Multiprocessing

What is multiprocessing?

Multiprocessing refers to the ability of a system to support more than one processor at the same time. Applications in a multiprocessing system are broken to smaller routines that run independently. The operating system allocates these threads to the processors improving performance of the system.

Why multiprocessing?

Consider a computer system with a single processor. If it is assigned several processes at the same time, it will have to interrupt each task and switch briefly to another, to keep all of the processes going.
This situation is just like a chef working in a kitchen alone. He has to do several tasks like baking, stirring, kneading dough, etc.

Multiprocessing: Running more than one process on a single processor

parallel processing: running a process on more than one processor.

```
import multiprocessing
  def print_cube(num):
          print("Cube of a number is",num * num * num)
def print_square(num):
   print("squre of a number is",num * num)
if __name__ == "__main__":
    # creating processes
    p1 = multiprocessing.Process(target=print_square, args=(10, ))
    p2 = multiprocessing.Process(target=print_cube, args=(10, ))
    # starting process 1
    p1.start()
    # starting process 2
    p2.start()
    # wait until process 1 is finished
    p1.join()
    # wait until process 2 is finished
    p2.join()
    # both processes finished
    print("Done!")
```

Program2:

```
# importing the multiprocessing module
import multiprocessing
import os
def f1():
    print("p1_id: ",os.getpid())
def f2():
     print("p2_id: ",os.getpid())
if __name__ == "__main__":
    # printing main program process id
    print("main process id",os.getpid())
    # creating processes
    p1 = multiprocessing.Process(target=f1)
    p2 = multiprocessing.Process(target=f2)
    # starting processes
    p1.start()
    p2.start()
    # wait until processes are finished
    p1.join()
    p2.join()
    # both processes finished
    print("Both processes finished execution!")
    # check if processes are alive
    print("p1 status is alive?:",p1.is_alive())
    print("p2 status is alive?:",p2.is_alive())
```

References:

www.python.org

www.geeksforgeeks.org