Logical clocks

- Logical clocks define a partial order of events in a distributed system

- They capture the causation (or lack thereof) between two events

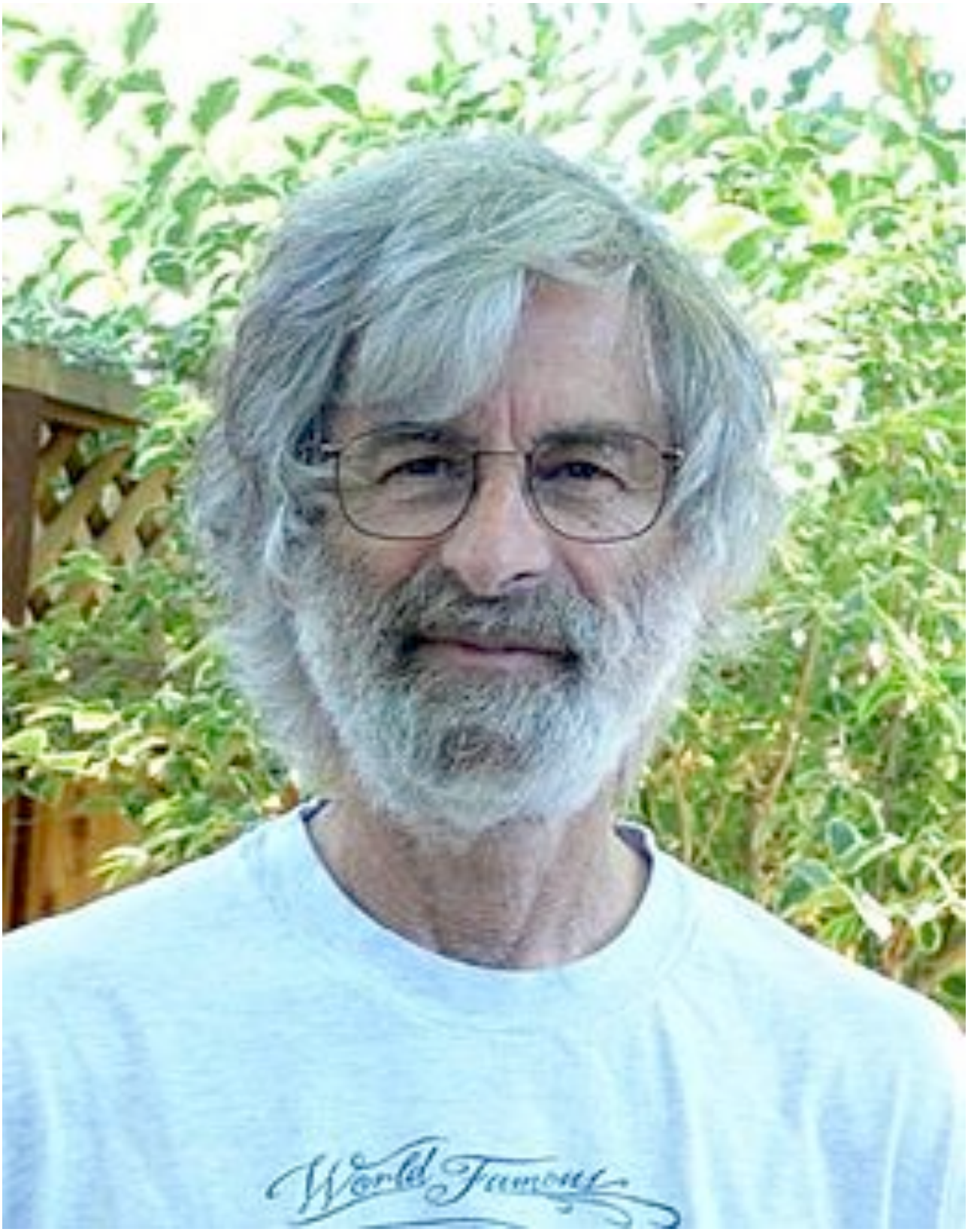
- They don’t capture physical time

- They exist in many forms (go check em’ out!)

- Lamport timestamps

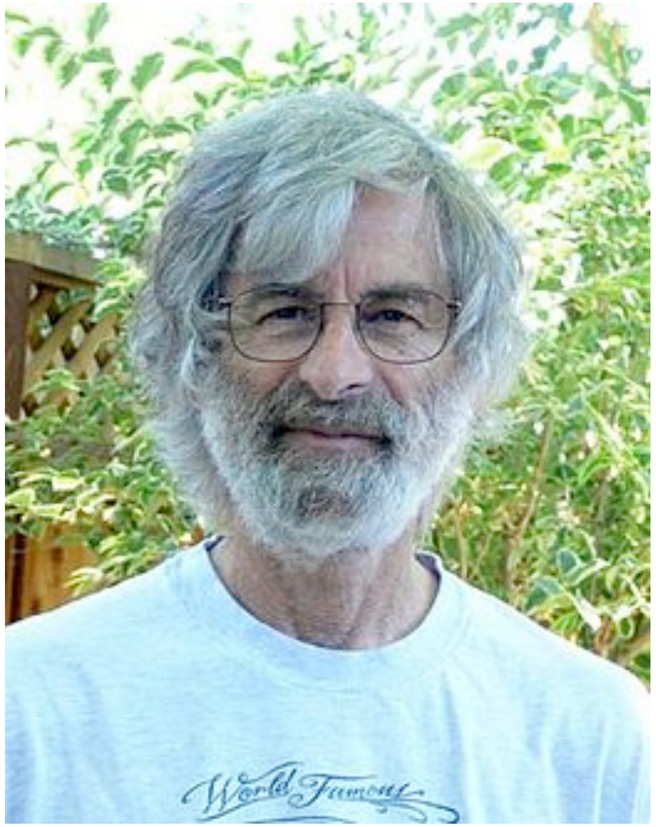
- Vector clocks

- Hybrid logical clocks
- …


World Famous

Leslie Lamport, inventor of the *happened-before* relation

Logical clocks



Leslie Lamport, inventor of the *happened-before* relation

- Logical clocks define a partial order of events in a distributed system
- They capture the causation (or lack thereof) between two events
- They don’t capture physical time
- They exist in many forms (go check em’ out!)
  - Lamport timestamps
  - Vector clocks
  - Hybrid logical clocks
  - …

Alright,

we have partial ordering, now what ?

Let’s say you have read everything about logical clocks, and implemented one for your system.

You have two routes to go from there:

- Add an arbitrary component to form a total order of events, then implement LWW
- Embrace the partially-ordered nature of distributed system, and implement *multi-value registers*