

Processes & Threads

Computing

What each one owns, and why the difference decides how you parallelise.

Process

- Its own memory space
- Crash affects only itself
- Expensive to create and switch
- Communicates via IPC, sockets, pipes
- True parallelism across cores

Thread

- Shares memory with its process
- One crash can take the whole process
- Cheap to create and switch
- Communicates through shared variables
- Needs locks to stay correct

THE DIFFERENCE THAT MATTERS

Shared memory makes threads fast and makes race conditions possible. Processes trade that speed for isolation.

Python's GIL means threads do not give CPU parallelism there — use multiprocessing for CPU work, threads for I/O waits.