



A biblioteca multiprocessing

A biblioteca multiprocessing

A biblioteca multiprocessing é uma biblioteca para suporte de criação de processos simular a threading.

Com a biblioteca multiprocessing é possível criar processos simultâneos no ambiente local ou remoto.

A biblioteca trabalha com a criação de subprocessos e não threads. Por conta disso, essa é uma biblioteca interessante para os programadores que possuem dois ou mais processadores, porque eles serão, de fato, aproveitados.

O pool

A biblioteca multiprocessing conta com um recurso chamado *pool* que permite que as várias entradas de dados sejam processadas de forma simultânea, através da criação e divisão do conjunto de dados para diferentes processos.

Sendo assim, essa biblioteca viabiliza o paralelismo de dados.

A biblioteca multiprocessing

A utilização do *pool* pode ser feita de uma forma bastante simples:

```
import multiprocessing mp
from multiprocessing import Pool

def f(x):
    return x*x

if __name__ == '__main__':
    with Pool(5) as p:
        print(p.map(f, [1, 2, 3]))
```

A biblioteca multiprocessing

A classe *Process* pode ser utilizada pelo programador para a criação de processos de forma manual e depois iniciá-los.

```
def f(name):  
    print('hello', name)  
  
if __name__ == '__main__':  
    p = Process(target=f, args=('bob',))  
    p.start()  
    p.join()
```

A biblioteca multiprocessing

No exemplo a seguir são criados dois processos para serem rodados em paralelo e um terceiro que será executado depois que os dois primeiros forem encerrados.

```
import os
from multiprocessing import Pool

processos = ('processo1.py', 'processo2.py')
outrosProcessos = ('processo3.py',)

def executaProcesso(processo):
    os.system('python {}'.format(processo))

pool = Pool(processes=3)
pool.map(executaProcesso, processos)
pool.map(executaProcesso, outrosProcessos)
```



Obrigada!

Ana Laurentino