Solaris SunOS

SunOS 5.0 Release Report

A Technical Report

Table of Contents

SunOS 5.0 Release Report

Kim Ingram

Introduction

SunOS™ 5.0 with enhanced ONC™ is the foundation of Solaris™ 2.0, SunSoft's™ next generation distributed computing solution. Solaris 2.0, SunSoft's single, integrated system software solution, is comprised of SunOS, enhanced ONC, OpenWindows™ V3, DeskSet™ V3, and OPEN LOOK®.

Solaris 2.0 incorporates the industry's leading 32-bit, multitasking, commercially-proven operating system, combined with the leading solution for heterogeneous connectivity. More than 1.3 million nodes are installed.

SunOS 5.0 is SunSoft's end-user operating system release based on System V, Release 4 (SVR4). The UNIX® standard, SVR4 embraces the leading UNIX variants: System V, BSD, SunOS and Xenix™, uniting more than 80% of the installed base of 10 million UNIX users.

Although the foundation of SunOS 5.0 is based on SVR4, SunSoft has added extensive functionality in areas such as symmetric multiprocessing with multithreads, real-time functionality, increased security and improved system-administration. This paper outlines the steps of migrating to SunOS 5.0 and describes important product features.

The continuing success story of UNIX System V spans more than two decades of advancement and standardization in the computer industry. By the mid-1980s, every major vendor in the computer market had decided that open systems were required for the success of future products. Today, UNIX System V is the cornerstone of the resulting open system market. Its versatility allows growth in areas such as distributed computing, graphical user interfaces, connectivity, security, internationalization support and much more.

In 1990, the worldwide UNIX market was $22.2 billion. The U.S. had the largest share of the total UNIX market followed by Europe, and the Pacific Basin. Compared with proprietary operating systems, systems sold with UNIX as the primary operating system constitute a larger growing market segment. The Pacific Basin and the balance of the world represent a future opportunity for UNIX.

UNIX System V was the leading industry UNIX variant in 1990. InfoCorp expects System V to continue to be the leading UNIX standard. A 15% yearly growth rate between 1990 and 1995 for UNIX System V is expected. In 1995, the UNIX systems market will be comparable to the size of the DOS market today. The data, provided by InfoCorp 1991, represents an independent analysis of the computer industry.

SunOS 5.0 gives software developers and end users the benefits of a standard operating system — broad compatibility, a growth path, reduced time to market. It also delivers a very functional, powerful product reflecting years of refinement. SunOS 5.0 provides many advantages: portability, scalability, interoperability and compatibility.

Portability

SunSoft's SunOS is portable across multiple vendor platforms. Software conforming to an Application Binary Interface (ABI) will run as shrink-wrapped software on all vendor systems with the same microprocessor architecture. This enables application developers to reduce their software development costs and bring products to market quickly and users to upgrade hardware while retaining their software applications and minimizing their conversion costs.

Scalability

Over time, applications become more widely used, and require more powerful systems to support them. To operate in a growing environment, software must be able to run in a wide power range and must be able to take advantage of the additional processing power. SunOS runs on machines of all sizes from laptops to supercomputers.

Interoperability

Heterogenous computing environments are a reality today. Users purchase systems from many vendors to implement the solutions they need. Standardization and clear interfaces are critical to a heterogeneous environment, allowing users to develop strategies for communicating throughout their network. SunOS systems can interoperate with every popular system on the market today, and applications running on UNIX can intercommunicate easily.

Compatibility

Computing technology continues to advance rapidly, but the need to remain competitive requires vendors to minimize their costs and to maximize their investments. SunSoft will ensure that as new technology is introduced, the existing software investment is preserved. Users can take advantage of today's solutions and still be compatible with tomorrow's technologies.

Preparing Applications for SunOS 5.0 Transition

Plan Ahead for the SunOS 5.0 Migration

As application developers look ahead and begin to plan for SunOS 5.0, there are steps which you can take today which will simplify the migration of your applications in the future. Developers can prepare for migration by going through this five-step checklist:

- Move to SunOS 4.1
- Convert to OpenWindows
- Compile using System V Libraries
- Dynamically link your application
- Choose an ANSI C Strategy

SunOS 4.1

SunOS 4.1 is the bridge to SunOS 5.0 for applications already running on SPARC® platforms. Using 4.1 as a development platform gives developers a head start in migrating to SunOS 5.0. The more work done on the application under SunOS 4.1, the less work will remain to be done on SunOS 5.0.

Software engineers who use Sun OS 4.1 as their development platform have access to the richest set of software development tools in the industry, available from Sun and from third-party sources.

Releasing an application based on SunOS 4.1 provides a path to SunOS 5.0 partially due to the large and rapidly growing installed base of SunOS 4.1 systems. SunSoft estimates that slightly more than half of its existing base of systems are currently running releases based on SunOS 4.1, and that number is growing by more than 200,000 systems yearly.

OpenWindows

The next step for applications headed for SunOS 5.0 is to convert existing SunView™ applications to run in a native OpenWindows environment. Although existing dynamically linked SunView applications continue to run in binary compatibility mode on SunOS 5.0, the native window system for SunOS 5.0 is OpenWindows. Therefore, SunOS 5.0 does not provide the facilities for creating new applications based on SunView.

For internal application developers in large organizations, the most sensible strategy is to run in SunView binary compatibility mode on SunOS 5.0 in the short term, and to convert to OpenWindows in the medium term. SunOS 4.1 is the best development system to use as a base for that strategy.

For ISVs, SunSoft recommends re-release of applications in native mode. This path provides ISVs access to the broad set of SVR4 platforms. Testing costs are minimized by rereleasing as opposed to testing under both binary compatibility and later for native mode.

System V Libraries

To minimize the conversion effort needed later on, SunOS 5.0 applications should be compiled in the SunOS 4.1 System V environment using the System V libraries. Because SunOS 4.1 conforms to the System V Interface Definition (SVID) Issue 2, applications can be written in a manner that is source-compatible with SunOS 4.1 today and with SunOS 5.0 tomorrow.

Applications which continue to rely on the BSD-derived libraries of SunOS will require more conversion effort.

The *Pipeline to Solaris 2.0* document from SunSoft will outline the guidelines that can help with this process.

Dynamic Linking

SunSoft recommends that applications be dynamically linked. Dynamic linking benefits applications by reducing the overall system-wide working set of most executables. This improves overall runtime performance and insulates the application from changes in the underlying operating system.

Applications that are intended to run on SunOS 5.0 using the Binary Compatibility Package (BCP) must be dynamically linked. The BCP is a transition tool that allows end users to migrate to SunOS 5.0 while running their existing SunOS 4.x software in the short term.This provides a smooth progression and simplifies the creation of a market for native SunOS 5.0 applications.

Looking ahead, developers who plan to release binary compatible SunOS 5.0 SPARC applications on the rapidly growing number of SPARC hardware platforms should be aware that applications are required to be dynamically linked by the SPARC Compliance Definition (SCD 2.0) — the SPARC binary compatibility guideline which enables UNIX shrink-wrapped applications.

ANSI C

The purpose of ANSI C is to promote application portability, which is also a major goal of SVR4. Does this mean that applications have to be modified for use with the ANSI C? Not necessarily. Software developers have several strategies available to them.

Developers can start by choosing an appropriate ANSI C strategy and then begin working with the Sun C 1.1, an ANSI C-compliant compiler. Sun C 1.1 can also be used to supply performance boosts to SunOS 4.x applications. Using Sun C 1.1 on SunOS 4.1, software developers who wish to take advantage now of ANSI features, such as function prototypes and improved type qualifiers, can do so knowing that changes to their sources will migrate to the ANSI C compilation environment on SunOS 5.0.

An alternative strategy is to leave SunSoft common C source files unchanged. The ANSI C compiler offers options that accept inputs of SunSoft common C sources. In this strategy, the compiler merely warns where common C and ANSI C semantics differ. This option will be available on a compiler based on SunOS 5.0 systems.

SunOS 5.0 Features

The standards and specifications supported by SVR4 — SVID3, SCD 2.0, XPG-3 (POSIX 1003.1), IEEE 754 and the DDI/DKI — make this release unique. The SunOS 5.0 interfaces are the standard SVR4 interfaces: ANSI C, OpenWindows and the Bourne shell. Those who are familiar with previous SunOS releases will find these major differences between SunOS 4.x and SunOS 5.0:

- Development of new SunView applications is no longer supported; however, existing SunView applications run in compatibility mode to facilitate migration of applications to the native SunOS 5.0 system.
- Networking is based on Transport Layer Interface (TLI), STREAMS™, and TCP/IP. Sockets are supported in a compatibility user-level library.
- In the SunOS 5.0 release, device drivers must conform to the Device Driver Interface/Device-Kernel Interface (DDI/DKI) specifications. Previous SunOS drivers must be revised to incorporate certain features of the DDI/DKI.
- ANSI C is the primary SunOS 5.0 programming language, not Sun C. However, the SunSoft C syntax is supported by the `-Xs` option to `cc`.
- New file system types are supported.

User Interfaces

Character User Interfaces. Not all SunOS 5.0 systems will have or require bit-mapped graphics: for example, a character-based print server. The SunOS 5.0 system software has three tools for creating forms and menus that display on character-based terminals. These tools help to create the familiar features of a bit-mapped graphical user interface on character-based terminals by reducing coding time.

- **Form and Menu Language Interpreter (FMLI).** The FMLI is a high-level language interpreter. Its principal advantage is that it allows you to create consistent, user-friendly interfaces across applications without extensive programming. FMLI was enhanced for its release in SVR4.
- **Framed Access Command Environment (FACE)**. The FACE is a menu-based interface to the shells. Users choose tasks from menus and enter a task's parameters into small windows.
- **The Extended Terminal Interface (ETI)**. The ETI is the standard programming interface for character-mode screen and text operations. It consists of three libraries: `curses`, `terminfo`, and a High Level Function Library:
  - The `curses` library now supports multibyte character code sets and color terminals.
  - The `terminfo` database contains information about the kinds of terminals that are installed on a system. Programmers use the information to generate screen displays.
  - The High Level Function Library subroutines create forms, menus, and panels.

SunOS 5.0 Internationalization Features

Internationalization makes it easier to customize applications for different languages and cultures. The standards that define internationalization are:

- ANSI C
- XPG-3 (the POSIX 1003.1 subset)
- Multi-National Language Supplement (MNLS)

Multiple Character Sets and Multibyte Characters. The UNIX operating system historically provided development support for the ASCII code set. However, the ASCII code set only supports a limited number of the world's spoken languages. The SunOS 5.0 release supports the Extended UNIX Code (EUC) as specified by the Multi-National Language Supplement (MNLS).

The Extended UNIX Code allows programs to use code sets other than ASCII, including 8-bit/1-byte European code sets and 2- and 3-byte Asian code sets.

Wide Characters. Wide characters is a new data type that simplifies internationalization programs to address characters instead of bytes. The SunOS 5.0 release supports wide characters in accordance with the Multi-National Language Supplement (MNLS) 3.2, ANSI C, and XPG-3 specifications. This enhances portability of existing MNLS and internationalized applications.

Locale Databases

Locale databases include code-sets tables, rules for collating the tables to adapt to a locale, and rules for expressing date and time. Users can add to the database, defining new code sets and devising rules for collating tables.

In addition to English, the SunOS 5.0 system software supports databases for these languages:

- French
- German
- Italian
- Swedish

Locale Conventions. Expressing system variables, like the date and time of day, in the formats or conventions of other countries is also an internationalization feature. The SunOS 5.0 system provides this using locale conventions. A locale is a language group that uses a consistent set of naming formats, or conventions. Convention information is stored in subdirectories in `/usr/lib/locale`.

Localized Commands. Many commonly used UNIX commands are localized in the SunOS 5.0 environment. The commands draw on locale information in subdirectories in `/usr/lib/locale`.

Localized Bitmap Fonts. The SunOS 5.0 system supports localized bitmap fonts.

Message Databases. For full internationalization, screen messages must be expressed in the language of the locale. Messages can be stored in language-specific databases in subdirectories. Application developers who want to create internationalized applications have an easier task because SunOS 5.0 support usage of message catalogs according to the XPG-3 specification.

System Interfaces

The SunOS 5.0 environment contains system interfaces (commands, system calls and library routines) from the ABI, SVID3, SVR4, and SunOS 4.1.x.

The Source Compatibility Package contains most of the SunOS/BSD system interfaces not included in the base SVR4, or which have changed from SunOS 4.1. Setting the `PATH` environment variable searches the directories where the SunOS/BSD system interfaces are stored in addition to searching the standard SunOS 5.0 directories. However, the compatibility package is provided only as interim support for SunOS 4.x applications; developers should port applications to the SunOS 5.0 release for a permanent solution.

Shells

There are three SunOS 5.0 command shells—Bourne, C, and Korn. The shells allow users to execute UNIX commands and scripts from a command line. The Korn Shell is new; it is an interactive shell for application developers.

SunOS 5.0 System Administration Features

The system administration features are grouped into these categories:

- *Device Information.* Administrators can use these optional utilities to obtain information about installed devices, including device names, attributes, and accessibility. Administration can be simplified by creating device allocation pools — a feature not previously found in UNIX systems.
- *File System Administration.* These utilities enable system administrators to create, copy, mount, debug, repair and unmount file systems, create and remove hard file links and named pipes, and manage volumes.
- *Interprocess Communication.* The two interprocess communication utility routines create, remove, and report on the status of the system's inter-process communication facilities (message queues, semaphores, and shared memory IDs). They provide information helpful in tuning the system.
- *Process Management.* The process management utilities control system scheduling. Using these utilities, administrators can generate reports on performance, logins, disk access locations and seek distances to better tune

system performance. In addition, they can change the system run level, kill active processes, time the execution of commands, and change the default scheduling priorities of kernel, timesharing, and real-time processes.

- *Software Package Management.* The ABI requires that operating system and application files be grouped in discrete *packages*. A package includes the components of the application, plus two information files: `pkginfo` and `prototype`. `pkginfo` contains parameters describing the package; `prototype` lists the full contents of the package. Packaging scripts and additional package information files may also be included in a single package.

  Software package management simplifies installing and updating software. Administration is simplified because the method for managing system software and third party application is now consistent. The tools for creating software packages are in an Application Packaging Tools library.

- *System Accounting.* The accounting utilities enable system administrators to track system usage by CPU, user, and process for better resource allocation.
- *System Information.* These utilities report system memory and system configuration. The system administrator can use the utilities to change the names of the system and the network node.
- *Time Zone Information.* Time zone utilities convert system time to local time.
- *User and Group Management.* With these utilities, a system administrator can create and delete entries in group and password databases, specify default home directories and environments, maintain user and system logins, and assign group and user IDs. The utilities support both primary and supplementary user groups.

SunInstall. SunInstall is the SunOS 5.0 software installation program. It is a flexible, easy-to-use tool for installing and upgrading systems. SunInstall has been upgraded to use a graphical user interface that is more convenient than the character menus of previous releases. The program saves the information in a database, creating a reliable record of every installation; the record can be used to interrupt and resume installations safely.

Service Access Facility (SAF). The SAF is an interface between requests coming from outside the system and the system access points that respond to the requests. Access to the system can be controlled from within the facility, making connection request management more reliable and consistent.

In the SAF, port monitors detect activity at external ports. SunOS 5.0 system software has three kinds of port monitors, managed by the Service Access Controller:

- `listen` is a general purpose network monitor that provides connection services regardless of a network's communications protocol.
- `inetd` is a DARPA port monitor for TCP/IP networks.
- `ttymon` is a serial port monitor. One `ttymon` process can monitor any number of serial ports. `ttymon` replaces `getty`, but the two are symbolically linked to maintain compatibility with applications that invoke `getty`.

SAF passes outside requests for network services to the Listener. Every transport provider has a Listener program. Listeners determine which services are needed to respond to the requests and spawn the service processes.

NFS® and RFS Administration. The NFS and RFS (Remote File Sharing) services are administered through the Distributed File System package. NFS and RFS provide the same functions as in previous releases. Command syntax for both has been made more consistent to simplify administration.

Network Information Service Plus. Network Information Service Plus (NIS+) is the distributed name service for networks. NIS+ is an enhanced version of NIS found in previous releases. NIS+ is backwards compatible with existing NIS networks. NIS+ provides:

- A hierarchical name space for simpler administration in large organizations
- Better security model than NIS
- A faster updating method of propagating changes to MIS map

NIS+ provides a flexible security model for name service entries. UNIX-style permissions (read, write, execute) can be assigned for every item in the name space; this offers tighter control than previous models.

SunOS 5.0 Security

ASET (**Automated Security Enhancement Tool**). ASET is a utility that improves security by allowing system administrators to check system files settings, including permissions, ownership, and file contents. ASET warns users about potential security problems and where appropriate sets the system files permissions automatically according to the security level specified.

ASET allows system administrators to quickly check and monitor systems on an ongoing basis. It enables system administrators to adjust the levels of security on their networks from the most open access to the restrictive.

Shadow Password File. The shadow password file improves security with improved password aging and login controls. Encrypted passwords are stored in a separate file, unreadable to potential intruders.

Authentication. The SunOS 5.0 security offers three authentication modes; administrators can choose the most appropriate for their site:

- UNIX authentication (traditional)
- Secure RPC (familiar to SunOS administrators)
- Kerberos (Kerberos network authentication service)

SunOS 5.0 File Systems

Directories. The SunOS 5.0 directory tree at the highest level is structured the same as in previous releases. Prior conventions for directory usage still apply:

- The /(root) directory contains the system boot files.
- The `/etc/*` contains machine-specific files.
- The `/usr` directory contains system files and directories that can be shared with other users. Files that will only run on certain types of machines, SPARC-specific for example, are stored in `/usr`; files like the `man pages`, which can be used by all types of machines, are stored in `/usr/share`.
- The `/home` directory is the mount point for the user directories. It contains the user's home directories and files.
- The `/var` directory contains system files and directories that are likely to change or grow over the life of the local system. These include system logs, `vi` and `ex` backup files and `uucp` files.

Symbolic Links. A symbolic link connects a local file to a file or directory on another file system. When you create a symbolic link in a local directory, it becomes a logical directory. A list of the logical directory shows all files in the symbolic link, just as though the files were on the local machine. You can create logical directories that include files anywhere on the network.

File System / File Management

VFS (Virtual File System). SunOS 5.0 system uses the Virtual File System (VFS)/ virtual node (*vnode*) approach to file management. (A *vnode* is any file (belonging to any file system type) accessed by the kernel.) Programs and users access different file systems through the standard VFS interface. Kernel functions also use the interface; however, tasks specific to a file system type must be performed by file system dependent routines.

The VFS/Vnode interface provides these features:

- **Standard file management system calls** for mounting and unmounting different file systems, such as NFS, RFS, UFS or High Sierra File System (`hsfs`). This provides standard system calls to access a variety of information, such as compact discs on the `hsfs`, or kernel processing instructions on the `proc` file system.
- **Dynamic Inodes** are files that are dynamically made available to the kernel as required. Dynamic allocation removes arbitrary limits on the number of files that can be open at any one time. Dynamic inodes is a SunOS 4.1.x feature ported to SunOS 5.0.
- **Fast Symbolic Links.**The SunOS 5.0 symbolic links are cached in the resident inode cache. As a result, references to frequently used destination inodes are returned faster, because the references do not require disk access.
- **Support for POSIX.1, BSD and XENIX** file management system calls and system interfaces.

SunOS 5.0 File System Types

The SunOS 5.0 environment supports three file system types: disk-based, pseudo, and network. The System V file system is not included in SunOS 5.0 because of its restrictions and limited usefulness compared to the UNIX file system, particularly for current SunOS customers.

Disk-based file system types

Disk-based systems are file systems written for a particular hard-disk format. There are three SunOS 5.0 disk-based file system types.

BSD Fast File System (BSD FFS). The BSD FFS is the SunOS 5.0 default file system type. BSD FFS combines the features of BSD and the 4.3 Tahoe Fast File System. A summary of its features are:

- BSD FFS allows blocks as large as 8K and supports all SVID3 file operations.
- Support for more inodes and cylinders per cylinder group; the number of inodes and cylinders has no limit.
- Supports new-generation hard disks by de-referencing variable-length list structures.
- SunOS 4.1 UNIX file systems can be mounted on SunOS 5.0 systems.

`hsfs` (High Sierra File System). The `hsfs` file system supports the High Sierra International Standards Organization (ISO) 9660 CD-ROM, and the RockRidge Group formats. Using existing system calls, `hsfs` provides dynamic, transparent access to files in either format.

pcfs. This is a DOS-based file system that allows users to access data and programs written for DOS-based personal computers.

Pseudo file system types

Pseudo file system types are logical groupings of files that reside in disk-based systems. There are nine SunOS 5.0 pseudo file system types.

`tmpfs` (temporary File System). `tmpfs` is a disk caching file system type. It stores disk and network reads and writes in local memory. `tmpfs` files are memory-resident; these are lost when the system reboots or loses power. The `tmpfs` file system is implemented as an optional kernel module.

`proc` (Process). The `proc` file system gives access to images of executing processes. The kernel creates files for the processes in the `/proc` directory, allowing debuggers and other development tools to access the address space of the processes with VFS system calls.

`specfs` (Special File System) `The specfs` file system type is a common-code interface to device files. It supports the character-special and block-special file types.

`fd` (File Descriptor). The `fd` file system type allows processes to pass file descriptors as filenames.

`fifo` (First-In First-Out). The `fifo` file system gives processes common access to named pipe files.

`namefs`. The `namefs` file system type allows dynamic mounting of file descriptors on file names; this redefines the meanings of the files. The `namefs` file system is most often used by STREAMS operations.

`xnamfs`. The `xnamfs` file system provides file compatibility with XENIX systems. It is a standard implementation of XENIX semaphore and data sharing, and it supports the XENIX VXNAM file type.

Network file system types

There are two SunOS 5.0 network file system types:

- **NFS.** NFS was developed by Sun and adopted by SVR4. To simplify interoperability, SunOS 5.0 can NFS mount file systems from previous SunOS releases and vice versa.
- **RFS (Remote File Sharing).** To simplify interoperability with existing SunOS RFS networks, SunOS 5.0 can mount RFS directories from previous SunOS releases and vice versa.

  RFS works on the directory level. When a client accesses data in RFS mounted directories, the data is cached in local memory where it can be accessed by local processes. This substantially reduces network traffic.

SunOS 5.0 Network Features

The SunOS 5.0 networking interfaces are very similar to SunOS 4.x interfaces. This is because SVR4 adopted most of the key features of the Internet Protocol(IP) suite network architecture. As a result, SunOS 5.0 machines are fully functional in SunOS 4.x networks.

The difference between SunOS 4.1.x and the new SunOS 5.0 networking facilities are:

- **Transport Layer Interface (TLI).** TLI replaces sockets as the default kernel-level network programming interface. However, sockets-based applications continue to run using a user-level compatibility library.

Modeled on the Transport Service Definition (ISO 8072), TLI defines a medium and protocol-independent interface between applications and network protocols. TLI allows applications to run over any transport protocol that supports the TLI. All transports, including TCP and UDP are accessible throughout the TLI. It is implemented as a user library using the STREAMS input/output mechanism.

- **Network Selection.** Network Selection, a new facility, enables an application to poll a computer's available networks until a usable network is found. The network administrator maintains a file of accessible networks; the file makes it unnecessary to code the network names into an application.
- **Name-to-Address Mapping (NAM).** NAM is a server lookup facility for TLI networks. Clients can find the addresses of servers on a network, connect to servers even if their addresses have changed, and select the most convenient routes to servers. NAM identifies servers by service, host computer, and network.

Sockets. Socket-based interprocess communication is the traditional BSD network interface. SunOS 5.0 networking implements sockets as a compatibility library based on STREAMS. All socket-based applications must be recompiled to run in the SunOS 5.0 network environment.

Internet Protocol (IP) Suite. In SunOS 5.0 networking, IP supports the full DARPA command set as implemented in both System V and BSD (including the special BSD command set, Berkeley Remote Commands). IP is compatible with any network that conforms to the DARPA standard. The suite is composed of the User Datagram Protocol(UDP) and the Transmission Control Protocol (TCP).

Transport Independent Remote Procedure Calls (TI-RPC). The TI-RPC libraries make up a standard protocol for invoking remote processes. TI-RPC library routines call processes with the syntax and semantics that are appropriate to each server. As a result, applications can use local syntax and semantics across the network.

TI-RPC provides the capability to develop distributed applications independent of the underlying network transport which can be specified at runtime. TI-RPC is compatible with RPC except for some minor differences in the TI-RPC library.

Remote Procedure Calls (RPC). SunOS 5.0 is interoperable with existing RPC systems. The RPC provided is unchanged for existing SunOS releases.

External Data Representation(XDR). XDR is a standard of data representation for networks that connect computers with different CPUs and operating systems. XDR defines data byte ordering and data type size. It enables processes on one computer to call procedures on other computers even though the computers' native data formats might be incompatible. XDR is unchanged from previous SunOS releases.

`inetd` Daemon. `inetd` is the BSD port monitor, or daemon, for internetworks. When a connection request arrives at a server, `inetd` spawns the server process and passes the network connection to the process.

SunOS 5.0 Kernel Features

Multithreaded (MT) kernel. MT provides for a symmetric multiprocessing kernel where multiple processors can execute the kernel at the same time. Applications can be structured as several independent computations rather than as one thread of control. The independent computations execute more efficiently because the operating system will handle the interleaving of the independent operations. This benefit of multithreaded kernel is known as application concurrency.

Virtual Memory. The SunOS virtual memory implementation was adopted by SVR4. Virtual memory replaces the previous System V memory management architecture. Its implementation in SunOS 5.0 has the same major features and functionality as in SunOS 4.1.x.

Virtual memory provides demand-paging; this allows the system to use disk space as pages of main memory. The core of the virtual memory system is a suite of routines that map files (including devices, ports, etc.) into the virtual address space of a process. This allows the process to access a file as locations in memory instead of having to use file operations and system calls. More than one process can map the same file, and with shared memory — another feature of virtual memory — the same copy of the file is mapped into each process.

STREAMS. STREAMS is a framework for character input and output (I/O) and has been implemented throughout SVR4. Most SunOS 5.0 character I/O is STREAMS-based, including pipes, along with *tty* and pseudo-*tty* terminal subsystems. The STREAMS-based pipes are compatible with pre-STREAMS SunOS program's pipes and with BSD 4.2 non-blocking I/O.

STEAMS is a flexible framework that is easily customized for application.

Autoconfiguration. Autoconfiguration routines load drivers and kernel modules, as required by the system, not necessarily at boot time. This feature eliminates additional time and effort required to build kernels when incorporating new devices, and also reduces performance loss due to unnecessary code bound into the kernel.

Paths to the system file and kernel modules are stored in the `/etc/system` file. The paths can also be specified at startup if you boot with the `-a` option.

Expanded Fundamental Types. ID data types (uid, pid, device IDs, etc.) and certain other data types are expanded to 32 bits. This improves the scalability of SunOS in large systems and for usage in large organizations.

Signal Interfaces. All versions of SVR4 implement the full POSIX.1 standard for signal interfaces. The interfaces enable programs to manipulate sets of signals, block and unblock signals, and examine pending signals before the signals are received.

Process Scheduling. SVR4 provides a flexible process scheduler that is well-suited for a variety of customer usage patterns.

The traditional UNIX timesharing scheduler is still available and recommended for most uses. An optional realtime scheduler is configurable for applications needing a fixed-priority scheduling mechanism.

Users can specify whether a process is timeshare or real-time.The system routinely adjusts the priorities of timesharing processes; only the user can change the priorities of real-time processes. A real-time process cannot be preempted; it has a higher priority than a timesharing processes. Processes can configure the properties of their schedulers while they run.

Users can make processes resident through memory locking, a privileged operation that holds pages in main memory that reduces time to access the page.

SunOS 5.0 offers a fully preemptible kernel to reduce dispatch latency. Other SVR4 kernels use the approach of multiple preemption points which results in somewhat longer latencies. The preemption causes the CPU to give a process more processing time. Reduced latency improves applications responsiveness and response to interrupts.

SVR4 also adopted the BSD high-resolution timing services, but with nanosecond granularity.

SunOS 5.0 Device Driver Interfaces

There are three types of interfaces for SunOS 5.0 device drivers: Device Kernel Interface (DKI); Device Driver Interface/Device Kernel Interface (DDI/DKI), brought to this release from SVR4; and Sun Device Driver Interface (Sun DDI). SunOS 4.x drivers must be revised to conform to the DDI/DKI.

Benefits of DDI/DKI conformance are:

- Device drivers have better source and binary compatibility guarantees across SPARC platforms than in previous SunOS releases. Developers and vendors can write one driver to support a peripheral on all SPARC platforms, thereby reducing development costs.
- Device drivers are loaded dynamically. This means that drivers are easier to install and devices are more easily accessed.

Device Driver Interface/Device-Kernel Interface (DDI/DKI). The DDI/DKI are a source standard that enhance portability of drivers across implementations of SVR4. These routines provide an interface to the kernel software, and are hardware independent.

Sun DDI. The Sun DDI interfaces to all SunOS 5.0 compatible architectures and kernel software.

Loadable Modules. Loadable modules are programs that can be executed as though they had been built with the kernel. They are device drivers and other kernel routines that the kernel can use without having been statically linked into the kernel.

When a module is first referenced, it is added to a running system. For example, not until data is sent to a set of ports is the driver loaded that controls access to the ports. This allows the kernel to be reconfigured dynamically. Developers can change one kernel module without having to change any other, and modules can be added and removed without having to rebuild the entire kernel. This makes driver development and installation easier.

SunOS 5.0 loadable modules model is an extended version of the model in previous SunOS releases.

SunOS 5.0 Standards

The standards to which SunOS 5.0 conforms can be classed as binary, source, or a combination of both. Source standards enhance compatibility of source code across SVR4 implementations. Binary standards enhance compatibility of binary code across SVR4 implementations.

Binary Standards

Application Binary Interface (ABI). The ABI specifies the binary environment for SVR4 applications. Supporting the ABI will allow developers to develop shrink-wrapped UNIX applications that will simplify the purchase and expand the availability of UNIX applications.

The ABI has these important advantages for developers:

- Applications that conform to the ABI will run on all ABI platforms. This allows development of shrink-wrapped UNIX applications and can simplify purchase and availability of the applications.
- The ABI is a complete and explicit definition of all aspects of binary interfaces, simplifying the adoption of applications to widely differing platforms.
- It is an open standard supported by many vendors.
- It has support for dynamic linking.

The ABI has processor-independent and processor-dependent interfaces.

- *Processor-independent* interfaces are common to all CPU platforms. They make up the generic ABI. The ABI interfaces for software installation, file formats, and libraries are examples of processor-independent interfaces.
- *Processor-dependent* interfaces are specific to a particular CPU. Examples of processor-dependent interfaces are the ABI interfaces for data representation, address space and page size, and dynamic linking tables.

If an application is written to both interfaces, it will run without a recompile on any ABI platform of the same CPU architecture.

SPARC Compliance Definition 2.0 (SCD 2.0). The SCD 2.0 is a superset of the SPARC ABI.

SPARC ABI. The SPARC ABI defines the dependent components specific to the SPARC architecture and instruction set. Applications written to the SPARC ABI can be developed with the assurance of running on all SPARC architecture-based systems.

Source Standards

System V Interface Definition, Third Edition (SVID3). The SVID3 defines many of the SVR4 system interfaces (commands, system calls, and library routines). Conforming to SVID allows developers portability across a wide array of SVR4 platforms increasing market opportunities while maintaining the existing software investment.

SVID3 is a source level standard. SunOS 5.0 is conformant to the SVID3. Applications that are written to SVID3 are source-compatible across all SVR4 implementations, including the SunOS 5.0 implementation. Applications that are ported to the SunOS 5.0 platform can conform to SVID3. This allows developers portability across a wide array of SVR4 platforms thereby preserving the investment in software.

POSIX P1003.1-1990. POSIX is a group of committees charted by the IEEE standards that define source level requirements for application portability. These working groups address features such as system calls, libraries, programmer tools, real-time processing, and system security.

POSIX.1 is the only completed, approved standard. The standard defines requirements for system interfaces. SunOS 5.0 conforms to POSIX P1003.1-1990.

X/Open Portability Guide, Issue 3(XPG-3). X/Open is a consortium of computer vendors who publish application-portability and connectivity guides. The XPG-3 addresses issues such as windowing, networks, system interfaces, file and data management, programming languages, and internationalization. Applications that conform to the XPG-3 specifications are portable at the source code level. Applications can be connected across networks independent of the network protocols when the networks conform to the XPG-3 standards.

The SunOS 5.0 system software conforms to XPG-3, and SunSoft expects the SunOS 5.0 environment to be XPG-3 BASE-branded.

American National Standards Institute X3.159-1989 (ANSI C). ANSI standard X3.159-1989 defines the program execution environment, syntax, semantics, library routines, and headers for the C language. ANSI C is the standard programming language of SVR4, and it is principal language supported by SunOS 5.0. For example, system header files include ANSI C features such as, function prototypes

ANSI C is incompatible with some SunSoft C features. SunSoft C source code must be edited before it is compiled with ANSI C. The C compiler `-X` option allows you to specify precedence or generate an error list when code differs from ANSI C. The *Compatibility and Migration Guide* explains the differences between SunSoft C and ANSI C in detail.

Internet Protocol (IP). The IP suite is the *de facto* industry standard for networking different kinds of computers. The IP defines protocols for network services such as email and file transfers.The IP package in SVR4 supports the DARPA command sets from both System V and BSD, plus other commonly used BSD networking commands.

Transport Level Interface (TLI). TLI, defines a protocol-independent interface between applications and network protocols. TLI replaces sockets as the native SVR4 network interface. Applications interface with the TLI instead of different network protocols, making applications more portable across differing network architecture. Socket-based applications can continue to run using a STREAMS-based compatibility library.

ANSI/IEEE 754 Floating Point Representations. ANSI/IEEE 754 defines a floating point standard to simply the portability of programs using floating point representations.

SunOS 5.0 Application Development

The SunOS 5.0 development environment provides two packages to make SunOs 4.x applications temporarily compatible in the SunOs 5.0 environment.

These compatibility packages are not substitutes for a full port. They are tools to use while you are doing a port. The Source Compatibility Package aids compilation of SunOS 4.x applications for the SunOS 5.0 system; the Binary Compatibility Package provides an environment where SunOS 4.x executables can run before being ported to the SunOS 5.0 environment. The packages run only under the SunOS 5.0 system software; you cannot use them on machines running a 4.x version Sun OS, nor create applications that will run on 4.x system software.

SunOS Binary Compatibility Package. The SunOS Binary Compatibility Package provides a SunOS 5.0 environment in which SunOS 4.x binaries can run. The applications must be dynamically linked and must:

- Not trap directly to the kernel
- Not write directly to system files
- Not use `/dev/kmem` or `libkvm` routines,
- Not use unpublished SunOS interfaces
- Not rely on non-SunOS drivers.

With this package, SunView applications can run under OpenWindows. You can continue using SunView applications until corresponding OpenWindows versions have been developed.

Expect applications to run somewhat more slowly under the Binary Compatibility package.

SunOS/BSD Source Compatibility Package. The *SunOS/BSD Source Compatibility Package Guide* is an enhanced version of the standard SVR4 BSD Source Compatibility Package. It contains BSD and SunOS commands, source compatibility libraries, and headers. With the package, you can create a SunOS 5.0 environment that is 4.x compatible and compile SunOS 4.x source code into an executable binary.

To create a SunOS-compatible environment, the package provides SunOS/BSD versions of the user commands. You can set the `PATH` environment variable to make the BSD commands the default, instead of the standard SunOS 5.0 commands. These SunOS/BSD commands are in `/usr/ucb`, whereas the standard SunOS 5.0 commands are in `/usr/bin`.

To compile existing SunOS 4.x code without porting it, the package provides libraries of most of the SunOS 4.x and BSD system interfaces. They are in the `/usr/ucbinclude` and `/usr/ucblib` directories. The package's versions of the C compiler (`/usr/ucb/cc`) and linker (`/usr/ucb/ld`) search these directories before they search the standard SunOS 5.0 directories. The binary code that is output by these commands is in ELF, the SVR4 binary format, so it is executable on SunOS 5.0. The code does not conform to the ABI, however, because the compatibility routines that the system uses often employ non-ABI interfaces.

Note – Do not rely solely on the Source Compatibility Package to compile 4.x source code successfully. Before compiling, you might have to edit your source for inconsistencies with the SunOS 5.0 environment.

Dynamic Linking. The SunOS 5.0 application environment supports dynamic linking of libraries. The linker uses the version numbers of the libraries and executables to link applications with the proper libraries, routines, and interfaces.

The linker supports the standard SVR4 system interfaces and object file formats. It is a user program, not a kernel function; it is mapped into the virtual address space of applications.

Executable Linking Format (ELF). SunOS 5.0 object and core files are in the new SVR4 binary format, called Executable Linking Format (ELF). Executables that are not in this format cannot be executed, except for SunOS 4.x object files running under the Binary Compatibility Package.

`a.out`. The input assembly language, as defined in the *SPARC Assembly Language Reference Manual,* corresponds with the SPARC instruction set defined in the *SPARC Architecture Manual,* Version 8.

SunSoft, Inc.
2550 Garcia Avenue
Mountain View, CA 94043

For more information, call 1 800 227-9227.

Printed in USA 9/91 91023-0 1.5K