Chapter
1

GETTING STARTED

**SYS-ED/
Computer
Education
Techniques, Inc.**

Objectives

You will learn:

- The features and advantages of UNIX.
- The components of UNIX.
- The purpose of application libraries.
- The differences between user and system mode.

1 UNIX: What is it

An operating system is a suite of programs which make the computer work; it is multi-user and multi-tasking system for servers, desktops and laptops.

The UNIX operating system was developed and introduced in the 1960s and has been in a constant state of evolution and development ever since. Some UNIX systems, including Solaris, also have a GUI graphical user interface. The GUI has many of the features of the MS Windows interface.

A knowledge of UNIX is required for operations which are not covered by a graphical program, or for when there is no windows interface available such as in a telnet session.

2 Components: UNIX Operating System

The UNIX operating system is made up of three parts:

• Kernel	• Shell	• Program

Kernel

The kernel is the core of the UNIX operating system; it allocates time and memory to programs and handles the filestore and communications in response to system calls.

Shell

The shell acts as an interface between the user and the kernel. When a user logs in, the login program checks the username and password, and then starts another program called the shell.

The shell is a CLI: command line interpreter. It interprets the commands the user types in and arranges for them to be carried out. The commands are themselves programs; when they terminate, the shell returns the user to another prompt.

A user can customize his/her own shell, and users can use different shells on the same machine.

Program

UNIX does not make a significant distinction between commands (e.g. user-level programs) for system operation and maintenance (e.g. cron), commands of general utility (e.g. grep), and general-purpose applications such as the text formatting and business applications.

3 Files and Processes

Everything in UNIX is either a file or a process.

- A file is a collection of data. They are created by users using text editors, running compilers etc.
- A process is an executing program identified by a unique PID: process identifier.

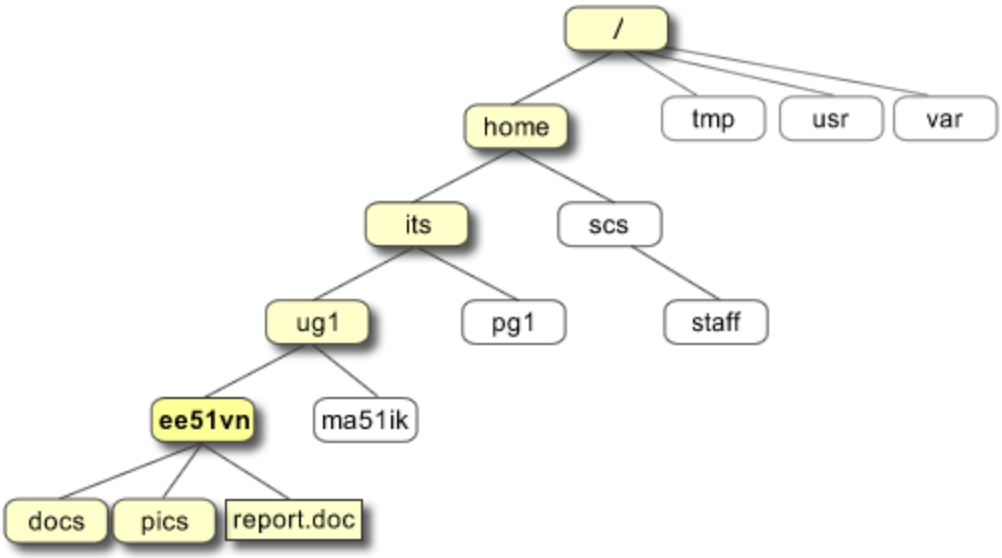
Examples:

- A document - report, essay etc.
- The text of a program written in some high-level programming language.
- Instructions comprehensible to the machine and incomprehensible to a user such as an executable or binary file.
- A directory, containing information about its contents, which may be a combination of other directories, subdirectories, and ordinary files.

4 Directory Structure

All the files are grouped together in the directory structure. The file-system is arranged in a hierarchical structure, like an inverted tree.

The top of the hierarchy is known as root; which is written as slash /.



5 UNIX Processes

A process is an instance of a running program. At any given time, a typical UNIX system will have many active processes, some of which are initiated when the machine was first powered up. Every time a user issues a command, UNIX starts a new process, and suspends the current process until the new process completes.

If three individuals are running the same program simultaneously, there will three processes.

All UNIX commands are programs, and thus contribute processes to the system when they are running. If 10 users are running mail, there will be 10 processes.

UNIX identifies every process by a Process Identification Number - pid; which is assigned when the process is initiated. When an operation is to be performed on a process, it typically will be referred to by its pid.

UNIX is a timesharing system, which means that the processes take turns running. Each turn is called a timeslice, which is set at a fraction of a second.

6 UNIX: Design and Concise Code

Prior to UNIX, a typical vendor operating system was written in assembly language and was extremely large in size.

UNIX has a relatively small amount of code written in assembly language. The kernel is written in assembly language; the remaining operating system code has been written in C, a high level language.

The development of the code for the UNIX operation system was done primarily in the high level language. Only small changes were necessary in the kernel. Over time, the kernel and associated software were extended until a complete operating system was written on top of the kernel in the C language.

7 UNIX Application Programming Interface

Many proprietary operating systems have a simplified view of application behavior.

- The typical application reads some data from disk, tape or a terminal and does some processing.
- Output is produced onto disk, tape, tape, terminal, or printer.

The operating systems generally provide facilities to support these types of facilities. However, as applications became more sophisticated, they required new features such as network access, multi-tasking, and interprocess communications. When a program makes use of these features, as a by product it will become more complex and difficult to maintain.

In UNIX because the C language was written to be used to implement an operating system rather than a traditional input-processing-output application, it was easy to draw upon these features from the C language without writing any assembly language.

When UNIX was distributed, users could write applications in C and make use of the operating system facilities. This allowed application developers to quickly develop sophisticated applications.

When adding new features in UNIX, such as networking, the standard approach is to provide an application program interface from the C language to access the new features.

8 UNIX Networking

In 1984, the University of California at Berkeley released version 4.2BSD which included an implementation of the TCP/IP networking protocols. This networking featureset and subsequent BSD releases provided a multi-vendor networking capability based on Ethernet networking. The networking support included, remote login, file transfer, electronic mail, and other important features.

As UNIX was ported onto different types of computer hardware the UNIX networking allowed many different types of systems to share and mutually use data. Networks consisting of many different systems could then be used as a large distributed system.

When SUN Microsystems added NFS: Network File System, this ability to share and mutually use data was significantly enhanced.

9 UNIX: Why Popular

Application portability and system programming issues have resulted in a number of hardware and software vendors selecting UNIX as their operating system. The net result is that there are a wide variety of UNIX systems available to users at attractive prices.

There are three primary reasons for the popularity of the UNIX operating system.

1. Since only a very small amount of code in UNIX is written in assembly language, it is relatively easy for a vendor to get UNIX running on their system.

    - Many vendors have made UNIX available on their systems in addition to proprietary operating systems.

    - UNIX runs on a wide variety of computer systems.

2. The application program interface allows many different types of applications to be implemented under UNIX without writing assembly language.

    - These applications are relatively portable across multiple vendor hardware platforms.

    - Third party software vendors can save costs by supporting a single UNIX version of their software rather than different vendor specific versions each requiring maintenance.

3. Vendor-independent networking allows users to easily network multiple systems from many different vendors.

Commencing in the 1980's, UNIX operating system has steadily gained in utilization and popularity.

10 User Interface

The UNIX operating system was designed and implemented by experienced programmers.

In recent years, there has been extensive work done in improving the user interface. Windowing interfaces have been added to UNIX such as X-windows, Suntools, NextStep, Motif, and OpenLook. Although these windowing interfaces do not change UNIX; they do provide a more intuitive interface. Each of the different user interfaces has some advantages associated with it.

Vendors such as SUN Microsystems have significantly improved the user interface for their particular versions of UNIX for users without windowing interfaces.

However, UNIX is less comprehensive and sophisticated in the end-user interface area than Microsoft Windows.

11 User Portability

Even with the shortcomings in its interface, a significant number of non-programmers use UNIX.

The primary reason for this is because UNIX runs on a variety of different computer systems ranging from small desktops to mainframe systems. Once a user has learned UNIX, the skills can be used on many different systems. This ability for a user to work on many different makes of computer systems without training is known as "user portability".

UNIX is an "open system" in that allows application portability, system interoperability, and user portability between many different computer vendor hardware platforms.

12 Solaris

Introduced by Sun Microsystems in 1992 as the successor to SunOS, the Solaris operating system, is typically referred to as Solaris. It is a free UNIX-based operating system.

Solaris is known for its scalability, especially on SPARC systems, as well as for being the origin for many innovative features such as DTrace and ZFS. Solaris supports SPARC-based and x86-based workstations and servers from Sun and other vendors. There also are currently efforts underway for porting it to additional platforms.

Solaris is certified against the Single UNIX Specification. Although developed as proprietary software, a majority of its codebase is now open source software as OpenSolaris.

12.1 Solaris History

In 1987, AT&T and Sun announced that they were collaborating on a project to merge the most popular UNIX variants on the market at that time: BSD, System V, and Xenix. This would become UNIX System V Release 4 (SVR4).[1]

On September 4, 1991, Sun announced that it would replace its existing BSD-derived UNIX, SunOS 4, with one based on SVR4. This was identified internally as SunOS 5; however a new name was placed on the code for marketing purposes: Solaris 2.

While SunOS 4.1.x micro releases were retroactively named Solaris 1 by Sun, the Solaris name is used for referring to the SVR4-derived SunOS 5.0 and later.

The justification for this new "overbrand" was that it encompassed not only SunOS, but also the OpenWindows graphical user interface and Open Network Computing (ONC) functionality.

The SunOS minor version is included in the Solaris release number; for example, Solaris 2.4 incorporated SunOS 5.4. After Solaris 2.6, Sun dropped the "2." from the number, so Solaris 7 incorporates SunOS 5.7, and the latest release SunOS 5.10 forms the core of Solaris 10.

12.2 Supported Architectures

Solaris uses a common code base for the platforms it supports:

• SPARC	• i86pc, which includes both x86 and x64.

Solaris is well-suited to symmetric multiprocessing, supporting a large number of CPUs. It has historically been tightly integrated with Sun's SPARC hardware and since Solaris 7 has included support for 64-bit SPARC applications. Solaris is marketed as a combined package with SPARC hardware. This has often led to more reliable systems, but at a cost premium over commodity PC hardware.

Solaris has also supported x86 systems since Solaris 2.1 and the latest version, Solaris 10, includes support for 64-bit x86 applications. This allows Sun to capitalize on the availability of commodity 64-bit CPUs based on the x86-64 architecture.

Sun has heavily marketed Solaris for use with:

- its own "x64" workstations and servers based on AMD Opteron and Intel Xeon processors.
- x86 systems manufactured by companies such as Dell, Hewlett-Packard, and IBM.

12.3 Versions

Solaris version	SunOS version	Release Date	Major New Features
Solaris 10	SunOS 5.10	January 31, 2005	Includes x64 (AMD64/EM64T) support. DTrace: Dynamic Tracing Solaris Containers. SMF: Service Management Facility (SMF) which replaces init.d scripts, NFSv4. Least privilege security model. Support for sun4m and UltraSPARC I processors removed. Support for EISA-based PCs removed. Adds Java Desktop System (based on GNOME) as default desktop. Solaris 10 1/06 added the GRUB bootloader for x86 systems and iSCSI Initiator support. Solaris 10 6/06 added the ZFS filesystem.
Solaris 9	SunOS 5.9	May 28, 2002 (SPARC) January 10, 2003 (x86)	iPlanet Directory Server. Resource Manager. Solaris Volume Manager, extended file attributes. IKE IPsec keying. Linux compatibility added. OpenWindows dropped. sun4d support removed.
Solaris 8	SunOS 5.8	February 2000	Includes Multipath I/O. IPMP, first support for IPv6 and IPsec. This was for manual keying only. mdb modular debugger. RBAC: Introduced Role-Based Access Control sun4c support removed.

Solaris version	SunOS version	Release Date	Major New Features
Solaris 7	SunOS 5.7	November 1998	The first 64-bit UltraSPARC release. Added native support for file system meta-data logging; this is UFS logging. Dropped MCA support on x86 platform. Last update was Solaris 7 11/99.
Solaris 2.6	SunOS 5.6	July 1997	Includes Kerberos 5. PAM. TrueType fonts. WebNFS, large file support. enhanced procfs. SPARCserver 600MP series support dropped.

13 Sun Java Enterprise System Components

The Solaris 10 Operating System includes licenses to run some of the most popular software components of the Sun Java Enterprise System.

The following Sun Java Enterprise System components are an integrated part of Solaris 10:

Sun Java System Directory Server	The industry's most widely deployed, general-purpose, LDAP-based directory server. Solaris 10 includes a license for 200,000 directory entries.
Sun Java System Application Server	A Java 2 Platform, Enterprise Edition (J2EE platform) - compatible application server for developing and delivering server-side Java applications and Web services.
Sun Java System Message Queue:	A leading enterprise message server that provides the capability for building and deploying distributed services.

14 Sun Java Desktop System

The Sun Java Desktop System incorporates the best of open source innovation with significant contributions from Sun to give users a comprehensive, integrated desktop client environment.

This highly economical, alternative desktop software solution delivers a wide range of functionality, including:

- A familiar desktop look and feel based on the open source GNOME desktop.
- Office productivity through the StarOffice 8 Office Suite, which provides integrated word processing, spreadsheet, presentation, drawing, and database capabilities.
- Additional desktop tools, including the Mozilla Web browser, instant messaging, and integrated Evolution e-mail and calendar software.

14.1 Open Source Software

Solaris 10 includes 187 software products including the following:

- Apache, Tomcat, and Zebra software for network and Web services.
- Bison, GCC, Perl, and Python tools for software development.
- IP Filter, TCP Wrappers, and Secure Shell utilities for security.
- GNOME, Mozilla, and Evolution software for desktop usability.
- PostgreSQL and MySQL databases.

15 Integrated Development Tools

Developers need integrated tools that are compatible with all the environments on which they must deploy applications.

Sun has provided software tools from the F/OSS world and complements them with Solaris 10 utilities such as Solaris Dynamic Tracing - DTrace.

F/OSS tools integrated into the Solaris 10 Operating System include:

- The standard GNU development utilities.
- Perl and Python programming languages.
- Library support includes UNIX standard functions.
- Popular F/OSS libraries such as Glib, GTK, JPEG, PNG, Tcl/Tk, TIFF, XML, and zlib.

16 DTrace

The Solaris 10 Operating System dramatically improves the way system administrators and developers can identify the reasons for suboptimal system and application performance.

Solaris Dtrace: Dynamic Tracing technology provides the capability to troubleshoot complex systemic problems in real time.

Additional Solaris features provide the ability to identify and resolve hardware problems, and streamline and automate patch management.

- Real-time troubleshooting of systemic problems.
- New tools for low-level system debugging.
- System hardware testing and analysis.
- Fine-grained project accounting.
- Enhanced patch analysis and delivery tools.
- Existing applications benefit from Solaris 10 enhancements without modification

17 System Analysis Tools

Solaris 10 provides the following system analysis capabilities and tools:

- Thread analysis and monitoring tools, including lockstat, truss, and pstack.
- Memory management and debugging tools, including libumem, a high-performance multithreaded memory allocation library with built-in monitoring functions.
- Support for Intelligent IPMI: Platform Monitoring Interface, an industry standard for management of x64/x86-based servers.
- Modular Debugger-mdb and Kernel Modular Debugger-kmdb, extensible tools for monitoring and analyzing applications and kernel routines.
- System and application core administration and debugging tools.

18 Network

Exponential growth in web connectivity, services, and applications is generating a critical need for increased network performance.

With the Solaris 10 Operating System, Sun meets networking challenges by improving network performance without requiring changes to existing applications.

The Solaris 10 Operating System:

- Supports current IPv6 specifications.
- Speeds application performance by about 50 percent via an enhanced TCP/IP stack.
- Supports the latest networking technologies, such as 10 Gigabit Ethernet, wireless networking, and hardware offloading.
- Accommodates high-availability, streaming, and VoIP: Voice over IP networking features through extended routing and protocol support.

19 File Systems

zFS

Available in Solaris 10, Solaris ZFS incorporates advanced data security and protection features, eliminating the need for fsck or other recovery mechanisms. By redefining file systems as virtualized storage, Solaris ZFS scales extremely well.

The Solaris ZFS technology will also advancements by automating tasks and protecting data from corruption, and providing virtually unlimited scalability

UFS

UFS: UNIX File System, is the primary Solaris file system. It was designed to handle small, cacheable files accessed randomly by individual processes.

The UNIX File System is the primary file system for the Solaris Operating System. UFS is stable and for most applications is the file system of choice. The Solaris UFS has its roots in the Berkeley Fast File System (FFS) of the 1980s; it is the result of more than 20 years of enhancement, evolution, and stabilization.

Enhancements over the last several Solaris releases include metadata logging to improve both reliability and performance.

19.1 I/O Performance

Significant improvements have also been made to improve I/O performance for databases, provide fast access to directories with large numbers of files, and provide the ability to create multiterabyte file systems.

Solaris Volume Manager software minimizes downtime by providing continuous data access, even in the event of a hardware failure

NFS V4 - Network File System, adds enhanced security features, performance, and cross-platform interoperability

20 Virtualization

Virtualization can be used for reducing IT infrastructure costs and better manage service levels.

Solaris Containers is part of a comprehensive offering of Sun virtualization technologies which also includes LDoms - Logical Domains. It uses virtualization to provide for the maintenance of the one-application-per-server deployment model, while sharing hardware resources.

Virtualization:

- Dynamically moves or replicates applications to fit the changes of the business.
- Lowers administrative costs by safely combining multiple applications on a single system.
- Reduces conflicts among applications running on the same system by isolating them from one another.
- Supports predictive self healing to minimize fault propagation and unplanned downtime.
- Enhances security by preventing unauthorized access and unintended intrusions.