SYSTEMS ENGINEERING · WEEK 14 · STATION VI

What is a System?

The three sentences, earned. The definition. The field.

THE SENTENCES, EARNED

Isolation is the semantic problem. Virtualization buys performance with self-reference. What a kernel cannot decide, it bounds.

ISOLATION

Weeks 2 to 4 and 7 to 9. Give each program the illusion of the whole machine: the door, translated addresses, a context of its own. A process is a meaning the kernel constructs, and threads are what happens when the construction is shared.

SELF-REFERENCE

Weeks 3 and 6. An OS by emulation equals one by virtualization; the difference is a factor of twelve and the kernel needing what it provides. Gödel II as an operating system; the TCB as its axioms.

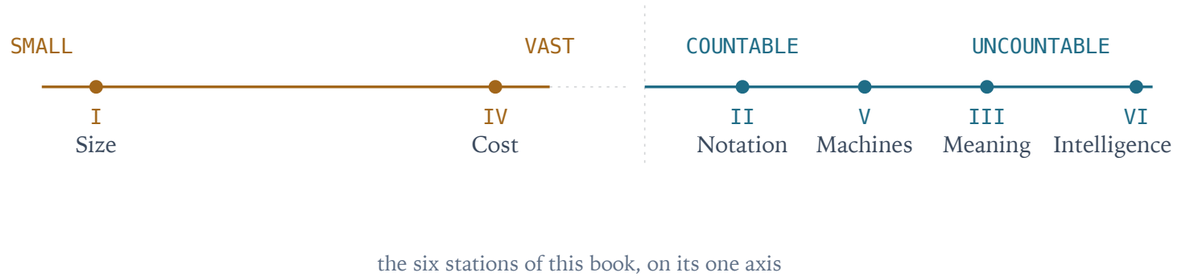
BOUNDS

Weeks 5, 9 to 12. The timer for halting, reachability for liveness, cycle detection for deadlock, the model checker's k for Rice, the profile for the budget. Every mechanism a chosen approximation, named.

A system, then: a construction of isolated meanings on one universal machine, held together by a small trusted ground and a set of bounds that stand in for what no program can decide.

THE AXIS

Where a systems engineer stands.



the six stations of this book, on its one axis

the six stations; this class dwelt in III and worked in IV

Station III is home: the universal machine, the theorems about what it cannot know about itself, and the kernel as their run-time consequence. Station IV is the workshop: the model checker for the properties, the profiler for the budget, physics for the floor.

The compiler class stands one station earlier, where notation becomes meaning. The introduction class walks the whole line. The three classes are one argument: proof against truth, and the gap between, which is where engineering lives.

INTELLIGENCE

Developing new languages, or new properties in existing ones, by discovering and understanding promising unproven truth.

The book's definition. Every notation was invented by someone who saw a truth before it could be proved: Turing's machine before there were computers, the process before there were kernels, the page before there was hardware to walk it.

Systems engineering, against the definition: the properties are isolation, liveness, progress, safety, each a new property in the language of the machine, each seen as true of good systems before anyone could prove it of one. The proofs came later, bounded, and the field advances by inventing the next property to want.

AND THE MACHINES

A generator produces notation in existing languages. It does not discover unproven truth, because truth is a property of meaning and it has none. That is not a limitation of this year's model; it is the definition. The person who states the next property, and checks it, is doing the thing.

BEYOND THE CLASS

What to read, what to build, what to measure.

READ

Anderson and Dahlin for the kernels that are not selfie. The seL4 papers for a proof relative to a TCB. Herlihy and Shavit for the interleavings this class only counted. The book's Recommended Readings, all ten.

BUILD

A hypster in real supervisor mode on real RISC-V. A collector that does not stop the world. A model of two contexts in rotor, which nobody has done yet, and which would let week 10 see week 8.

MEASURE

Everything, and change the metric before it decays. The profile is the start of the argument. Twenty watts is the standing challenge.

Thank you. Keep your selfie; it is the machine, the kernel, and the proof of both, in twelve thousand lines you have read.