

## *Greetings*

Michael D. O'Dell Editor-in-Chief

*H*ello again.

In the days of yore and magic, a name was a very powerful thing. It was a word to conjure with in the truest sense — knowing someone's true name made her or him vulnerable to all manners of spells and other mischief. Modern distributed computing systems aren't much different in some sense — names are now objects in their own right, but are still near-magical and endowed with special abilities. The design of naming systems has come into its own as a topic of study, and the first paper in this issue, "An Experimental Implementation of the Tilde Naming System" by Comer, Droms, and Murtagh, is in this area. The paper presents a naming system primarily designed for distributed filesystem collections and then goes on to describe a prototype implementation used to evaluate the design in actual use. The results are intriguing and provide much food for thought.

Our next paper, "A Comparison of Basic CPU Scheduling Algorithms for Multiprocessor Unix" by Curran and Stumm, examines via simulations some of the performance and complexity trade-offs of several scheduling algorithms for multiprocessor Unix systems. Multiprocessors have been around for a long time, and one might think that this is well-trodden ground. The results of this study, however, may well surprise your intuition, and further demonstrates the importance of system optimization over component optimization.

In our third paper, Dewan and Vasilik present "An Object Model for Conventional Operating Systems". They give us an object model which is used to extend conventional Unix-like operating systems to support persistent, shared objects accessible

from multiple programs. "Objects in the large" is a particularly interesting problem because many real systems don't run on just one machine, much less in one address space, so coping with system structure beyond the "program" level would seem to be critical to the success of "object oriented design". The system described in this paper is a currently in use supporting research in distributed applications.

This issue finishes volume three, and as is customary, I call your attention to the inside front cover and the members of the Editorial Board for this volume. It is these brave souls who bear the brunt of the critical task of reviewing submissions. The work is sometimes inconvenient because we strive for timeliness, but they all give generously of their time and energies to help the authors make our journal what it is. My deepest thanks to all of them. And as is traditional now, some will be departing and some new names will be appearing next volume.

In addition to the Editorial Board, others have graciously read particular papers when asked. In particular, special thanks to Gene Spafford for guest editing the multiprocessor issue and all the folks he drafted for that effort (see 3.1, p. 9). A very special thanks also to the following for helping with their particular expertise when it was needed: Ray Essick, Mike Karels, Sam Leffler, Marc Pucci, John Puttress, and Paul Rabin.

My last duty this month is a very sad one. Tom Strong, our resident typesetter, whose handiwork you have been reading for years in all kinds of Association publications, but especially this one. died earlier this year. His dedication to quality and craftsmanship has been deep and unflinching; his countenance and his contribution to *Computing Systems* will be sorely missed.

So long, Tom.