

Foreword

Given life by Turing Award winning Bell Labs computer scientist Ken Thompson at Murray Hill, N.J., in August 1969, UNIX spent its early years as a research curiosity. When I met up with Unix in the summer of '82, however, it already possessed the one characteristic that destined it to dominate a major chunk of the world's market for operating systems—portability. UNIX kicked off the era of open systems, the first wholesale paradigm shift in the history of computing, by being the first portable operating system.

Portability is so crucial because it symbolizes everything that open systems is about, and is the critical computing ingredient for the Information Age. You may hear people use the word primarily to talk about their applications that can run on more than one type of computer platform, but, at its highest level of abstraction, portability is much more. When you think about using standard network interfaces to pass data between different computers, that's portability of information; running applications across a range of devices from desktop to mainframe—or even supercomputer—is portability across scale; and the ability to swap out old technology for the latest technical advances without dramatically affecting the rest of your installation is portability through time. All this is necessary to support the extremely sophisticated levels of information malieability that corporations need to make the Information Age really work.

UNIX was always technically cool, advanced, insanely great, etc. So cool that Bell Labs began giving it away to colleges and universities in 1975 because they thought it would be a good recruitment tool—they believed graduate computer engineers would want to work at the place that produced such an elegant piece of technology. But UNIX's all-important portability didn't come about until 1977. Before that, UNIX's technical qualities alone had lured many Bell operating company department heads to Murray Hill, where they learned about UNIX from its small team of creators and began deploying it on Digital Equipment Corporation computers throughout the Bell System. By 1977, AT&T found itself buying a much larger percentage of Digital's annual output than seemed comfortable. (AT&T didn't want to be responsible for a precipitous drop in Digital's fortunes if it had to stop buying for any reason.) So that year, UNIX's creators ported UNIX for the first time, to a non-Digital computer whose only significant characteristic was that it was a non-Digital computer.

After that, UNIX was portable, and entrepreneurs ported it to new microcomputers like crazy. That's when I came on the scene, as a computer industry news reporter covering all that entrepreneurial energy. Even in 1982, the manifest destiny felt by the people in the UNIX industry was clear. And the idea of a common operating system atop different hardware platforms so powerfully fired the imaginations of information systems managers in major corporations that, today, UNIX has become their de facto server operating system.

Given that you've purchased or are considering this book, you already know that UNIX is ubiquitous. What UNIX is not, however—even with the modern graphical user interfaces that paint a pretty face on it—is easy to program or administer compared to DOS or NetWare. Just as a 747 is a bit more complicated to run than, say, a glider, UNIX's increased flexibility and power come with the price of greater complexity.

This book, which delves deeply into the underpinnings of UNIX systems and offers detailed information on many different brands of UNIX , can be your first step on an enjoyable journey into the powerful, technically elegant world of open, portable computing.

Mike Azzara, associate publisher/editorial director, *Open Systems Today*.

