Multiprocessing vs. Multithreading

Multiprocessing

Pros

- Separate memory space
- Code is usually straightforward
- Takes advantage of multiple CPUs & cores
- Avoids GIL limitations of CPython
- Child processes are interruptible/killable
- **A must with CPython for CPU-bound processing**

Cons

- Inter Process Communication (IPC) a little more complicated with more overhead
- Larger memory consumption

Multiprocessing vs. Multithreading

Multithreading

Pros

- Lightweight , low memory consumption
- Shared memory, makes access to state from another context easier
- Allows you to easily make responsive UIs
- **Great option for I/O - bound applications**

Cons

- CPython, subject to the GIL
- Threads are not interruptible/killable
- If not following a command queue/message model (using the Queue module), then manual use of synchronization becomes a necessity
- Code is usually harder to understand and to get right due to the potential of **race conditions** increases dramatically