

Multi-Tasking is Easy

Multi-tasking is easy.

Multi-tasking has been conflated with the harder problems of *time-sharing* and *memory-management*.

Time-sharing is useful to only a small set of domains – sharing a computer between multiple users.

This used to be an important consideration when CPUs were expensive (and not well-understood) and when memory was expensive.

Conflating multi-tasking with time-sharing and memory-sharing has caused large amounts of accidental complexity.

When cost and time-sharing and memory-sharing are dropped from the equation, we are left with the bare essentials of multi-tasking – a number of distributed processor¹ nodes communicating with one another across wires.

I know of no popular programming languages that target the design of such simple distributed systems. Most common programming languages target the design of single nodes within such a simple network.

There are some languages that target this domain, like *bash*, FBP², and coordination languages, but they are not very popular. *Bash* is relatively popular, but it is not recognized as a bona-fide programming language.

Concurrency is a design paradigm. *Parallelism* is a design issue and should not be conflated with *concurrency*.³ A parallel solution requires the use of the concurrent paradigm, but, the concurrent paradigm does not imply parallelism. *Time-sharing* has been conflated with *parallelism* which has been conflated with *concurrency*. This conflation has caused much confusion and has caused much accidental complexity. *Parallelism* has, further, been conflated with *memory-sharing*. A further complication is the use of a text-only syntax to describe *time-shared*, *parallel* solutions. This text-only fixation, has led to complicated concepts like *futures*, *promises*, *exceptions*, *etc.*, all of which can be more simply described in a concurrent paradigm.

1 Not called CPUs, since none of them is Central.

2 Flow-Based Programming <https://jpaulm.github.io/fbp/index.html>

3 See Rob Pike's talk "Concurrency is not Parallelism" <https://vimeo.com/49718712>