

Greetings

Michael D. O'Dell Editor-in-Chief

*W*elcome to the second issue of *Computing Systems*. First of all, let me thank everyone for the warm reception extended to the first issue, and also say that your thoughts on this journal are most heartily solicited.

A sage member of the USENIX Board once observed, "Every meeting is an experiment in running a meeting." So too with publishing this journal. We have two important firsts in this issue: our first contribution by a European author, and the addition of a new section. We hope the European offering is but the first of a continuing line of international contributions. The new section, on the other hand, is considerably more experimental.

A New Section

This issue presents the first publication of a new section of *Computing Systems* called "Controversy." We view this section much like the "op-ed" page in a newspaper such as *The New York Times*. Opinions on controversial topics will be printed, possibly along with ensuing dialog. Submissions meant for "Controversy" should be marked as such, and should contain succinct, well-made arguments about issues of interest on the cutting edge, one way or another. Responses, equally well-thought-out, are welcome, but blind religious passion is inappropriate. Decisions regarding publication of "Controversy" items lie solely with the Editorial Staff. While strong convictions breed strong rhetoric, we hope that "Controversy" will provide illumination, not conflagration.

In This Issue

The papers in this issue have a decided UNIX bent, but all deal with how UNIX can and must cope with the evolution of hardware and software technologies.

Wedding multiprocessors and UNIX is a topic currently making the rounds, with various authors claiming that adapting to multiprocessors requires serious extension or modification to some central UNIX notions. *Enhanced Resource Sharing in UNIX*, by J. M. Barton and J. C. Wagner, shows how powerful resource sharing can be added to UNIX process semantics without up-ending the existing, well-understood model.

Exploiting potential concurrency when rebuilding software is the concern of *Design and Implementation of Parallel Make*, by E. H. Baalbergen. This paper shows that merely “doing several things at once” is not all there is to the problem.

Watchdogs – Extending the UNIX Filesystem, by B. N. Bershad and C. B. Pinkerton, describes a mechanism for embedding an agent process in the filesystem, allowing it to pose as a file or directory, with the agent providing the object semantics. This provides an extremely general way to extend the filesystem for all kinds of interesting purposes.

In the paper *Yacc Meets C++*, by S. C. Johnson, the creator of Yacc describes his work in extending Yacc to handle C++. This was accomplished by integrating the attributes from attribute grammars with the abstract type capabilities of the C++ programming language. The result is an experimental tool *y++*.

Finally, our first contribution to “Controversy,” *Can UNIX Survive Secret Source Code?*, by M. Lesk, raises a question with wide-ranging implications. Lesk’s experience and perspective spans the entire history of UNIX, and his viewpoint is quite thought-provoking, no matter where one eventually comes down on the issue. The recent formation of the Open Software Foundation makes Lesk’s commentary even more timely.

Contributions

I will close my comments this month by thanking the contributors for their efforts, and by admonishing you, the reader, to become a contributor. The Editorial Staff does not write the papers which appear here – we can only publish what we receive. Therefore, it is of paramount importance that you share your hard-won experience by writing about it. This is not a test – you will not lose points for grammar and spelling; those things can be corrected by our most able Managing Editor. Everyone will lose, however, if you possess some understanding and fail to share it.

Thanks for coming, and I'll see you again next quarter.