos recursos y memoria, mientras que los demonios se ejecutan de forma independiente y no están diseñados para interactuar con el usuario directamente. En ambos casos, se utilizan para mejorar la eficiencia y el rendimiento de los programas, y para realizar tareas en paralelo o en segundo plano mientras el programa principal sigue ejecutándose.

Programa:

```
import threading
import multiprocessing
import time
import daemon

# Función que simula una tarea pesada y larga
def tarea():
    for i in range(100):
        print("Tarea en proceso...")
        time.sleep(1)
    print("Tarea completada.")

# Ejecución sin hilos ni demonios
print("Ejecución sin hilos ni demonios:")
tarea()
```

```

# Ejecución con hilos
print("\nEjecución con hilos:")
hilos = []
for i in range(3):
    hilo = threading.Thread(target=tarea)
    hilos.append(hilo)
    hilo.start()

for hilo in hilos:
    hilo.join()

# Ejecución como demonio
print("\nEjecución como demonio:")
def run():
    tarea()

with daemon.DaemonContext():
    run()

```

Explicación del código:

En la función `tarea()` creamos una tarea que simplemente imprime "Tarea en proceso..." cinco veces, con un retardo de un segundo entre cada impresión, y luego imprime "Tarea completada.".

En la primera ejecución, simplemente llamamos a la función `tarea()` sin utilizar hilos ni demonios.

En la segunda ejecución, creamos tres hilos, cada uno ejecutando la función `tarea()`, y esperamos a que todos los hilos terminen con el método `join()`.

En la tercera ejecución, envolvemos la función `tarea()` en una función `run()` que se ejecutará como un demonio utilizando el módulo `daemon`. El demonio se encarga de ejecutar la función en segundo plano, de forma que el programa principal puede seguir ejecutándose. En este caso, simplemente llamamos a la función `run()` dentro de un contexto de demonio (`with daemon.DaemonContext():`), lo que crea y ejecuta el demonio.

Conclusión:

En conclusión, el uso de hilos y demonios es importante en programación porque permiten mejorar la eficiencia y el rendimiento de los programas al ejecutar tareas en paralelo o en segundo plano. Los hilos son útiles cuando se necesita ejecutar múltiples tareas simultáneamente dentro de un mismo proceso, y permiten una comunicación y sincronización entre ellos a través del uso de locks y semáforos. Por otro lado, los demonios son útiles cuando se necesita ejecutar tareas de forma automatizada y continuada en segundo plano, sin necesidad de interacción con el usuario. Los demonios se pueden utilizar para realizar tareas de mantenimiento, actualización de datos, entre otras.

Fuentes:

Python 3 para impacientes. (2016, 16 diciembre). python-para-impacientes. Recuperado 12 de marzo de 2023, de <https://python-para-impacientes.blogspot.com/2016/12/threading-programacion-con-hilos-i.html>

An Introduction to Scaling Distributed Python Applications. (n.d.). Educative: Interactive Courses for Software Developers. <https://www.educative.io/blog/scaling-in-python>

GeeksforGeeks. (2021, August 17). *Python Daemon Threads.* <https://www.geeksforgeeks.org/python-daemon-threads/>

How to Use Daemon Threads in Python. (2022, September 11). Super Fast Python. <https://superfastpython.com/daemon-threads-in-python/>