

Fifty Years of Operating Systems

SIGOPS SOSP History Day
October 4, 2015

The Story

Operating systems are a major enterprise within computing. They are hosted on over a billion devices connected to the Internet. They were a \$33 billion global market in 2014. The number of distinct new operating systems each decade is growing, from 9 introduced in the 1950s to an estimated 350 introduced in the 2010s.

Operating systems became the subject of productive research in late 1950s. In 1967 operating systems research leaders organized the SOSP (symposium on operating systems principles), starting a tradition of bi-annual SOSP conferences that has continued 50 years. The early identification of operating system principles crystallized support in 1971 for operating systems to become part of the computer science core curriculum.

In October 2015, as part of SOSP-25, we celebrated 50 years of OS history. Ten speakers and a panel discussed the evolution of major segments of OS, focusing on the key insights that were eventually refined into cherished OS principles (sigops.org/sosp/sosp15/history). This is the introduction to the video record.

--- Peter J. Denning, Organizer

The Speakers

Overview of the Day

After a short summary of how the SOSp series began in 1967, EmCee Jeanna Matthews introduces the speakers. She has photos of them in their younger days when they were inventing OS principles.



Jeanna Matthews

The Founding of OS conferences



Jack Dennis

Jack Dennis launched the SOSp series in 1967. He saw an opportunity to bring out the emerging principles of operating systems and communication networks.

The Speakers

OS Foundations

Peter Denning shows the evolution of OS's from batch systems and then to time-sharing, distributed, and mobile-cloud systems. A body of significant principles evolved over time, including two of major focus in his own research, location independent addressing and locality.



Peter Denning

Protection and Security



Butler Lampson

Butler Lampson traces a long history of protection mechanisms in spite of which security remains a major problem. He considers isolation, access control, access policy, information flow control, cryptography, trust, and assurance. In the end, people dislike the inconvenience security causes.

The Speakers

System Languages and Abstraction

Barbara Liskov examines the evolution of abstractions, such as processes and software layers, to organize complex systems. Some abstractions are separate service processes invoked by RPC, others are overlaid on a user's process by monitors. Many have found their way into system programming languages. Communication is a major issue.



Barbara Liskov

File and Memory Management



Mahadev Satyanarayanan
(Satya)

Satya highlights three major themes in the search for better memory systems. Scale sought larger and faster memories that kept up with ever faster processors. Transparency hid complex physical structures behind a simple address space. Fault tolerance made memories robust with partitioning and replication. Hierarchical file systems are deeply embedded into the Internet and will not soon disappear.

The Speakers

Fault Tolerance

Ken Birman examines fault tolerance, a system's resistance to failure of memory hardware, and consistency, the system's ability to correctly reconstruct lost data from multiple copies. Enormous progress with these issues enabled the modern cloud to scale reliably to huge sizes.



Ken Birman

Virtualization



Andrew Herbert

Andrew Herbert traces the history of the OS structuring principle of virtual machines. This principle appears in virtual machine monitors, desktop hosted virtual machines, layered abstract machines, and a standard environment for program execution in distributed Unix.

The Speakers

Hardware and Architecture

Dave Patterson shows that the old debates about RISC versus CISC architecture for processors and NUMA versus clusters for parallel processing are mostly settled. Moore's law is nearly spent. For continued improvements, we look now to new memory architectures, open source instruction sets, and custom chips: a new hardware-software partnership.



Dave Patterson

Parallel Computing and the OS



Frans Kaashoek

Frans Kaashoek traces the history of parallel programming in OS, starting with parallel I/O and CPUs, then distributed systems, and then systems with multicore processors. Because software must be parallel to exploit multicore processors, the OS community is going through a rebirth of research in parallel computing.

The Speakers

The Network and the OS

Dave Clark digs through his long experience in getting network protocols (notably TCP/IP) to work efficiently with the OS. It was a long hard slog to gain deep understanding of the efficiency of each little part of the protocol software. Eventually the protocols were successfully integrated and today's OS all include the network.



Dave Clark

The Rise of Cloud Computing Systems



Jeff Dean

Jeff Dean shows that older approaches to parallel processing such as transaction processing and high-performance computing emphasized performance but did not scale well. When fault tolerance was emphasized instead, parallel systems scaled to tens of thousands of processors and millions of users without loss of performance, realizing the old dream of a computing utility available cheaply to everyone.

The Speakers

Panel: Is Achieving Security a Hopeless Quest?

Despite all the work in OS to provide protection and improve security, cyber crime has grown into a major social issue. There seem to be no solutions to loss of data and theft of identity. Does the OS community bear a responsibility for this mess?



Margo Seltzer



Mark Miller



David Mazieres



YY Zhou