

The SOLO operating system: a Concurrent PASCAL program.

Software—Practice and Experience 6, 2 (April-June 1976), 141-149.

This overview of the function and implementation experience of SOLO, a "one-user time-sharing system" for the Digital Equipment PDP-11/45, shows the power of simplification. Just as designers of complex systems have been dismayed at the incredible growth of complexity as interacting functions pile atop one another, the author delights in the simplicity and elegance that come with spartan frugality of function. The system uses PASCAL as its command language while providing an interactive program development environment somewhat more spartan than, say, UNIX. No debugging system is described (I suppose that PASCAL programmers are expected to write only correct programs) and the addition of breakpoint debugging might do violence to the simple design.

The interest in this system is not in its function, but in its implementation, which is largely in PASCAL and is very organized. Thus it should be worthy of study. The author would convince us that the existence of SOLO makes it "... now realistic and attractive to replace a huge ineffective 'general-purpose' operating system with a range of small efficient systems for special purposes." Although there is certainly an element of truth in that statement, there is also a sweeping overgeneralization: one can equally well develop an argument that even a narrowly dedicated real-world system requires a surprisingly wide selection of the functions supplied by those "ineffective 'general-purpose' operating systems," and the challenge is to discover how to provide those functions in a simple way, rather than to omit them. One wonders, for example, how the concepts of SOLO would fare in implementing a near-future office automation system that dedicated a personal computer to each office worker, but whose functional specifications involved inter-computer communication of messages, an office-wide filing system, high availability, intense display support, graceful recovery from hardware errors, and other such amenities that a manufacturer might consider important in a product. The evidence of SOLO provided by this paper is not convincing.