Chapter 1

Introduction To System Configuration and Operation

One of the first jobs of a system administrator is to bring his or her system online with an existing network (or standing alone), and to configure the system to meet the needs for which the system was installed. This configuration usually involves installing any necessary software and hardware, setting the name and network address of the system, creating accounts for the expected users, and generally taking a system from "out of the box" uniformity and customizing it to meet your preferences and your user's needs.

The tasks of installing necessary hardware are described in the documentation for the hardware. Software installation is described in the *IRIX Admin: Software Installation and Licensing* volume. This Guide describes the tasks you perform once the system has been powered-up to bring your system from its initial distributed state to the state in which you will use it.

This Guide assists you by describing the procedures most administrators use to configure their systems and explaining the reasons why these procedures exist and why they work the way they do. Some of these tasks are typically performed only at times of major change - when a system is commissioned, when ownership changes, or when there has been a significant hardware upgrade. Others are ongoing tasks or tasks that may come up during standard usage of an installed system.

This chapter provides directions to the information that system administrators use in the course of their tasks. Also, background information is presented on the general nature of the job of system administration. There are many good books on system administration listed in Appendix E, "Bibliography and Suggested Reading" of this Guide, and these are available through your local computer bookstore. Your Silicon Graphics system is similar to the systems described in many of these books, and it is different in significant areas as well. The principles of good system administration, though, are constant.

Principles Of Good System Administration

The following sections outline basic principles of good system administration. Each administrator must make individual decisions about the best practices for his or her site. The principles discussed here are generally considered as wise and safe practices.

Accounts and Passwords

To make your site as secure as possible, each user should have an account, with a unique user ID number, and each account should have a password. Users should never give out their passwords to anyone else under any circumstances. For more information on passwords and system security, see the *IRIX Admin: Backup, Security, and Accounting* volume.

Superuser (root) Account

Most system administration is performed while the system administrator is logged in as *root* (the superuser). This account is different from an ordinary user account because *root* has access to all system files and is not constrained by the usual system of permissions that controls access to files, directories, and programs. The *root* account exists so that the administrator can perform all necessary tasks on the system while maintaining the privacy of user files and the sanctity of system files. Other operating systems that do not differentiate between users have little or no means of providing for the privacy of users' files or for keeping system files uncorrupted. UNIX-based systems place the power to override system permissions and to change system files only with the *root* account.

All administrators at your site should have regular user accounts for their ordinary user tasks. The *root* account should be used only for necessary system administration tasks.

Access to the root Account

To obtain the best security on a multiuser system, access to the *root* account should be restricted. On workstations, the primary user of the workstation can generally use the *root* account safely, though most users should not have access to the *root* account on other user's workstations.

Make it a policy to give *root* passwords to as few people as is practical. Some sites maintain locked file cabinets of *root* passwords so that the passwords are not widely distributed, but are available in an emergency.

User Privacy

On a multiuser system, users may have access to personal files that belong to others. Such access can be controlled by setting file permissions with the chmod(1) command. Default permissions are controlled by the *umask* shell parameter. (See "Configuring Default File Permissions With umask" on page 100 for information on setting *umask*.)

However, to make it easier to exchange data, many users do not set their *umasks*, and they forget to change the access permissions of personal files. Make sure users are aware of file permissions and of your policy on examining other users' personal files.

You can make this policy as lenient or stringent as you deem necessary.

Checking the Password File

At least once a week, you should run the *pwck*(1M) and *grpck*(1M) programs to check your */etc/passwd* and */etc/group* files for errors. You can automate this process using the *cron*(1) command, and you can direct *cron* to mail the results of the checks to your user account. For more information on using *cron* to automate your routine tasks, see "Automating Tasks with at(1), batch(1), and cron(1M)" on page 24.

The *pwck* and *grpck* commands read the password and group files and report any incorrect or inconsistent entries. Any inconsistency with normal IRIX operation is reported. For example, if you have */etc/passwd* entries for two user names with the same User Identification (UID) number, *pwck* will report this as an error. *grpck* performs a similar function on the */etc/group* file. Note that the standard *passwd* file shipped with the system generates several errors.

Hardware Affects Software

Be aware that changing hardware configurations can affect the system, even if the change you make seems simple. Make sure you are available to help users with problems after the system is changed in any way.

Software Upgrades Affect Users

Changing the software also affects the system, even if the change you make is as trivial as a small upgrade to a new version of an application. Some software installations can overwrite customized configuration files. Your users may have scripts that assume that a utility or program is in a certain directory and a software upgrade may move the utility. Or the new version of the software simply may not work in the same way as the old version.

Whenever you change the software configuration of your systems, you should let your users know and be ready to perform some detective work if seemingly unrelated software suddenly stops working as a result. Make sure you are available to help users with problems after the system is changed in any way.

Before you upgrade your system to new software, check your user community to see which parts of the old software they use, and if they might be inconvenienced by the upgrade. Often users need extra time to switch from one release of an application to a newer version.

If possible, do not strand your users by completely removing the old software. Try to keep both versions on the system until everyone switches to the new version.

Notifying Users of Scheduled Maintenance

In general, try to provide the user community as much notice as possible about events affecting the use of the system. When the system must be taken out of service, also tell the users when to expect the system to be available. Use the "message of the day" file */etc/motd* to keep users informed about changes in hardware, software, policies, and procedures.

System Shutdown Procedures

Many administrative tasks require the system to be shut down to a run level other than the multiuser state. This means that conventional users cannot access the system. Just before the system is taken out of the multiuser state, users on the system are requested to log off. You should do these types of tasks when they will interfere the least with the activities of the user community.

Sometimes situations arise that require the system to be taken down with little or no notice provided to the users. This is often unavoidable, but try to give at least five to fifteen minutes of notice, if possible.

At your discretion, the following actions should be prerequisites for any task that requires the system to leave the multiuser state:

- When possible, perform service-affecting tasks during periods of low system use. For scheduled actions, use */etc/motd* to inform users of future actions.
- Check to see who is logged in before taking any actions that would affect a logged-in user. You can use the */etc/whodo*, */bin/who* and */usr/bsd/w* commands to see who is on the system. You may also wish to check for large background tasks, such as background compilations, by executing *ps -ef*.
- If the system is in use, provide the users advanced warning about changes in system states or pending maintenance actions. For immediate actions, use the */etc/wall* command to send a broadcast message announcing that the system will be taken down at a given time. Give the users a reasonable amount of time (five to fifteen minutes) to terminate their activities and log off before taking the system down.

Malicious Activities

You should set a policy regarding malicious activities. These include:

- deliberately crashing the system
- breaking into other accounts; for example, using password-guessing and password-stealing programs
- forging electronic mail from other users
- creating and unleashing malicious programs, such as worm and virus processes

Make sure that all users at the site are aware that these sorts of activities are potentially very harmful to the community of users on the system. Penalties for malicious behavior should be severe and the enforcement should be consistent.

The most important thing you can do to prevent malicious damage to the system is to restrict access to the *root* password.

Maintaining a System Log Book

It is important to keep a complete set of records about each system you administer. A system log book is a useful tool when troubleshooting transient problems or when trying to establish system operating characteristics over a period of time. Keeping a hardcopy book is important, since you won't be able to refer to an online log if you have trouble starting the system.

Some of the things that you should consider entering into the log book are:

- maintenance records (dates and actions)
- printouts of error messages and diagnostic phases
- equipment and system configuration changes (dates and actions), including serial numbers of various parts (if applicable)
- copies of important configuration files
- the output of *prtvtoc*(1M) for each disk on the system
- the */etc/passwd* file
- the */etc/group* file
- the */etc/fstab* file
- the */etc/exports* file

The format of the system log and the types of items noted in the log should follow a logical structure. Think of the log as a diary that you update periodically. To a large measure, how you use your system will dictate the form and importance of maintaining a system log.

In addition to the system log, you may find it helpful to keep a user trouble log. The problems that users encounter fall into patterns. If you keep a record of how problems are resolved, you do not have to start from scratch when a problem recurs. Also, a user trouble log can be very useful for training new administrators in the specifics of your local system, and for helping them learn what to expect.

Collecting User Requests

You should provide a convenient way for your users to report problems. For example, set up a "trouble" mail alias, so that users with problems can simply send mail to *trouble* for assistance.

Tasks of the System Administrator

The system administrator is responsible for all tasks that are beyond the scope of end users, whether for system security or other reasons. The system administrator will undoubtedly use the more advanced programs described in this guide.

A system administrator has many varied responsibilities. Some of the most common responsibilities addressed in this Guide are:

- Operations—seeing that the machine stays up and running, scheduling preventive maintenance downtime, adding new users, installing new software, and updating the */etc/motd* and */etc/issue* files. See Chapter 4, "Configuring The IRIX Operating System." Also see Chapter 5, "Configuring User Accounts and Managing User Issues."
- Failure Analysis—troubleshooting by reading system logs and drawing on past experience. See "Maintaining a System Log Book" on page 6.
- Capacity Planning—knowing the general level of system use and planning for additional resources when necessary. See Chapter 6, "Configuring Disk and Swap Space," and Chapter 11, "System Performance Tuning."
- System Tuning—tuning the kernel and user process priorities for optimum performance. See Chapter 11, "System Performance Tuning."
- Resource Management—planning process and disk accounting and other resource sharing. See the *IRIX Admin: Backup, Security, and Accounting* guide.
- Networking— interconnecting machines, modems, and printers. See the *IRIX Admin: Networking and Mail* guide.
- Security—maintaining sufficient security against break-ins as well as maintaining internal privacy and system integrity. See the *IRIX Admin: Backup, Security, and Accounting* guide.
- User Migration—helping users work on all workstations at a site. See the *IRIX Admin: Networking and Mail* guide.

- User Education—helping users develop good habits and instructing them in the use of the system. See Chapter 5, "Configuring User Accounts and Managing User Issues."
- Backups—creating and maintaining system backups. See the *IRIX Admin: Backup, Security, and Accounting* guide.

Administration Tools

Depending on the exact configuration of your system, you may have the following tools available for performing system administration:

System Manager
: This tool, available on graphics workstations, provides easy access to system administration functions. It features a quick and easy method of performing most system administration tasks. The System Manager is available only on those systems that have graphics capability.

Command-line tools
: The IRIX system provides a rich set of system administration tools that have command-line interfaces. These are especially useful for automatically configuring systems with shell scripts and for repairing the system in unusual circumstances, such as when you must log in remotely from another system.

  For example, using command-line tools, a site administrator can alter the system automatically at designated times in the future (for instance, to distribute configuration files at regular intervals). These commands are available on all IRIX systems.