Inferno: la Commedia Interattiva

Sean Dorward
Rob Pike
Dave Presotto
Howard Trickey
Phil Winterbottom

Bell Labs, Lucent Technologies
Murray Hill, New Jersey 07974

As telecommunications, entertainment, and computing networks merge, a wide variety of services will be offered on a diverse array of hardware, software, and networks. Inferno provides a uniform execution environment for applications and services in this chaotic world. Inferno comprises a distributed, networked operating system; a virtual machine; a programming language; naming, access, and security protocols; and so on. Above these, it provides standard interfaces to access networks, graphical toolkits, and other system features. These standard interfaces, coupled with the architecture-independent virtual machine, an operating system that can run on bare hardware or as user processes above a commercial operating system, and the ability for Inferno programs to adapt themselves dynamically to the current environment, offer an unprecedented level of portability for applications and services.

Programs for Inferno are written in Limbo, a new concurrent language. Limbo programs are compiled into byte-codes representing instructions for a virtual machine called Dis. Dis is a simple three-operand memory-to-memory machine, supplemented by a few specialized operations for handling higher-level data types such as arrays, strings, processes, and communication channels. When loaded into memory for execution, the byte-codes may be interpreted or optionally compiled into instructions for the underlying hardware. The Dis instruction set was designed expressly for efficient compilation: many Dis operations translate into single instructions on an x86 processor.

The implementation of Dis incorporates a real-time garbage collector that manages memory below the level of the instruction set. Similarly, the multitasking and communication primitives hide all details of synchronization and scheduling.

Although the implementation is all new, the Inferno operating system draws heavily on the lessons of Plan 9. The system is built around two simple ideas: every resource in the system, local or remote, is represented by a hierarchical file system; and a user or process assembles a private view of the system by constructing a file name space that connects these resources. Because the process model incorporates a virtual machine, operating system functions such as process creation and IPC are implemented at the instruction level. Resources such as file descriptors, network connections, and windows can then be connected to the garbage collector. Inferno also provides mutual authentication, encryption, digesting of connections, and key management to make secure communications simple.

The comparison of Inferno and Java is informative. While the two systems have similar goals, the resulting implementations are strikingly different.

More information and a developers' kit is available for download from http://inferno.lucent.com. Source code licenses are available to universities for research or teaching as well as to commercial companies interested in deployment.