

Chapter 1

multiprocessing 模块

1.1 multiprocessing 模块综述

The multiprocessing module provides a low-level interface to process- and thread-based parallelism. Its main components are as follows:

进程 一个当前进程的派生（forked）拷贝，创建了一个新的进程标识符，并且任务在操作系统中以一个独立的子进程运行。你可以启动并查询进程的状态并给它提供一个目标方法来运行。

池 包装了进程或线程。在一个方便的工作者线程池中共享一块工作并返回聚合的结果。

队列 一个先进先出（FIFO）的队列允许多个生产者和消费者。

Pipe A uni- or bidirectional communication channel between two processes.

Manager A high-level managed interface to share Python objects between processes.

ctypes 允许在进程派生（forked）后，在父子进程间共享原生数据类型（例如，整型数、浮点数和字节数）。

同步原语 锁和信号量在进程间同步控制流。

1.2 使用多进程和多线程来估算 π

1.2.1 Using numpy

The main reason that numpy is faster than pure Python when solving the same problem is that numpy is creating and manipulating the same object types at a very low level in contiguous blocks of RAM, rather than creating many higher-level Python objects that each require individual management and addressing.