Python_v_Multiprocessing_(Day 31)_[DATA MINDS]

October 5, 2023

MULTIPROCESSING - With the help of multiprocessing we can easily aquire the resources of the system and utilise them in a best possible way

[1]:
```
import multiprocessing
def test():
    print("this is my multiprocessing program")
if __name__== '__main__':
    m=multiprocessing.Process(target=test)
    print("This is main program")
    m.start()
    m.join()
```

```
This is main program
this is my multiprocessing program
```

[2]:
```
test()
```

```
this is my multiprocessing program
```

[3]:
```
def square(n):
    return n**2
if __name__=='__main__':
    with multiprocessing.Pool(processes=4) as pool:
        out=pool.map(square,[1,2,3,4,5,6,7,8,9])
        print(out)
```

```
[1, 4, 9, 16, 25, 36, 49, 64, 81]
```

[ ]:
```
def producer(q):
    for i in range(10):
        q.put(i)

def consume(q):
    while True:
        item=q.get()
        if item is None:
            break
        print(item)
```

```
if __name__=='__main__' :
    queue=multiprocessing.Queue()
    m1=multiprocessing.Process(target=producer,args=(queue,))
    m2=multiprocessing.Process(target=consume,args=(queue,))
    m1.start()
    m2.start()
    queue.put("Virat Tiwari")
    m1.join()
    m2.join()
```

```
0
1
2
3
4
5
6
7
8
9
Virat Tiwari
```

[ ]:
```
import multiprocessing
def square(index,value):
    value[index]=value[index]**2
if __name__=='__main__':
    arr=multiprocessing.Array("i",[2,3,4,5,6,7,8])
    process=[]
    for i in range(7):
        m=multiprocessing.Process(target=square,args=(i,arr))
        process.append(m)
        m.start()
    for m in process:
        m.join()
    print(list(arr))
```

[ ]:
```
import multiprocessing
def sender(conn,msg):
    for i in msg:
        conn.send(i)
    conn.close()

def receive(conn):
    while True:
        try:
            conn.recv()
        except Exception as e:
```

```
            print(e)
            break
        print(msg)
if __name__=="__main__":
    msg=["My name is Virat Tiwari","I am Data Science Student"]
    parent_con,child_con=multiprocessing.Pipe()
    m1=multiprocessing.Process(target=sender,args=(child_con,msg))
    m2=multiprocessing.Process(target=receive,args=(parent_con,))
    m1.start()
    m2.start()
    m1.join()
    child_con.close()
    m2.join()
    parent_con.close()
```

```
['My name is Virat Tiwari', 'I am Data Science Student']
['My name is Virat Tiwari', 'I am Data Science Student']
```

Thank You So Much !!

Yours Virat Tiwari :)