Greetings

Michael D. O'Dell Editor-in-Chief

*W*ell, We Made It!

Hello again, and welcome to the final issue of the first volume of *Computing Systems*. Seems like we started this whole thing well over a year ago. Oh well, *tempus fugit*,[1] I guess.

Contents

The papers this issue are, as usual, a quite eclectic collection. In no particular order, we have "An Unorthodox Approach to Undergraduate Software Engineering Instruction" by Robert A. Morris of the University of Massachusetts Boston Campus which discusses an approach he used teaching software engineering. While it might first appear that education isn't particularly covered by our portfolio, I would claim that future advanced computing systems will spring from the well-trained minds of today's students, so we have more than a passing vested interest in the problem of training minds to understand software.

One of the most exciting parts of computer science is getting to "be there" at the creation of new things. The evolution of the C++ programming language has been a surprisingly public gestation, and Bjarne Stroustrup continues the tradition with the paper "Type-safe Linkage for C++." This paper is of particular interest because the author chose to solve a difficult problem in a manner which didn't simply wish the problem away by telling users to abandon their current environment for something which is conceptually "The Right Thing." Ripping-up the existing world can

1. I've always wanted an opportunity to use that phrase!

certainly be attractive, but in the end, solutions arrived at via a more pragmatic route may well be more widely applicable than ones created by the convenience of intellectual fiat.

And finally, we have a rather lengthy paper about a system which excited me the first time I heard about it, and which has continued to excite me ever since. Papers describing new operating systems don't come along very often because seriously building a new operating system is a dramatic undertaking. We are very pleased to have the major reference paper describing the CHORUS Operating System appear in *Computing Systems*. CHORUS is a new distributed operating system which provides the UNIX system call interface for the convenience of existing software, but then adds a generous helping of originality for programmers building more ambitious systems. Particularly interesting is the fact that CHORUS has been completely rewritten three or four times (depending on how you count) critically seeking a balance between theoretical perfection and the dual goals of performance and the semantic accuracy of its UNIX interface. CHORUS is written largely in C++, is completely distributed, runs on multiprocessors, has realtime capabilities, and is running on several rather different processor families.

Reflections

It has certainly been an interesting year, both for computing in general and for *Computing Systems* in particular. One might say it was a year of lost innocence. Regardless of how dire you personally think the event really was, The Great Internet Worm Incident has already been deemed a "coming of age" for networks, system security, and software reliability. The world of computing is now rather different for it having happened. How it has changed, and whether it is better will take some time to understand.

The *Computing Systems* staff and Editorial Board certainly learned what it takes to turn things around in a lot less than leisurely fashion (although I do think Peter Salus secretly knew all along what it would take and simply didn't tell anyone for fear they would be scared off). At any rate, we made it through the

first year, and fully intend to be back with the next issue, Volume 2, Number 1.

Thanks

I want to especially thank all the people that made Volume 1 possible. First and foremost there are the authors, for without them there would be nothing to publish. Without exception, all those who submitted manuscripts were quite gracious when the news from a reviewer was less than good, and were prompt and attentive to the reviewers' comments when rewrites were requested. Many, many thanks to all of you.

The Editorial Board is the backbone of the effort. They willingly volunteer their precious time to make the journal what it is. Without them, it wouldn't be a journal. My special thanks to each and every one of you. In addition to the people listed on the masthead, several others have reviewed manuscripts on occasion and I would like to thank them here: Rick Adams, Hal Feinstein, Ed Gould, Alicia Gronke, Andrew Hume, Mark Linton, John Weidner, and Dave Yost. The make-up of the Editorial Advisory Board will change with the advent of Volume 2: first of all, we don't want readers to go stale; secondly, we thought that by asking about a third of the Board to step down this year and another third next year, we could institute three year terms.

The USENIX Production Staff generates the camera-ready images which the crew at the University of California Press makes into a real, physical object. The heroic efforts and quality work by all concerned in the face of some rather challenging copy are greatly appreciated. Many thanks.

Finally, I would like to thank Peter Salus, our august Managing Editor, for making it all happen. My world is quite hectic these days, and without Peter to keep track of all the details, to remind me when I haven't done something I should have remembered to do, and to do all the other important tasks which generally amount to juggling seven sharp objects while riding a pogo stick, it just wouldn't happen. Without Peter, the journal couldn't get in the door, much less get out the door. Peter, my deepest thanks for a job done with style and flair.

And to you, Gentle Reader, thanks for coming. "We do it all for you" is almost certainly someone's service mark, but in this case, it is in fact quite true.

All the best to you and yours for the coming year. As for me, I'll see you next Volume. (I like the sound of that!)

P.S. Start the year right. Submit a manuscript!