

Greetings

Michael D. O'Dell Editor-in-Chief

*T*his issue begins with a paper addressing a problem which ought to be of wide interest: the rapid searching of large text collections. Various techniques for text inversion have been in use for years, but many schemes suffer from a common malady—the index-inversion techniques are quite slow and expensive for a non-trivial amount of text. Alistair Moffat's paper "Economical Inversion of Large Text Files," presents some interesting new work on high-speed indexing of text without sacrificing retrieval performance. Such work may make real text retrieval applications which are beyond the scope of UNIX *grep* more commonplace.

When building systems, one of the greatest challenges is deciding how much effort to spend on which parts of the system. The paper "Efficient Demultiplexing of Incoming TCP Packets" by McKenney and Dove presents an analysis of such a tradeoff. The Berkeley networking code has evolved over many years with improved algorithms to address new challenges presented by the expanding Internet environment, usually at the cost of increased complexity. McKenney and Dove present another challenge, but rather than simply jumping in and doing an "obvious hack," they present a carefully constructed analysis of the failings of the existing algorithms and of several alternatives which they considered prior to adopting the most complex change. The final modification presented indeed addresses their original performance concern, but it is the analysis which drives their decision process past several simpler alternatives which is most valuable to contemplate.

Finally, we introduce a new section to *Computing Systems: Experiences*. In some sense, this moniker is redundant, as our general efforts focus on reconciling experience with theory, but let me make the case another way. Those of you who have been building systems for years have amassed a large corpus of knowledge about how one goes

about building working systems. This covers everything from intuition about DMA versus interrupt-driven data transfers to software organization, and to some large extent this is completely uncaptured in the archival literature. It is seldom taught in operating systems courses these days and in times past, when such presentations were commonplace at meetings like the USENIX conferences, there was no proceedings volume generated. An analogy to consider from chemistry is “lab technique.” Building systems, like chemistry, is a laboratory science—you plug in boards and wire things together just like you assemble apparatus and handle reagents. If you don’t know how to deal with the apparatus or safely handle the reagents, you won’t do much chemistry. Likewise, if one doesn’t possess the large corpus of intuition about all the different ways one can approach an interfacing problem, then it will be frustrating and much time will be spent rewalking trodden ground. The deep problem is that without an archival literature, one can only earn one’s stripes (scars, actually) the hard way.

Computing Systems has decided to take up the challenge of developing an archival literature capturing Experiences. As a first step down this path, we offer Atkins, et al., “Overcoming Data-Transfer Bottlenecks . . . ,” in which they discuss the challenges of interfacing computers with fundamentally very different views of communication.

Let me close by offering a challenge to the system builders out there: This is a great opportunity to help capture the information we think of as “lore of the guild” and “oral tradition.” Contribute something so that newcomers aren’t forced to plow the same ground again and again.