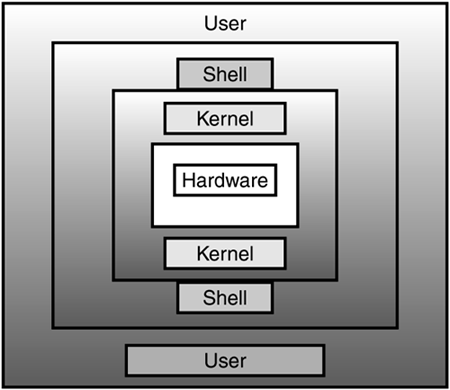
**SOLARIS ARCHITECTURE**

The Solaris operating system, which retains the basic structure of Unix, contains three main parts: the kernel, the shells, and the file systems (see Figure 1.1). Each is discussed within this section.

Figure 1.1. The Solaris operating system has three main parts: the kernel, the shells, and the file systems.



**The Kernel**

The kernel forms the core of an OS, providing basic services to all parts of the OS.

It is responsible for managing the system resources, such as devices, memory, CPU, and so on.

It allocates memory and time for various users and processes running on the system and assigns priorities for the processes. The kernel includes an interrupt handler, scheduler, and a supervisor to control the following functions:

* Memory management
* I/O services
* Process management
* System accounting
* Interrupt and error handling
* File management and security

The OS or an application program requests the services of the kernel through a specified set of program interfaces called the system calls.

**The Shell**

The shell is the layer above the kernel. The kernel starts a separate process that interacts with users, which is known as the shell. In a multiuser environment, such as Solaris, the shell must isolate the user interface from the kernel. This enables the kernel to allocate the system resources efficiently to all the users working on the operating system.

The shell acts as an interface between the user and the kernel. It interprets the commands executed by the user. No task is allowed to take direct control of the operating system in Solaris. Instead, each user gets a copy of the shell to interact with the kernel. The shell communicates with the kernel through system calls.

NOTE :

System calls are a set of routines that allow an application to access the kernel services.

**Shells in Solaris**

The features of the shells present in Solaris are as follows:

* Wild Card Patterns— Files that match a particular wild card pattern can be grouped together and actions can be performed on them.
* Interactive Processing— Communication between the system and the user takes the form of an interactive dialogue with the shell. This is known as interactive processing.
* Shell Scripts— These files contain a sequence of commands to be performed. This feature enables system administrators to perform their administrative tasks effectively by writing programs in the shell language.
* Background Processing— Some processes are time-consuming and are not interactive in nature. Such processes are done in the background while you continue using the system to do other tasks.
* Shell Variables— By storing data in variables, you control the behavior of shells as well as that of some programs and utilities. For example, the PATH variable stores the path names of all the directories to be searched for an executable file. The user controls the search process by specifying the alternate search path.
* Input/Output Redirection— Programs are instructed to obtain the input from a file other than the standard input device, which is the keyboard. This is called input redirection. Programs are instructed to send the output to a file rather than to the standard output, which is the visual display unit. This is called output redirection.
* Programming Language Constructs— The shell has features that enable it to be used as a programming language. Through the use of programming features, complex shells are built to perform complex operations.
* Pipes— Pipes can be used to develop simple programs that do complex operations with minimum effort. This reduces the necessity of writing new programs for the complex operations.

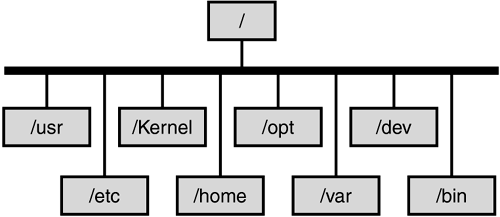
**File System**

The file system in Solaris is hierarchical in nature, resembling a tree-like structure, with the root directory at the top of the file system hierarchy. The root directory forms the base directory, which holds the core of the operating system—the kernel. The root directory contains the following directories:

* /usr— Contains system files and directories that are shared with other users. The/usr/bin directory, a subdirectory of /usr, contains executable utilities.
* /etc— Stores the system-related data that is used for obtaining the system configuration data and for administration purposes. It contains important files, such as /etc/password and /etc/hosts.
* /kernel— Contains important kernel files, loadable modules, and device drivers. The kernel is portable across hardware platforms.
* /export/home— Stores the home directories of the various users.
* /opt— Stores the application software, which may accompany the OS, the bundled software, as well as software obtained from the third parties.
* /var— Stores system log files, such as /var/adm/messages, and user mail directories.
* /dev— Contains device-related files, such as the device drivers for the system hardware.
* /bin— Contains the essential system binaries.

Figure 1.2 depicts the Solaris file system.

Figure 1.2. The file system in Solaris.



Solaris supports three types of file systems: disk-based, network-based, and virtual.

The file systems stored on physical media such as hard disks and CD-ROMs are referred to as disk-based file systems. They facilitate constant storage and management of the data present in different physical storage media. The Unix file system (UFS), PC file system (PCFS), CD file system (ISO 9660), and DVD Universal Disk Format (UDF) are some of the available formats in which disk-based file systems can be written.

Network-based file systems store files on a remote server. Although the files are stored in a remote location, they can be accessed as easily as local files.

Virtual file systems are special memory-based systems. By providing access to kernel-related information, they enable efficient system management.

**Glossary**

* **interactive processing**

An interactive dialog with the shell through which communication between a system and the user is implemented.

* **shell scripts**

The file that stores a frequently used set of shell commands. The file can later be used to execute the stored commands with a single command.

* **background processes**

Processes that run in the background while the user continues to work with other tasks. These processes are time-consuming and are not interactive in nature.

* **shell variables**

Variables containing data that can be used to control the shell's behavior, as well as other programs and utilities. For example, the PATH variable stores the pathnames of all the directories to be searched for an executable file.

* **Input/Output Redirection**

The technique of instructing programs to obtain the input from a file other than the standard input device is called input redirection. Programs are instructed to send the output to a file rather than to the standard output, which is the Visual Display Unit. This is called output redirection.

* **programming language constructs**

Features that enable a shell to be used as a programming language. These features can be used to build shell scripts that perform complex operations.

* **pipes**

A mechanism that enables simple programs to be connected to perform more complex functions. This helps minimize the need to develop new programs.