

---

# Introduction



---

## What is the Inferno system?

The Inferno system is a small, portable and secure network operating system designed to be network independent and deliver services through a variety of existing and emerging networks, providing universal access to resources and information.

The Inferno system runs on very small amounts of memory allowing it to run on small devices such as screen phones, set-top boxes, Personal Digital Assistants (PDAs) and Network Computers (NCs). The Inferno system and useful applications can run on these devices with less than 1 megabyte (MB) of RAM.

Applications written for the Inferno system are platform independent. This means that the same program, *without modification*, can run on any platform that supports the Inferno system. The applications run on a virtual machine that allows you to write the program once and run it in many environments.

The Inferno system can run as the native operating system, or hosted on top of an existing operating system. Currently, the Inferno system runs as a native operating system on Intel x86, AMD 29000, MIPS, Motorola

68030 and ARM processors, and hosted on Windows 95, Windows NT, and on the Solaris SPARC Unix system.

The Inferno operating system includes:

- The network operating system kernel
- A programming language called Limbo
- A “virtual machine” called Dis
- A communications protocol, called Styx
- Inferno API (application programming interface)
- Platform-independent graphics libraries
- Security and authentication
- Toolkits

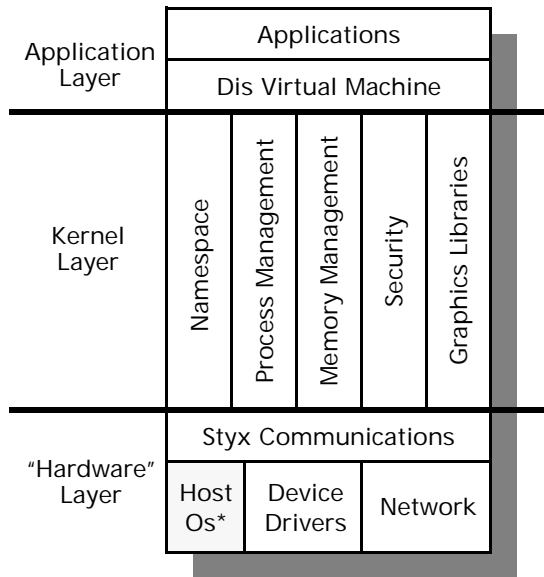
Limbo is a general purpose programming language intended for applications running distributed systems on small computers. It supports modular programming and its syntax is similar to that of the C language. Limbo programs use the Inferno API that includes interfaces for networking, security, graphics, and other services.

Limbo is interpreted by the Inferno system’s virtual machine, called Dis. Dis can interpret the Limbo program, or compile it “on-the-fly” at run time (also known as “just-in-time” or JIT compiling). Dis allows a Limbo application to execute on any supported platform without modification. It provides a robust garbage collection system to reclaim memory no longer in use. This feature adds to the efficiency of the Inferno system’s memory usage.

Dis transfers Limbo program instructions to the Inferno kernel that provides the necessary services for process, file, and other resource management.

If the application requires a remote resource, the Inferno kernel uses the Styx protocol to communicate with the remote machine. Styx insulates Inferno applications from details of the physical network. It provides transparent communications over a variety of networks with strong security capabilities built in.

Figure 1-1. Inferno Components Diagram



\* Host OS available in Emulation Environment

You will read more about these components later in this guide.

## Why Use the Inferno system?

---

The Inferno solution provides flexibility not available with other networking software. It is a small, portable network operating system that includes a general purpose programming language, security and networking support. It provides efficient, secure and reliable computing for local area, wide area, and global networks.

The Inferno network operating system enables you to:

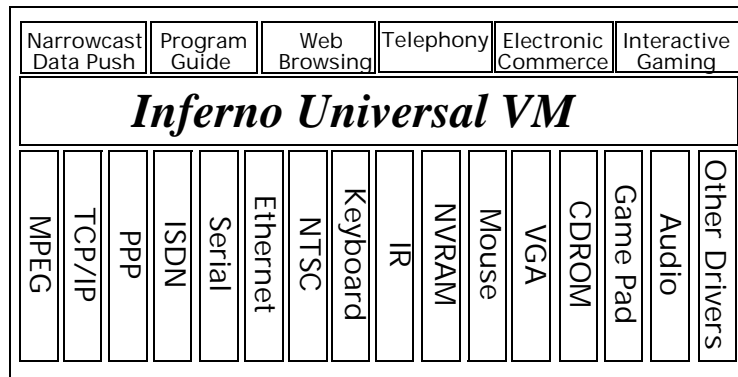
- Write applications once and run them on any Inferno-enabled system on the network. For an application, the same source and binary code can be made portable across platforms in native and hosted environments. See diagram showing Portability Across Platforms in Figure 1-1 below.
- Prepare for the future; run your Inferno applications on today's platforms and networks and run them unmodified on platforms and networks of the future
- Create a pool of resources (called a namespace) that can be transparently accessed by the application as if all the resources are local
- Easily manage the diverse set of resources that make up your Inferno namespace where data, devices, network connections and interfaces to services are all represented as a hierarchy of file names.
- Securely transmit data across the network

The growing network operating system market includes networked device and server manufacturers, application developers, content providers, network service providers, server manufacturers, and small client device manufacturers.

## How the Inferno System Can Be Used

---

There are numerous applications for which the Inferno system is a natural solution.



Examples of applications that are well suited for the Inferno system include:

- Distributed applications with multiple user interface devices (for example a PC, TV, and hand-held computer)
- Internet access devices with limited memory (Internet terminals)
- Multimedia personal computers
- Television set-top boxes
- Game consoles
- Advanced telephones
- Personal Digital Assistants (PDAs)

- Inexpensive network computers (NCs) that need to share data or resources

For example, a large computer may store the information necessary for accessing movies, including information about the movie, such as the length of the movie, date of production, actors, and price to rent. A very small VCR or set-top box running the Inferno system can attach to this server and use the resources on the server as if the resources were local to the small device.

A demonstration user environment and several sample applications that illustrate specific features of the Inferno operating system are included with the software distribution. The window manager environment and the applications are discussed in Chapter 3, *Using the Inferno System*.

The Window Manager, shows the Inferno system's capabilities of providing a window-based graphical user interface like Windows 95 or X-Windows. This environment uses the Inferno system's built-in graphics library called Limbo/Tk, an implementation of the Tk toolkit. The Window Manager includes demonstrations for e-mail, Internet and Web access, multimedia viewers, system tools, and other typical desktop utilities.

## Unicode Support

---

Unicode is a widely accepted standard for representing multiple languages and symbols. Unlike ASCII or other standards, it can represent tens-of-thousands of characters. Internally, the Inferno system represents all text strings as Unicode characters.

When binary data is transmitted over a network connection, it is automatically translated into UTF-codes, which are compatible with Unicode as well as ASCII codes, for transmission. This makes the transmission byte-order independent, so that Inferno-enabled systems do not have to worry about the underlying machine architecture.

Inferno device drivers take care of this automatically. However, API functions are provided for UTF conversion, if necessary. For example, an HTML web browser, which may be receiving data from non-Inferno systems, does a Unicode translation of data for the Inferno system.

## Inferno Terminology

---

The Inferno system introduces several terms:

- Inferno network operating system

The package of components that includes an operating environment, the Limbo programming language, the Dis virtual machine, and the Styx communications protocol.

- Limbo programming language

Limbo is a language that is much like the popular programming languages C and Pascal. It is modular, so that it can be distributed among many applications. Limbo uses the Inferno API, a set of library functions, that includes interfaces for networking, file management, graphics, and more. A program written in Limbo runs in exactly the same way in any Inferno environment.

- Dis virtual machine

The Dis virtual machine takes program instructions and interprets them or compiles them at run time. Dis allows applications written for the Inferno system to be executed on any platform that supports the Inferno system.

- Styx communications protocol

Styx is the standard protocol used by the Inferno system to communicate across a network. Styx provides file manipulation as well as other generic operating system resource functions. Styx commands are independent of the underlying network protocols or hardware.



- Namespace

Files accessible to an application comprise that application's *namespace*. An application can build its own unique private view of the resources and services that it needs, whether local or remote. Each set of resources and services is represented as a hierarchy of files and is accessible by familiar file access operations.

## Inferno Specifications

---

Tables 1-1 through 1-3 outline the currently supported hardware, software, and general features of the Inferno operating system.

Table 1-1 Supported Hardware

*Microprocessors (native)	<ul style="list-style-type: none"> <li>■ Intel x86, Pentium</li> <li>■ SPARC</li> <li>■ AMD 29000</li> <li>■ ARM</li> <li>■ MIPS (R4k or later)</li> <li>■ Motorola 68030</li> </ul>
Ethernet Adapters	<ul style="list-style-type: none"> <li>■ NE 2000</li> <li>■ NE 3210</li> <li>■ 3COM 509</li> <li>■ 3COM 589</li> <li>■ Western Digital SMC WD80[01]3</li> <li>■ NSC DP8390</li> <li>■ NSC SMC 83C90</li> </ul>
Graphics Adapters	<ul style="list-style-type: none"> <li>■ VGA standard (includes SVGA)</li> <li>■ Sirtus Logic 542X</li> <li>■ MACH 64</li> <li>■ S3</li> </ul>
Sound	<ul style="list-style-type: none"> <li>■ SoundBlaster 16 on native Intel x86</li> <li>■ Solaris on SPARC</li> </ul>
Modems	<ul style="list-style-type: none"> <li>■ Hayes serial compatible</li> </ul>

\* Not included in the evaluation copy of the Inferno system

Table 1-2 Supported Software

---

Operating Systems	<ul style="list-style-type: none"><li>■ Microsoft Windows NT 3.51, 4.0</li><li>■ Microsoft Windows 95/98</li><li>■ Sun Solaris 2.5.1 and 2.6 on SPARC</li></ul>
Networks	<ul style="list-style-type: none"><li>■ TCP/IP</li><li>■ UDP/IP</li></ul>
Databases	<ul style="list-style-type: none"><li>■ ODBC in Windows 95 and NT</li><li>■ Informix</li><li>■ Sybase</li><li>■ Oracle</li></ul>
Languages	<ul style="list-style-type: none"><li>■ Limbo</li><li>■ *PersonalJava</li></ul>

---

\* Not included in the evaluation copy of the Inferno system

Table 1-3 Inferno Features

Kernel	<ul style="list-style-type: none"> <li>■ Complete system and basic applications run in 1Mb</li> <li>■ Runs directly on a wide range of hardware architectures, or hosted on standard operating systems</li> </ul>
Limbo programming language	<ul style="list-style-type: none"> <li>■ Portable, machine-independent</li> <li>■ C-based</li> <li>■ Type-safe</li> <li>■ Multi-threaded</li> <li>■ Modularized</li> <li>■ Interpreted or just-in-time (JIT) compilation</li> <li>■ Inter-process communication channels</li> <li>■ Abstract data types</li> <li>■ Automatic garbage collection</li> <li>■ Designed for network applications</li> <li>■ Enables networked administration and application distribution</li> <li>■ Dynamically configurable</li> <li>■ Integrated into OS</li> <li>■ Executes within 1.5 to 2.5 times of compiled C</li> </ul>
Dis Virtual Machine	<ul style="list-style-type: none"> <li>■ Language neutral (can support multiple languages)</li> <li>■ Memory-to-memory architecture</li> </ul>
Styx Communication Protocol	<ul style="list-style-type: none"> <li>■ Provides a common interface to all networks</li> </ul>

Table 1-3 Inferno Features—*Continued*

---

Security	<ul style="list-style-type: none"><li>■ Public key encryption/authentication</li><li>■ 40-bit RC4 (international)</li><li>■ Unlimited length RC4 (US)</li><li>■ 56-bit DES-ECB (US)</li><li>■ 56-bit DES-CBC (US)</li><li>■ SHA</li><li>■ MD5</li></ul>
Developer's Tools	<ul style="list-style-type: none"><li>■ Compiler</li><li>■ Limbo Debugger</li><li>■ Graphics libraries, including Limbo/Tk implementation</li></ul>
Sample Applications	<ul style="list-style-type: none"><li>■ E-mail</li><li>■ HTML Web Browser</li><li>■ Text Editor</li><li>■ Notepad</li><li>■ QuickTime and .AVI players</li><li>■ .GIF viewer</li><li>■ System Tools</li><li>■ Telnet</li><li>■ Window Manager</li><li>■ Connection Server</li><li>■ File Server</li><li>■ Ftp Access</li><li>■ Memory Usage</li><li>■ Clock</li><li>■ Dis Module Manager</li><li>■ Task Manager</li><li>■ Shell</li></ul>

---

## Inferno Specifications

---