

Python – Interrompendo um Thread

Em um programa multithread, pode ser necessário interromper uma tarefa em um novo thread. Isso pode ocorrer por vários motivos, como: (a) O resultado da tarefa não é mais necessário ou (b) o resultado da tarefa foi extraviado ou (c) O aplicativo está sendo encerrado.

Um thread pode ser interrompido usando um objeto `threading.Event`. Um objeto `Event` gerencia o estado de um sinalizador interno que pode ser definido ou não definido.

Quando um novo objeto `Event` é criado, seu sinalizador não é definido (falso) para iniciar. Se seu método `set()` for chamado por um thread, seu valor de flag poderá ser verificado em outro thread. Se for verdade, você pode encerrar sua atividade.

Exemplo

Neste exemplo, temos uma classe `MyThread`. Seu objeto começa a executar o método `run()`. O thread principal fica suspenso por um determinado período e então define um evento. Até que o evento seja detectado, o loop no método `run()` continua. Assim que o evento for detectado, o loop termina.

```
from time import sleep
from threading import Thread
from threading import Event

class MyThread(Thread):
    def __init__(self, event):
        super(MyThread, self).__init__()
        self.event = event

    def run(self):
        i=0
        while True:
            i+=1
            print ('Child thread running...',i)
            sleep(0.5)
            if self.event.is_set():
                break
            print()
        print('Child Thread Interrupted')

event = Event()
thread1 = MyThread(event)
```



```
thread1.start()

sleep(3)
print('Main thread stopping child thread')
event.set()
thread1.join()
```

Quando você executa este código, ele produzirá a seguinte **saída** -

```
Child thread running... 1
Child thread running... 2
Child thread running... 3
Child thread running... 4
Child thread running... 5
Child thread running... 6
Main thread stopping child thread
Child Thread Interrupted
```