SECTION 5. SOLARIS OVERVIEW

This section provides an overview of the Solaris command language used in SARSS implementations.

5.1 General.

a. The Operating System (OS) is a set of programs that controls the computer and provides tools to help you do your work. Solaris is Sun's implementation. The Solaris 2.x is sometimes referred to as SunOS 5.x; this section won't propagate that terminology. Any OS, especially one as powerful as Solaris consists of several different parts:

(1) *Memory Management*. Determines how much memory to allocate for each program. If enough memory is not available to run a given program, the memory manager will move other programs, or parts of them, to temporary disk storage as necessary. This is known as swapping or paging.

(2) *File System*. The file system is a structure used to locate and store programs and files on disk.

(3) *System Programs*. The System Programs are the software programs accessible directly by users, enabling them to manipulate files and devices in various ways for getting basic work done. These programs may include utility programs, text editors, language compilers, debuggers, and shells.

(4) *Device Drivers*. The Device Drivers or programs stand at the lowest level and allow control of actual hardware devices. These drivers negotiate between the kernel and the hardware bus.

(5) *Scheduler*. A Scheduler decides which user programs are run, when, and for how long. On a primitive OS such as MS-DOS, there is no scheduler because only one program at a time runs under the control of the OS.

b. Solaris organizes the activities of the SARSS Host system which provides a link between the computer and its users, terminals, printers, and peripheral devices. It allocates computer resources; allows many users to share the computer's resources without sacrificing productivity (Timesharing) and enables the user to execute more than one process simultaneously (Multitasking).

5.2. Components of Solaris. Solaris consists of many different processes.

a. *Kernel*. The kernel is the memory resident program that coordinates the functioning of the computer's internals. The kernel is the nucleus of the OS, but it runs invisibly so users are unaware of its activities.

(1) It includes the scheduler, memory-management routines, and device drivers, as well as a large number of built-in system functions that are hidden from the regular user.

(2) Typically, the kernel has to be "tuned" to accommodate the performance characteristics of the STAMIS it supports.

b. *Shell*. The shell is the medium through which users and the kernel communicate. The shell is a command interpreter that interacts directly with the user, prompting for commands and causing the commands to be executed. The shell runs commands when you type their names, expands wild-card characters (* and ?) and takes care of redirecting input and output. There are three main shells:

(1) *Bourne Shell*. Named after its inventor at AT&T Bell Laboratories.

(2) *C Shell*. Has the ability to access all files and directories. Some commands and some files essential to the performance of Solaris are heavily protected so that only the super-user can access, execute or change them.

(3) *Korn Shell*. Combines many of the benefits of the C Shell with a high degree of compatibility with the Bourne Shell.

c. *User Tools, Programs, and Commands*. There are six categories of tools, programs, and commands users should be familiar with. These are:

(1) Information Management allows users to remove, create and organize data files and directories into logical groupings and allows users to control the access of their data by other users.

(2) Electronic Communication includes the hardware-to-hardware communication of the system itself, as well as the commands that permit computers to communicate with other computers, terminals to communicate with other terminals, and users to communicate with other users. They can write or send mail to each other internally.

(3) Programming Environment assists users in maintaining programs by supplying language interfaces and utility programs, such as the C compiler. The shell also provides the ability for writing scripts.

(4) Text Processing provides programs such as the screen editor (vi), for creating and changing text. Word processing and spreadsheet packages are also available with the Solaris system.

(5) Additional utility programs provide over 200 commands for flexibility of data manipulation and programming. Solaris treats its input and output devices (such as terminals, printers, and tape drives) as files, so that I/O interfacing is simplified at the command level.

(6) Solaris commands consist of the command itself, options, and arguments that further instruct the command:

command [-options] argument1 ... argument2 ...

5.3 Solaris File Structure. This paragraph provides an introduction to the Solaris file system structure. It is intended to provide a basic overview of the Solaris environment.

a. *Solaris File System*. The Solaris file system is a little more complex than most file systems because of its unique file tree structure. Although it's called a tree, it is an upside-down variety as illustrated in figures 5-1 and 5-2.

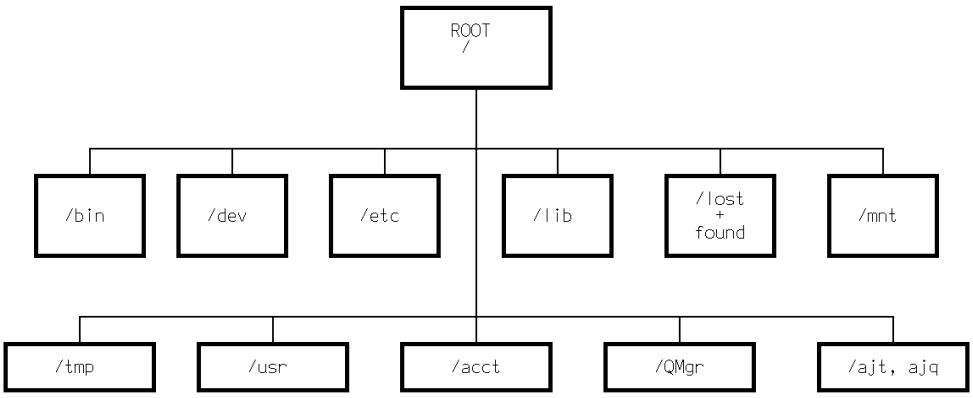

Figure 5-1. Solaris System File Structure.

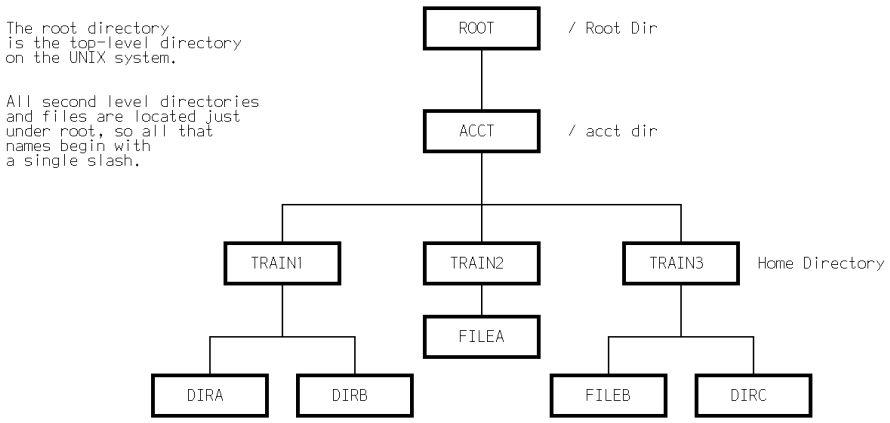

Figure 5-2. Solaris User File Structure.

b. *File Systems*. It is imperative that this file system be kept in perfect order, and checking the file system frequently is the only way to ensure its health and safety. File systems are logical collections of related files and directories that correspond directly to the physical structure of disk drives, once they are formatted.

c. *Solaris File Theory*. Understanding the Solaris structure will help you to work your way through file system corruption you may encounter. Files are the basic unit of the Solaris system. There are several different types of files:

(1) *Directories*. Contain a list of file names and other information related to these files.

(2) *Special Files*. Not disk files as such, but special path names that refer the input/output channels to the hardware.

(3) *Hidden Files*. These are ordinary files or directories preceded by a period (.) or "dot" and do not appear in an "ls" listing of directory contents.

(4) *Ordinary Files*. Contain text, data, or numbers.

5.4 Solaris File Naming Conventions. Several general conventions are adhere to in creating Solaris file names. Be descriptive. Distinguish between upper and lower case. File names can consist of upper case letters (A-Z), lower case letters (a-z), numbers (0-9), periods (.), commas (,), underscores (_), and plus (+). Names cannot contain special

characters (/ " ' * ; ? [ ] ( ) ! $ & : @). Names cannot contain embedded spaces. File names must be unique within a directory.

5.5 Pathnames.

a. Connecting directories and/or files together using a slash (/) is called a pathname. An example of a pathname is the path between the root and the user's home directory. When first assigned a user name, the user is also assigned a directory called his "Home Directory." A user is automatically placed in the home directory upon login. An example is /ajt/usr/ajt01.

b. A user can move from directory to directory by specifying the entire pathname, (full/absolute pathname), or from the current working directory using the relative pathname.

(1) A current working directory is represented by a period (.).

(2) A parent directory (one level up) is represented by two periods (..).

c. The command used to move between different directories is the "cd" command. Some examples are shown below.

(1) Move from root to the ajt01 directory.
cd /ajt/usr/ajt01 (full pathname)

(2) Move from the ajt01 directory to the ajt02 directory.
cd ../ajt02 (relative pathname)

(3) Move to the parent directory.
cd ..

(4) Move to the home directory.
cd

(5) The cd command will move you to your home directory from anywhere within the file system.

5.6 Specific Solaris File Structure.

a. *Solaris File Structure*. The Solaris file structure always starts with a system supplied directory called "root" which is represented by the symbol "/". The root is an origin or starting point for every directory or file that is created on the system.

b. *Directories Associated with Root File System*.

(1) /bin - Contains executable programs and commands.

(2) /dev - Contains special files that define all of the hardware on the system, such as the terminals, the line printers and disks.

(3) /etc - Contains utilities and system configuration files.

(4) /lib - Contains libraries for programs and languages.

(5) /lost+found - Used by fsck to save disconnected files of each file system.

(6) /mnt - Used to temporarily mount file systems.

(7) /usr/adm - Contains the directories and files needed for the SA menus.

(8) /var – Contains sub-directories for Solaris subsystems (varying data).

c. *Additional File Systems*. These separate file systems are unavailable in single user mode unless they are individually mounted.

(1) /tmp - Used for temporary files, which are deleted when the system is rebooted.

(2) /usr - Where "user commands" are established.

(3) /ajt,ajq,ajp - Contains application programs and working directories.

d. The key system files are described below:

(1) `/etc/issue`. Contains the signon message that is displayed when the terminal is turned on.

(2) `/etc/motd`. Contains the message of the day that is displayed when the user logs on. You can use this file to inform users each day of specific requirements, when the system is scheduled to shut down, etc.

(3) `/etc/passwd`. Contains the *usernames* and other information on all users. Maintaining this file is an integral part of a systems's security. Without an entry in this file, users are unable to log in to SARSS or Solaris. The /etc/passwd file is maintained by Admintool, an Open Windows platform tool with a graphical user interface. It contains seven fields separated by colons. A password file record format is shown in figure 5-3.

loginid:x:UID:GID:comment:home_directory:login_shell

Figure 5-3. Sample /etc/passwd File Record.

(a) *loginid* – Also called *username*, this field represents the user's login name. It is unique. The field is restricted to eight characters in length and does not contain uppercase characters.

(b) *x* – This field is a placeholder for the user's encrypted password, which is stored in the /etc/shadow file.

(c) *UID* – This field contains the user identification (UID) number that is used by Solaris to identify the user. UID numbers usually range from 100 to 60000. Values 0 through 99 are reserved for system accounts. UID 60001 is reserved for the *nobody* account. UID 60002 is reserved for the *noaccess* account. Duplicate UIDs are legal but should not be used. If two users have the same UID, they have identical access to the files each user creates.

(d) *GID* – This field contains the group identification (GID) number that is used by the system to identify the user's primary group. GID numbers for usually range from 100 to 60000.

(e) *comment* – This field usually contains the user's full name. This field is also referred to as the *gcos-field* for historical reasons.

(f) *home_directory* – This field contains the path name of the user's home directory.

(g) *login-shell* – This field defines the user's default login shell, which can be */bin/sh, /bin/csh, or /bin/ksh*.

(4) `/etc/shadow`. Contains the *usernames*, and encrypted passwords for all users. An x in the password field of the /etc/passwd database is a place holder for the encrypted password that is stored in this file. Only the super-user can access /etc/shadow. When a password is encrypted, it appears as a series of numerals and uppercase and lowercase letters unrelated to the actual password. This means that not even the super-user can read the password; only the system can read the special code. If a user forgets the password, the only recourse is to issue a new one. The /etc/shadow file should never be edited directly. It is maintained through Admintool or the command line utilities; useradd, usermod, and passwd. Each record contains nine fields, separated by colons. A shadow file record format is shown in figure 5-4.

loginid:password:lastchg:min:max:warn:inactive:expire:flag

Figure 5-4. Sample /etc/shadow Record.

(a) *loginid* – Also called *username*, this field contains the user's login name.

(b) *password* – This field may contain the following entries: a 13-character encrypted user password; the string *LK*, which indicates an inaccessible account; or the string *NP*, which indicates no password.

(c) *lastchg* – This field indicates the number of days between January 1, 1970, and the last password modification date.

(d) *min* – This field contains the minimum number of days required between password changes.

(e) *max* – This field contains the maximum number of days the password is invalid before the user is prompted to specify a new password.

(f) *warn* – This field contains the number of days that the user is warned before the password expires.

(g) *inactive* – This field contains the number of inactive days allowed for that user before the user's account is locked.

(h) *expire* – This field contains the date when the user account expires. Once exceeded, the user can no longer log in.

(i) *flag* – Reserved for future use, set to zero.

(5) `/etc/inittab`. Contains all devices that are connected to the system. It controls port and specialized processing including what processes are run during system startup accessed by Solaris during system initialization.

Blank Page