Guest Editorial

Peter Honeyman University of Michigan

File systems, a fundamental component of operating systems, receive a great deal of attention in conferences and journals devoted to operating systems research and practice. The field itself is broad, with issues related to name spaces, mass storage, distributed systems, parallel architectures, replication—the list is long.

In the Spring of 1992, the USENIX Association sponsored a workshop on file systems. Included in this issue of *Computing Systems* are three noteworthy papers from that workshop. These papers provide a snapshot of current research in the design and measurement of file systems for contemporary operating systems. The first, "A Comparison of Three Distributed File System Architectures: Vnode, Sprite, and Plan 9," by Brent Welch, examines architectural distinctions among three influential distributed file system architectures. Sprite and Plan 9 go to lengths to separate the naming and access aspects of file systems; Welch offers a valuable insider's view of why such an architecture is important, and where it succeeds.

Jeffrey C. Mogul's paper, "Recovery in Spritely NFS," continues his earlier work on making the NFS protocol safe for humanity. Here, he describes the recovery mechanisms necessary to support aggressive client caching in the face of failure. Mogul combines recent work on recovery in client/server systems, in which recovery is initiated and managed by the server, with techniques for maintaining correct behavior in the presence of network partitions or lengthy delays.

The third paper, "Optimal Write Batch Size in Log-Structured File Systems," by Scott Carson and Sanjeev Setia, extends their earlier work in the analysis of performance implications of policies for managing write operations in file systems. Their work has a certain beauty to it, in that they are able to describe intricate performance issues by relating them solely to hardware characteristics of the implementation platform. This leads to surprisingly simple conclusions from a rather detailed analysis.

I hope you enjoy reading and learning from these papers. I thank the staff of *Computing Systems*, especially Peter Salus, for facilitating the production of this special issue and owe a great debt to the workshop program committee, Mike Kazar, Larry McVoy, Mendel Rosenblum, and Liuba Shrira, for their able assistance in reviewing and selecting papers, both for the workshop and for this publication.