Multiprocessing

The multiprocessing library uses separate memory space, multiple CPU cores.

Consider below application which demonstrates the concept of Multiprocessing

```
import multiprocessing
import os

print("---- Marvellous Infosystems by Piyush Khairnar-----")

print("Demonstration of Multiprocessing")

def fun(number):
    print('parent process of fun:', os.getppid())
    print('process id of fun:', os.getpid())
    for i in range(number):
        print(i)


def gun(number):
    print('parent process of gun:', os.getppid())
    print('process id of gun:', os.getpid())
    for i in range(number):
        print(i)

if __name__ == "__main__":

    print("Total cores available : ",multiprocessing.cpu_count())

    print('parent process of main:', os.getppid())
    print('process id of main:', os.getpid())
    number = 3
    result = None

    p1 = multiprocessing.Process(target=fun, args=(number,))
    p2 = multiprocessing.Process(target=gun, args=(number,))

    p1.start()
    p2.start()

    p1.join()
    p2.join()
```

Output of above application

```
MacBook-Pro-de-MARVELLOUS:Today marvellous$ python multip.py
---- Marvellous Infosystems by Piyush Khairnar-----
Demonstration of Multiprocessing
('Total cores available : ', 4)
('parent process of main:', 1251)
('process id of main:', 2464)
('parent process of fun:', 2464)
('process id of fun:', 2466)
0
1
2
('parent process of gun:', 2464)
('process id of gun:', 2467)
0
1
2
MacBook-Pro-de-MARVELLOUS:Today marvellous$
```