

Example of Multiprocessing #2:

This program uses multiprocessing to calculate and determine prime numbers and then append those numbers to a list.

```
'''Prime Number finder'''
def is_prime(n):
    if n <=1:
        return False
    for i in range(2, n):
        if n % i == 0:
            return False
    return True
```

It uses a simple algorithm to determine if the given number is prime or not.

```
'''Get Prime Numbers'''
def get_primes(range_min, range_max):
    primes = []
    for n in range(range_min, range_max):
        if is_prime(n):
            primes.append(n)
    return primes
```

This function is used to find all of the prime numbers within a given range of numbers and appends them to a list.

```
'''Store Prime Numbers into Message Queue'''
def queue_primes(msgQueue, processNum):
    print("Child process ", mp.current_process().pid, " starting")

    myprimes = get_primes(int(processNum * max/numProcesses), int((processNum + 1)*max/numProcesses))

    for prime in myprimes:
        msgQueue.put("Child process " + str(processNum) + " with process id " +
                    str(mp.current_process().pid) + " calculated" + "\n" + str(prime))

    print("Child process ", mp.current_process().pid, " closing")
```

This is the driver function for this program. It creates a list using the get_primes function and then places each of those prime numbers into a queue which will be accessed by the specific process that is calling this function at the time.

```
#Create message queue for processes
messageQueue = mp.Queue()

#Create and start processes
for p in range(numProcesses):
    process = mp.Process(target=queue_primes, args=(messageQueue,p))
    processes.append(process)
    process.start()

#Join processes
for process in processes:
    process.join()

#print results
while not messageQueue.empty():
    print(messageQueue.get())
```

This is the main function for this program. It creates a queue using the multiprocessing module and then creates the processes that will be used to queue the primes. Once a process is created, it is appended to a list and then the process is started. Then we join each of the processes using the built in join() method.