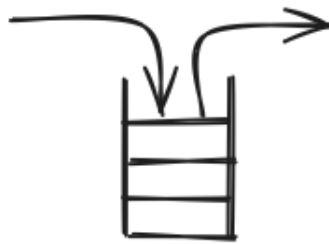


Local and Distributed

Stacks and Queues

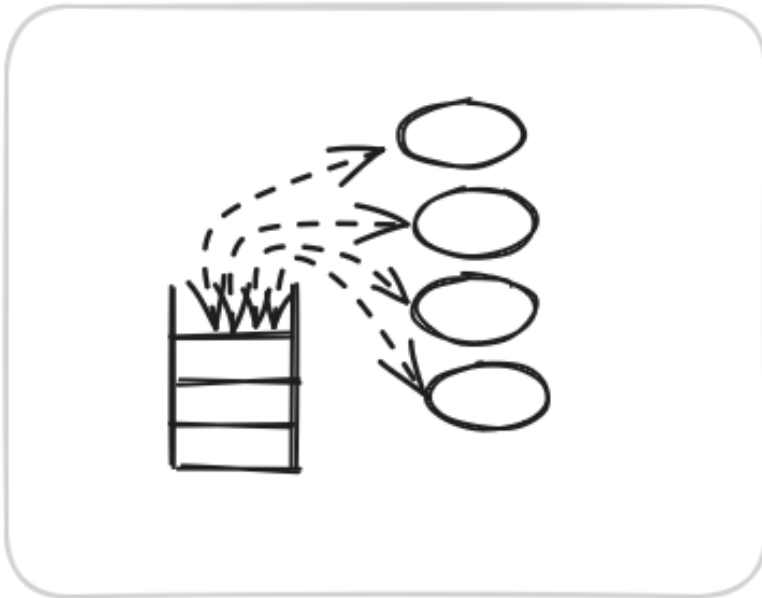
Stacks vs. Queues

Stacks are also known as "Lists".



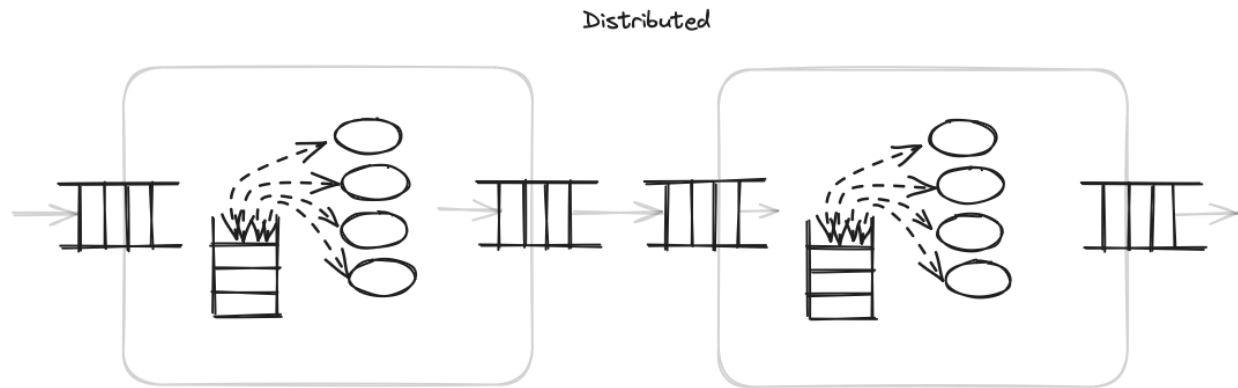
Stack-based languages support single-threaded, local only subroutines.

Local Only



It is possible to graft multitasking (also known as time-sharing with shared memory) onto stack-based notations, but, it is not convenient and requires extra work and extra software.

Multi-threading, as it was originally intended, involves connecting multiple single-threaded “Actors” together, via queues.



In 1950, it was too expensive to use more than one CPU “Actor” per computer, so we spent time and effort to time-share CPUs.

These efforts, also, included the use of shared memory.

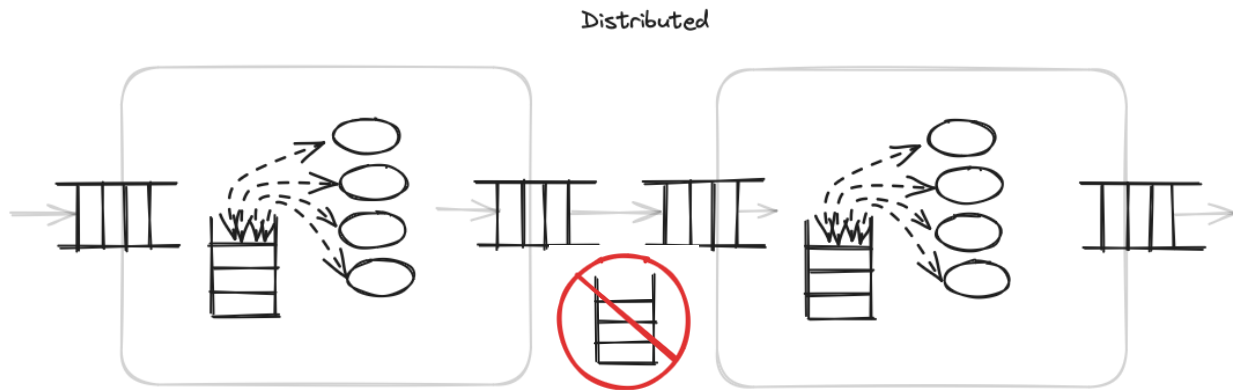
This combination of single-CPU time-sharing and shared memory led to numerous *gotchas*, for example, the Mars Pathfinder disaster, semaphores, etc., and, more recently the concept of *callbacks*, *promises*, etc.

In 2024, it is inexpensive to use multiple physical CPUs. Memory is quite inexpensive, so the concept of sharing memory is no longer needed, except in certain situations.

Shared memory is the default in 1950s-style programming languages, but, shared memory is no longer needed as the default in programming languages.

In spite of this new reality, most popular programming languages, like Python, Haskell, Rust, etc. continue to use 1950s-style defaults (CPU time-sharing and shared memory and heaps).

Note that, in internet programming and other forms of distributed programming, like robotics, blockchain, etc., it is not practical to share stacks across networks.



Stack-based languages and notations are not convenient for expressing distributed programs.

Appendix - See Also

References

<https://guitarvydas.github.io/2024/01/06/References.html>

Blogs

<https://guitarvydas.github.io/>

<https://publish.obsidian.md/programmingsimplicity> (see blogs that begin with a date 202x-xx-xx-)

Videos

<https://www.youtube.com/@programmingsimplicity2980>

Books

leanpub'ed (disclaimer: leanpub encourages publishing books before they are finalized - these books are WIPs)

<https://leanpub.com/u/paul-tarvydas>

Discord

<https://discord.gg/Jjx62ypR>

all welcome, I invite more discussion of these topics, esp. regarding Drawware and 0D

Twitter

@paul_tarvydas

Mastodon

(tbd, advice needed re. most appropriate server(s))