

---

We have pondered the prospect of a Nock issue for some time. These articles represent a *summa computralis* in a sense, an account of decades of work in making a fundamental computational engine into a practical reality on which many people program every day. Nock is something less than a programming language, and something more than a bare Turing machine. It encodes structure through nouns, and telos through its kelvin decrement system. “Perfect” is the enemy of “good enough”, as they say, and an agreed-upon standard better than an eternal search. Nock sits at the nexus of an uneasy tension between minimalism and pragmatism, yet it sits with grace.

As Urbit proceeds and Nockchain grows, it is incumbent to preserve the technical and historical record of Nock computing and its intent. Some of the material in this issue is syndicated for posterity, like ~timluc-miptev’s excellent Nock tutorial and ~barpub-tarber’s description of Ax. Other articles represent the first exposition in print of major components of practical Nock evaluation, such as the history of Nock as a family of combinator calculi or Nock virtualization as a design target; we also highlight ~XXX’s exposition of PLAN. (Besides Ax/-Conk and PLAN, we note as well tibru and SKEW, omitted from this collection.) We are particularly pleased to publish the description of NockPU, the first bare-metal Nock interpreter, by ~mopfel-winrux.

USTJ also expands herewith its artistic portfolio. ~norwyl-morlec, who produced iconic ASCII art splash pages for the preceding issues, returns. We are further privileged to feature new art by longtime Urbit-associated artists ~dalhec-banler, ~litmus-ritten, ~tocrex-holpen, and ~natnex-ronret.

We gratefully acknowledge our supporters, not only our readers and authors but the institutions that have made USTJ possible, namely the Urbit Foundation and West Martian Limited. We particularly call out the labors of the Urbit core development team from Tlon Corporation’s foundation in 2013 to the present day, and the many userspace application developers who built on shifting sands gradually settling. This volume is dedicated to your labors.

Computer science happens in the trenches. ☒